GAZİOSMANPAŞA BİLİMSEL ARAŞTIRMA DERGİSİ (GBAD)
Gaziosmanpasa Journal of Scientific Research
ISSN: 2146-8168
http://dergipark.gov.tr/gbad
Araştırma Makalesi (Research Article)

Cilt/Volume : 8
Sayı/Number: 2
Yıl/Year: 2019
Sayfa/Pages: 35-48

***Alınış tarihi (Received)*:** 30.12.2018
***Kabul tarihi (Accepted)*:** 26.09.2019

Identification of OOV Words in Turkish Texts

Enis ARSLAN[1*], **Umut ORHAN**[2]

[1] *Çukurova Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana,*
[2]*Çukurova Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana,*
**Corresponding author e-mail: enisarslan@gmail.com*

ABSTRACT: In this study, we present a semantic graph network model which is capable of detecting out-of-vocabulary (OOV) words in Turkish texts. In natural language processing (NLP) field, morphological analyzers can encounter unknown words (UW) during word processing. This mostly occurs when these kind of tools depend on a dictionary to find the probable lemmas in order to further process parsing. Sometimes, an analyzer is unable to find any candidates because of the non-existence of the lemma candidates in the dictionary. This results in degraded parsing output. The proposed model for OOV detection is able to define OOV words which are suitable for dictionaries. Also co-occurrence relations of the lemmas in texts are modelled as a semantic sub-graph and it is used to discover collocations to propose as new lemma candidates.

Keywords: *Unknown words, Collocation, Co-occurrence, OOV words*

Türkçe Metinlerde Sözlük Dışı Kelimelerin Tespiti

ÖZET: Bu çalışmada, Türkçe metinler için sözlük dışı kelime (SDK) tespiti yapabilen anlamsal bir çizge ağı modeli sunulmuştur. Doğal dil işleme (DDİ) alanında, biçimbirimsel çözümleyiciler, kelime analizi esnasında bilinmeyen kelime (BK)'lerle karşılaşabilmektedirler. Bu durum daha çok, bu tip araçların çözümleme esnasında aday bulabilmeleri için bir sözlüğe bağımlı oldukları durumlarda oluşmaktadır. Bazen, bir çözümleyici madde başı adaylarının sözlükte mevcut olmaması sebebiyle hiçbir madde başı adayını bulamamaktadır. Bu durum çözümleme çıktı değerini düşürebilmektedir. Sözlük dışı kelime (SDK) tespiti için önerilen model, sözlükler için uygun olabilecek sözlük dışı kelimeleri tespit edebilmektedir. Ayrıca çizge veri tabanında birliktelik ilişkileri kullanılarak bir anlamsal alt-ağ oluşturulmuş ve yeni eşdizimliliklerin madde başı olarak önerilecek şekilde keşfedilmesi amacıyla kullanılmıştır.

Anahtar Kelimeler: *Bilinmeyen kelimeler, Eşdizimlilik, Birliktelik, Sözlük dışı kelimeler*

1. Introduction

Out-of-vocabulary (OOV) words are the word sequences which form new words those do not exist in the current dictionaries [Nakagawa, 2004]. In some research, OOV term is used in the same meaning with unknown words (UW) term. To imply the difference, in general scope, UW term can also be regarded as the first encountered form of an unknown word which is met in a sentence. OOV term denotes the lemmatized form of a word if it does not exist in the dictionary. In NLP literature, defining the UWs in the sentences is called unknown word identification (UWI).

In many natural language processing (NLP) studies, UWI is a challenging problem for the researchers. This problem is generally studied under lemmatization [Nakagawa, 2004], information retrieval (IR) [Loponen and Kalervo, 2010], cross lingual IR (CLIR) and speech recognition [Parlak and Saraclar, 2008][Arısoy et al., 2006][Bazzi and Glass, 2002] topics. Sometimes UW’s are used to measure the performance of the morphological analyzer by adding them to the test datasets [Jongejan and Dalianis, 2009]. By definition, lemmatization is the process of transformation of the inflected word forms to lemma forms. Lemmatization is an essential normalization step for many NLP tasks like information retrieval (IR), machine translation (MT) [Erjavec and Džeroski, 2004]. Because of the performance effect, in such applications, better lemmatizer design study is going on for the agglutinative languages like Turkish [Çöltekin, 2014][Silfverberg et al., 2016].

Lemmatizers generally use rules and dictionaries but they are weak when they meet an OOV word during lemmatization [Loponen and Kalervo, 2010]. In information retrieval (IR) studies, when the text is not correctly lemmatized because of OOV words, expected results may not be met. In speech recognition field this situation will increase recognition errors [Arısoy et al., 2006]. Also some state-of-the-art POS taggers have performance problems because of the non existing words (OOV) in the train set [Lafferty et al., 2001].

Detection of the OOV words is crucial, especially in morphologically complex languages like Czech, Turkish, Finnish, but this is also problematic for other languages [Loponen and Kalervo, 2010]. Turkish Language is free word order by its nature. Because of this, in Turkish, many new words and complex n-grams can be produced. This is generally the main cause of meeting new OOV constructs (increase in OOV-rate) in a text [Arısoy et al., 2006].

The basic methodology to follow, to decrease the OOV-rate is to extend the dictionary. But this requires more human labor.

In this study, we aim to detect OOV words in Turkish texts which are generally seen as UWs of a morphological analyzer or lemmatizer output. Collocation is the aggregation of some words in necessity of composing a meaning. We also present a semantic sub-graph consisting of cooccurences which are useful in detecting collocations as new lemma candidates. In the second part of the related work in this topic is presented, in the third part, methodology is given and fourth part consists of the results of the study.

2. Related Work

Analysis of OOV words depend on the nature of the researched language. Because of the agglutinative morphology, Turkish and Finnish languages are high in OOV rate [Parlak and Saraclar, 2008].

Morphologically rich languages like Turkish, Finnish and Czech generally rely on statistical or rule based approaches to generate OOV words. to be integrated to morphological analyzers. To solve the OOV word problem, generally, the main idea is to incorporate the new words to dictionary but it is a challenging task [Korobov, 2015].

In Finnish [Loponen and Kalervo, 2010] , they have developed a corpus and dictionary independent lemmatizer for this purpose. They have created a rule-set from a supervised dataset and by using the statistical information they define the OOV words.

Korobov’s study presents a rule-based tool named pymorphy2 which is used to generate possible OOV words for Russian language by keeping the track of the analyzer units.

In [Parlak and Saraclar, 2008], in spoken term detection topic, they have used lattice-based indexing method for OOV word detection, similarly to the morph-based indexing methods for Finnish.

Statistical methods has the advantage to be independent from the dictionaries [Arısoy et. al. 2006] [Arısoy et. al. 2009]. In a rule-based study, Brill [Brill, 1995] has first applied transformation to the data and scored the output in labelled data. Rules with highest ranking are selected and applied to whole dataset. The composed data was accepted as the new input data and this learning process went on until the scores remained unchanged. In Daciuk’s [Daciuk, 1999] study, FSAs was directly produced from data and the affixes are naturally found in contrary to the other studies. This proposed method has the speed advantage provided by FSAs and can be used for the dictionary enlargement studies. In another study, Asahara [Asahara and Matsumoto, 2004] has used character-based chunking method for OOV detection in Japanese. This methodology uses a morphological analyzer and a chunker.

3. Methodology

In this study, we propose a semantic graph system that is capable to detect the UWs and expand a static dictionary by adding new lemmas. This system can discover the lemmas of some UWs by using ability of the Finite State Automatas (FSA) designed for the Turkish language. For this purpose, we have built a four phase system. In the first phase all lemmas in the dictionary of Turkish Language Association (TLA) are added as nodes into a semantic graph network. In the second phase, news sentences from the corpus [Tahiroglu, 2014] were processed by applying tokenization and lemmatization to obtain the lemmas. During this process, co-occurrence information of the tokens in each sentence were added as ‘graph relations’ to the lemmas, thus a semantical network was established. In the third phase collocations were discovered by using the semantical graph network. and in the fourth phase lemma candidates which are one of the lemmatization outputs of UWs in the sentences, were pruned. These stages are depicted in Figure 1.

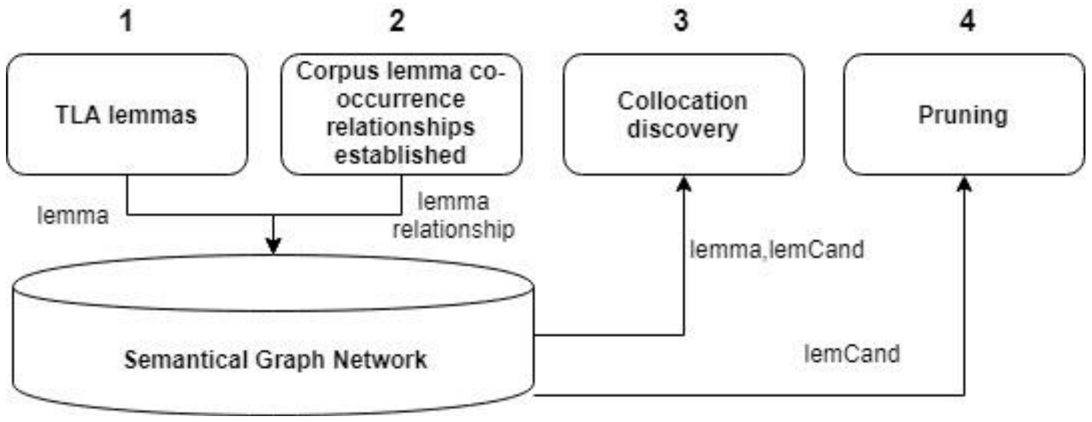


Figure 1. System flowchart

50,000 sentences which were retrieved from Çukurova University Turkoloji Corpus [Tahiroglu, 2014] were used as the dataset. In order to obtain certain amount of frequency values, every 1,000 sentences, collocation discovery and lemma candidate pruning was applied iteratively.

3.1. Adding Dictionary lemmas to the semantical graph network

In order to begin constructing a semantical graph network, all of the lemmas of the TLA were added as nodes with 'Lemma' label and certain properties such as 'frequency', 'postag'. Lemma statistics of the TLA classified with noun and verb postags are listed in Table 1.

Table 1. TLA lemma statistics in static graph

	Verb	**Noun**	**Total**
Lemma count	12,706	66,658	79,364

There are 535 lemmas with both verb and noun postags in the graph. Seperate nodes are created for each of these nodes in the same name with different postags. We accept and behave these node as different entities in order to allow these nodes to be able to establish different graph relations with other nodes.

3.2. Lemmatization and developing the semantical graph network

At this stage, tokenization was applied for all sentences which were received from the corpus. During tokenization, punctuation and all words containing the numerical characters were ignored. In order to detect the possible inflected forms of the TLA compound words in the sentences, each token was evaluated seperately. Compound word candidates were detected by checking the token phrase in the sentence with the similar compound words (in lemma format) in the semantic graph network by using the Affix validation function (AVF) [Arslan and Orhan 2017]. AVF accepts the surface form of a word and its possible lemma as input. It takes the difference of these words to obtain the affix part. And this affix part is evaluated in the FSAs to be a proper affix for Turkish language. If this function returns 'True' a token pointer in the sentence feeds forward by the length of the phrase.

Tokenization ends when the pointer is at the end of the sentence. For example when the below sentence is tokenized:

‘Dışarıda hava günlük güneşlik görünüyordu.’

Semantic graph network is searched for the compound terms starting with the words ‘dışarıda’ and ‘hava’. When tokenizer checks for the word ‘günlük’, the function returns ‘günlük güneşlik görün’ compound word which is found in the semantic graph network. Following that, ‘günlük güneşlik görünüyordu’ and ‘günlük güneşlik görün’ phrases are checked with AVF. If AVF returns ‘True’, ‘günlük güneşlik görünüyordu’ is a token. In the last case, the tokens in the sentence are:

‘Dışarıda’,’hava’,**’günlük güneşlik görünüyordu’**

Lemmatization

Following the tokenization process, a morphological analysis was applied for each token. This analysis results in detecting the lemma and postag of the token and increasing the corresponding frequency values.

In the first stage of the morphological analysis, any token from the token list was checked to be an existing lemma in the semantic graph network. If it exists as a lemma, the lemma’s node frequency values are incremented by 1. In the other case, it is an inflected word. Lemma candidates for this word are detected from the semantic graph network. This word and the lemma candidates are sent to AVF for validation. Validated lemma candidates’ frequency values are incremented by 1 as before. If lemmatization process returns null value, this token word is accepted as an UW and ‘Lemma discovery function (LDF) runs.

Lemma Discovery Function (LDF):

Lemma Discovery Function (LDF) takes the UW token as an input and looks for the lemma candidates inside the token by iteratively removing one character from the right. When token (inflected word) as X and its length is denoted by N, the lemma candidates can be detected as below:

```
N = LEN(X) - 1
$X_{cand}$ = LEFT(X, N)
WHILE (N >1)
  checkVal = AVF ($X_{cand}$, X, “isim”)
  IF (checkVal)
      CREATE Node {name:$X_{cand}$, postag:“isim”}
  checkVal = AVF ($X_{cand}$, X, “fiil”)
  IF (checkVal)
      CREATE Node {name:$X_{cand}$, postag:“fiil”}
  N = N – 1
  $X_{cand}$ = LEFT($X_{cand}$,N)
END WHILE
```

In the pseudo-code above, ‘X_{cand}’ denotes a lemma candidate. ‘checkVal’ value is obtained by sending the lemma candidates and the inflected word to AVF with ‘noun’ and ‘verb’ postags. AVF sends the word pairs to the related Finite State Machines (FSM) according to the given postag input. The result is 'true' if this pair (an inflected word and a lemma candidate) are validated with AVF. Validation results in a new lemma candidate node labelled as 'LemCand' (X_{cand}). This new graph node is linked to the inflected word with a 'MORPH' relation in the graph. A comparison for each postag is done because, in Turkish, there can be a lemma with two different postags having the same name. In each iteration, a character is removed from the right of the word to obtain a lemma candidate. This goes on when the length of the candidate is greater than 1. In each iteration, exceptions for Turkish language like consonant voicing changes, consonant dropping are considered and the lemma candidate creation are handled accordingly. For example:

'Emekli aylığımı bugün aldım.'

when 'aylığımı' word is an UW, it is checked with LDF to validate the inital candidates 'aylığım', 'aylığı', 'aylığ'… In this example, LDF detects consonant voicing change and adds the 'aylık' word as a candidate for grammatical validation.

LDF is described with an example below:

We know that the lemma of the word 'ötekileştirebilensin' is 'ötekileştir'. While TLA dictionary does not contain the lemma 'ötekileştir', lemmatization output for this is word is: UW. Thus we can say that 'ötekileştir' is an OOV word. The word 'ötekileştirebilensin' is sent to LDF in order to discover any lemma candidates inside this surface form. An example of the intermediate steps during the execution of LDF are shown in Table 2:

Table 2. Lemma discovery for the word 'ötekileştirebilensin'

iteration	candidate (A)	comparison result
1	ötekileştirebilensi	FALSE
2	ötekileştirebilens	FALSE
3	ötekileştirebilen	TRUE
4	ötekileştirebile	FALSE
5	ötekileştirebil	TRUE
6	ötekileştirebi	FALSE
7	ötekileştireb	FALSE
8	ötekileştire	FALSE
9	ötekileştir	TRUE

'ötekileştirebilensin' word is sent to lemmatization with each candidates listed in column A and if the comparation results as 'True' a lemma candidate node is created with the name in A and labelled as 'LemCand'. 'LemCand' labelled nodes are created with a few types of frequency parameters. Frequency types of nodes are listed in Table 3:

Table 3. Frequency types of nodes

Frequency label	**Description**
freqSLem	Lemma* is in nominative form in the sentence
freqNLem	Lemma* is in inflected form in the sentence
freqYLem	Lemma* is with ‘verb’ postag as the last word in the sentence
* Lemma, LemCand, VerLemCand, CollCand	

Lemma candidates with label ‘LemCand’ are mostly meaningless lemmas. All tokens in a sentence are processed in this sequence.

Following the lemmatization process, all related TLA lemmas and lemma candidates (LemCand) are linked together with a new relation type ‘COOCCUR’. Linking condition is provided when two lemmas occur in the same sentence. As a rule, tokens with ambiguity should not conribute as lemmas to this network. This rule is applied to obtain a more accurate graph in semantic means.

At first, during the processing of each token, the lemmas are collected in a list. If the token is an existing lemma it is added to the list directly. If it is an inflected word, it is lemmatized and when the lemmatization output consists of only one lemma, it is added to the lemma list. All lemmas collected in the list are linked with one-to-one relation with a ‘COOCCUR’ relation type. When lemma list of a sentence is named as ‘candList’, links are established as follows:

```
FOR EACH ELEMENT i IN candList
  FOR ELEMENTS DIFFERENT FROM i AS j
    IF i IS CONNECTED TO j
        relFreq++
        IF i IS NEIGHBOR TO j
            neighFreq++
        END IF
    ELSE   %% if i not connected to j yet
        ESTABLISH_RELATION(i,j,‘COOCCUR’)
        relFreq=1
    END IF
  END FOR
END FOR
```

while ‘relFreq’ value of ‘COOCCUR’ type denotes lemma candidates cooccurring in any place of the sentence (CONNECTED), neighFreq value gives the count of the bigrams to be exactly located as neighbors in the same sentence (NEIGHBOR). Each lemma node in candList are checked with the others to have any relation. If there does not exist any relation between, a relation is established with type ‘COOCCUR’ and ‘relFreq’ value is set to 1. If there is an existing relation, ‘relFreq’ value is incremented by 1. If they are neighbors in the same sentence, neighboring frequency ‘neighFreq’ value is also incremented by 1.

The flowchart for the composition of the semantic graph is presented in Figure 2.

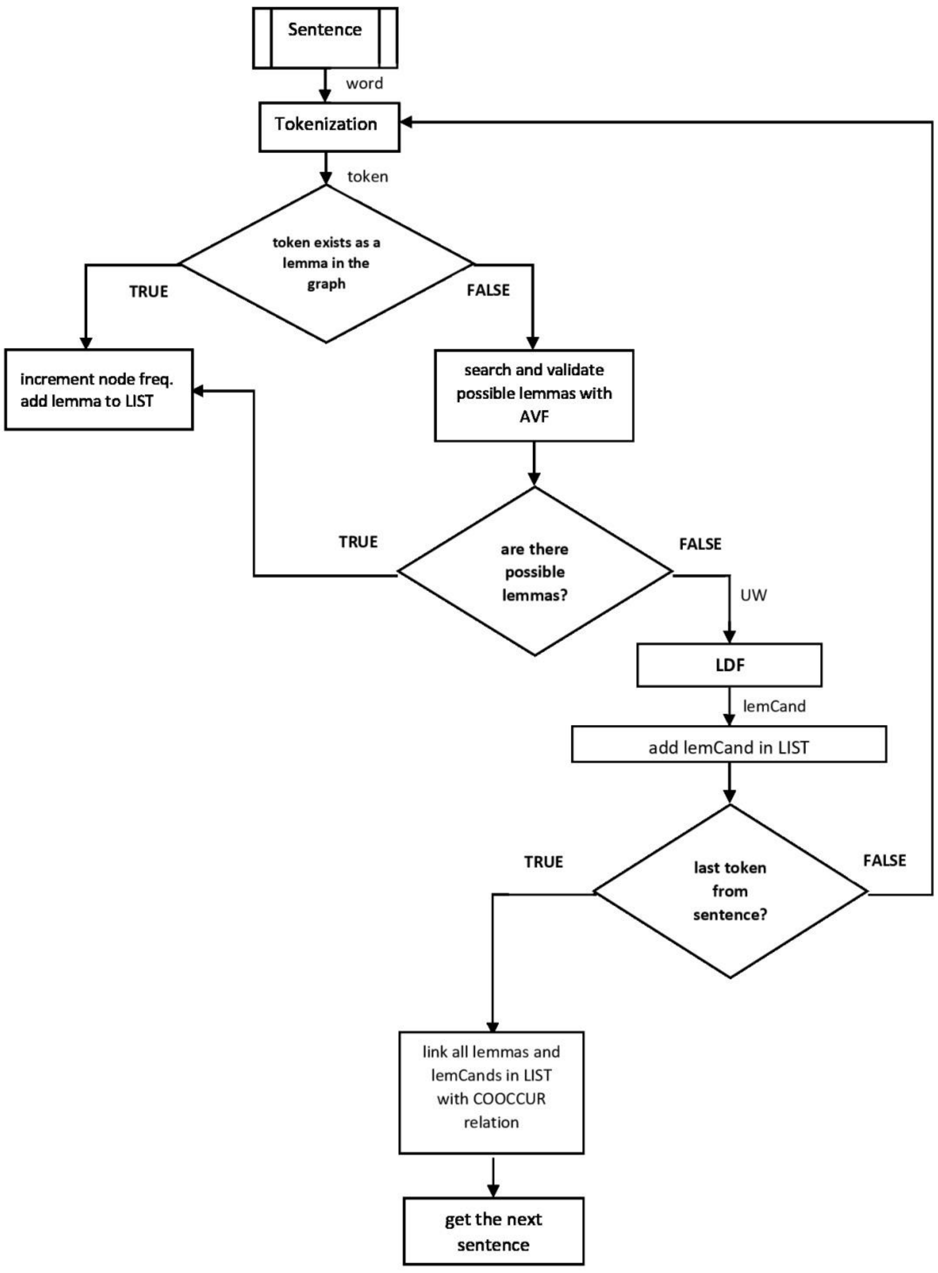

Sentence
word
Tokenization
token
token exists as a lemma in the graph
TRUE
FALSE
increment node freq. add lemma to LIST
search and validate possible lemmas with AVF
TRUE
are there possible lemmas?
FALSE
UW
LDF
lemCand
add lemCand in LIST
TRUE
last token from sentence?
FALSE
link all lemmas and lemCands in LIST with COOCCUR relation
get the next sentence

Figure 2. Composition of the semantic graph

3.3. Discovering the collocations

'COOCCUR' relation type in the semantic graph network provides us the neighboring information that can be used to discover collocations. In order to select the appropriate collocations, lemmas and LemCands in the semantic network should provide the following conditions:

n: 'first word in the bigram', m: 'second word in the bigram', R:'COOCCUR relation within any n and m'

```
IF (R.neighFreq/R.relFreq)>0.86 AND (r.neighFreq>10)
THEN CONCATENATE n,m
```

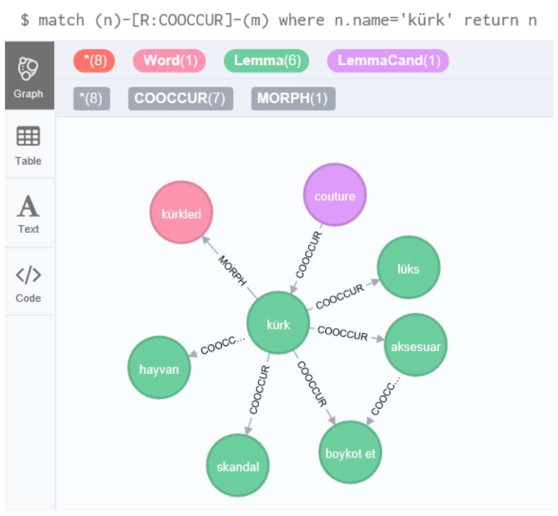
By using the R.neighFreq/R.relFreq ratio, it is aimed to detect the words with high neighboring values which can exist between some of the words in a sentence. Static parameters such as 0.86 are selected empirically. The first threshold was given 0.86 value because noise under this value was very high. As a second threshold, 10 was selected to limit the results which satisfy the first condition.

Discovered collocations are created as new nodes labeled as 'CollCand' in the semantic graph network with the postags 'verb' and 'name' seperately. All the node and relation types in the semantic graph network are listed in Table 4:

Table 4. Node and relation labels of the semantic graph network

Label	Type	Description
Lemma	Node	Lemma from TLA Dictionary
LemCand	Node	Lemma candidate discovered by LDF
VerLemCand	Node	Verified Lemma candidates selected after pruning (Section 3.4)
CollCand	Node	Collocation candidates discovered in Section 3.3
COOCCUR	Relation	Relations established between lemmas and LemCands

Some types of nodes and relations listed in Table 4. are shown in an example for the word 'kürk' in Figure 3:



```
$ match (n)-[R:COOCCUR]-(m) where n.name='kürk' return n
```

Figure 3. Different types of nodes and relations for the word ‘kürk’

As seen in Figure 2, some of the semantical connections linked to the word ‘kürk’ can be seen like ‘lüks’, ’aksesuar’, ‘skandal’ with ‘COOCCUR' relation type.

3.4. Pruning of the lemma candidates

Due to being produced combinationally by LDF, some of the LemCands may be meaningless and with double postags. By using the frequency types’ values listed in Table 3, candidates with double postags can be reduced to one postag and the ones with one postag can be verified or eliminated from the system according to their frequency values. In the first case, selection for the candidates with double postags was done according to the following algorithm:

```
Lemma candidate’s (C) different nodes with the same name with different postags as:
$N=C_{isim}$ and $M=C_{fiil}$
$N_{sum}= freqSLem_N+freqNLem_N$
$M_{sum}=freqSLem_M+freqNLem_M$
IF($N_{sum} > M_{sum}$)
  SELECT N INTO LIST
ELSE IF($N_{sum} < M_{sum}$)
  SELECT M INTO LIST
ELSE IF ($N_{sum} = M_{sum}$)
  FOR (M)
    IF ($freqYLem/ M_{sum}$) > 0.2
      SELECT M INTO LIST
   END FOR
```

In the first stage of the algorithm, the largest value from frequency sums of the nodes are selected. If these sum values are the same, another frequency value named ‘freqYLem’ is checked for the postag ‘verb’. This formula denotes that if a lemma exists at the end of the sentence, it is highly probable to be a ‘verb’. In the second stage, the selection for the lemma candidates with single postags are done with the following condition:

```
K=Lemma candidate node with single postag
IF (K.freqSLem+K.freqNLem) > 2
```

SELECT K INTO LIST

The values 0.2 and 2 in the preceding algorithms were empirically selected and better parameter values can be achieved with many trials. Following the selections assuring the conditions in the algorithms, the labels of the lemma candidates existing in the ‘LIST’ are updated with ‘VerLemCand’.

4. Experimental Results

4.1. Semantic graph network

Semantic graph network statistics after the processing of all the sentences in the corpus are listed in Table 5:

Table 5. Statistics of the semantic graph network

Description	Label	Count
Lemma node	Lemma	79,364
Lemma candidate node	LemCand	47,979
Verified Lemma candidate node	VerLemCand	3,173

As seen in the table, the candidates produced with the ‘LemCand’ label are very high in number. After the selection of the candidates with high frequency values there remains 3,173 verified lemma candidates. In other words, only 6,6% of the lemma candidates are useful discoveries.

4.2 Outputs of the LDF function

The top 5 discovered candidates with ‘VerLemCand’ label in semantic graph network are shown in Table 6.

Table 6. Discovered OOVs with high frequencies

OOV	Frequency value	Existence in the TLA Large Dictionary
nintendo	23	No
iskender	17	Yes
musaddık	14	Yes
işgücü	13	Yes
ötekileştirmek	4	No

The last column of Table 6 shows denotes the discovered word when it exists in the parent dictionary of TLA named ‘Büyük Sözlük’. Sometimes the discovered words can not exist in ‘Güncel Sözlük’ but can exist in ‘Büyük Sözlük’ of TLA. It was observed that the discovered candidates were generally proper nouns, abbreviations and spelling errors. In the table, proper nouns like ‘nintendo’,’mourinho’ takes attention.

4.3 Results of the collocation discovery

The collocation discoveries occurring with high frequency values of ‘COOCCUR’ relation type in the semantic graph network are listed Table 7:

Table 7. Words discovered and proposed as collocation

Discovered word	Frequency value
başta olmak	93
gerek olmak	88
konu olmak	59
açıklama yapmak	42
daha fazla	26
üzerinde bulunmak	22
maç kazanmak	21
maç oynamak	21
bayram günü	21
anne baba	19

This method is also applicable to discover trigrams (or n-grams) like ‘görüş dile getirmek’.

Because of the difficulty of producing rules for all types of compound words, sometimes it is possible to propose erroneous candidates with ‘CollCand’ label. For example, system proposes the words ‘maliye bakanlık’ and ‘basın açıklama’. But their correct form should be ‘maliye bakanlığı’and ‘basın açıklaması’. The cause of this error is the use of lemmas to compose collocation candidates. This error can be corrected as ‘Maliye bakanlık’ by the upcoming lemmatization iterations when this candidate is added to semantic graph network and processed by further inflected words coming from the sentences. For example when an inflected word comes from a sentence like ‘Maliye bakanlığından’ it will match ‘Maliye bakanlık’ lemma candidate After the verification with AVF, freqNLem will be incremented, freqSLem will remain 0. After many iterations if freqSLem value remain zero, while freqNLem has big value, ‘Maliye bakanlık’ lemma node will be exchanged with alternative root ‘Maliye bakanlığı’.

Because AVF applies the grammatic controls from the right of the words, some collocations which have also inflection in the first word were also produced erroneously. For example the system produces ‘sovyet birlik’ collocation but the right form of the word should be ‘sovyetler birliği’.

R-Precision value is given in Table 8:

Table 8. R-precision value for the methodology

Correctly discovered	2,276
Incorrectly discovered	897
Total	3,173
R-precision value	71.7%

As seen in Table 8, there are 3,173 distinct discovered lemmas (VerLemCand). All of these lemmas are checked by expert and 897 were useless. Most of the discovered lemmas are Named entitities (NE), bigrams as compound names and some new words recently began to be used by people. The success rate of the methodology is listed as 71,7% where it is expected that this value can be increased by the process of bigger corpus.

5. Conclusion

Like all other languages, Turkish language interacts with the outer world and many new words are included with their inflections making the language richer in words. It is not always possible to detect and add these words manually to the dictionaries. Because of this, discovering the OOV words by using the computer aided methods, both will increase the success of the NLP methods and will contribute to dictionary extension facilities.

In our knowledge, there does not exist any focused study conducted for discovering OOVs in Turkish language. So it was not possible for us to give comparable results for this study. Also, although TLA experts sometimes adds new lemmas to the dictionary, it was not possible to find any statistics about the new contributions the experts has made. In this study, only 50,000 sentence from Turkish newspapers were processed and 3,173 new lemmas were discovered to be added to the dictionary.

Proposed lemma candidates were selected according to frequency thresholds and double checked by experts. We believe that, processing more sentences with more detailed rules will allow the system to be able to make more precious selections and as a result erroneous selections will decrease.

6. Acknowledgements

This study is a part of the research programme with project number 215E256, which is financed by the Scientific and Technological Research Council of Turkey (TUBITAK).

7. References

Arısoy, E., Dutağacı, H., Arslan, L.M., 2006. A unified language model for large vocabulary continuous speech recognition of Turkish. *Signal Processing*, *86*(10), pp.2844-2862.

Arısoy, E., Can, D., Parlak, S., Sak, H. and Saraçlar, M., 2009. Turkish broadcast news transcription and retrieval. *IEEE Transactions on Audio, Speech, and Language Processing*, *17*(5), pp.874-883.

Arslan, E, Orhan, U. 2017. Using Graphs in Construction of a Lemmatization Model for Turkish, International Mediteranean Science and Engineering Congress, IMSEC.

Asahara, M., Matsumoto, Y., 2004, August. Japanese unknown word identification by character-based chunking. In *Proceedings of the 20th international conference on Computational Linguistics* (p. 459). Association for Computational Linguistics.

Bazzi, I., Glass, J., 2002. A multi-class approach for modelling out-of-vocabulary words. In *Seventh International Conference on Spoken Language Processing*.

Brill, E., 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics*, *21*(4), pp.543-565.

Çöltekin, Ç., 2014. A set of open source tools for Turkish natural language processing. In *LREC* (pp. 1079-1086).

Daciuk, J., 1999, July. Treatment of unknown words. In *International Workshop on Implementing Automata* (pp. 71-80). Springer, Berlin, Heidelberg.

Erjavec, T., Džeroski, S., 2004. Machine learning of morphosyntactic structure: Lemmatizing unknown Slovene words. *Applied Artificial Intelligence*, *18*(1), pp.17-41.

Jongejan, B., Dalianis, H., 2009. August. Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1* (pp. 145-153). Association for Computational Linguistics.

Korobov, M., 2015. April. Morphological analyzer and generator for Russian and Ukrainian languages. In *International Conference on Analysis of Images, Social Networks and Texts* (pp. 320-332). Springer, Cham.

Lafferty, J., McCallum, A. and Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Loponen, A., Kalervo, J., 2010. A dictionary-and corpus-independent statistical lemmatizer for information retrieval in low resource languages. *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, Berlin, Heidelberg, 2010.

Nakagawa, T., 2004. Chinese and Japanese word segmentation using word-level and character-level information. In *Proceedings of the 20th international conference on Computational Linguistics* (p. 466). Association for Computational Linguistics.

Silfverberg, M., Ruokolainen, T., Lindén, K. and Kurimo, M., 2016. FinnPos: an open-source morphological tagging and lemmatization toolkit for Finnish. *Language Resources and Evaluation*, *50*(4), pp.863-878.

Parlak, S., Saraclar, M., 2008. "Spoken term detection for Turkish broadcast news." *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008.

Parlak, S., Saraclar, M., 2008. March. Spoken term detection for Turkish broadcast news. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on* (pp. 5244-5247). IEEE.

Tahiroglu, B.T., Akalın, S.H., Ozkan, B., 2014. Turkce Cevrim Ici Haber Metinlerinde Yeni Sozlerin (Neolojizm) Otomatik Çıkarımı. In Turkce Uzerine *Derlembilim Uygulamaları*, Karahan Kitabevi.