# All-Words Word Sense Disambiguation for Turkish

## Türkçe için Çoklu-Kelime Belirsizlik Giderme

Onur Açıkgöz[1], Ali Tunca Gürkan[1], Burak Ertopçu[1], Ozan Topsakal[1]
Berke Özenç[1], Ali Buğra Kanburoğlu[1], İlker Çam[1]
Begüm Avar[2], Gökhan Ercan[1], Olcay Taner Yıldız[1]
onur.acikgoz@isik.edu.tr, ali.gurkan@isik.edu.tr, burak.ertopcu@isik.edu.tr, ozan.topsakal@isik.edu.tr
berke.ozenc@isik.edu.tr, bugra.kanburoglu@isik.edu.tr, ilker.cam@isik.edu.tr
begum.avar@boun.edu.tr, gokhan.ercan@isik.edu.tr, olcaytaner@isikun.edu.tr
[1]Department of Computer Engineering, Işık University, İstanbul, Turkey
[2]Department of Linguistics, Boğaziçi University, İstanbul, Turkey

***Özetçe*—Bir kelimenin geçtiği bağlam içindeki anlamını belirlemek, doğal dil işleme alanında, zorlu ve çokça uygulaması olan bir problemdir. Bu problemin literatürdeki bilinen adı, kelime belirsizlik gidermedir. Bir çok yayın İngilizce dili ve verileri üzerine yoğunlaşmış çalışmalardır. Bu çalışmada kullandığımız veri kümesi, Penn Treebank Corpus'dan derlenmiş ve Türkçe'ye çevirilmiş 1400 cümleden oluşmaktadır. Çalışmamızın amacı, 6 farklı öznitelik çıkarım algoritmasının performanslarını farklı sınıflandırıcılarla ölçmektir. Kullandığımız sınıflandırma algoritmaları; C4.5, Random Forests, Rocchio, Naive Bayes, KNN, Linear ve multilayer Perceptron'dır. Yayınımızın amacı, açıklanan özniteliklerin morfolojik açıdan zengin olan bir dilde (Türkçe), farklı sınıflandırıcılarla verdiği performansı ölçmektir.**

***Anahtar Sözcükler*—*Kelime belirsizlik giderme; Türkçe Penn-Treebank corpus***

*Abstract*—Identifying the sense of a word within a context is a challenging problem and has many applications in natural language processing. This assignment problem is called word sense disambiguation (WSD). Many papers in the literature focus on English language and data. Our dataset consists of 1400 sentences translated to Turkish from the Penn Treebank Corpus. This paper seeks to address and discuss 6 different feature extraction methods and its classification performances using C4.5, Random Forests, Rocchio, Naive Bayes, KNN, Linear and multilayer Perceptron. This paper calls into question how the described features perform on a morphologically rich language (Turkish) with several classifiers.

*Keywords—Word sense disambiguation; Turkish Penn-Treebank corpus*

## I. INTRODUCTION

NLP (Natural Language Processing) is a component of Artificial Intelligence and investigates the capability of a computer program or system to understand and resolve human speech and textual documents. The development process of NLP applications is challenging due to the requirement of structured, unambiguous voice commands and texts in many different languages that the computer program tries to understand. The problem arises due to the fact that, in general, human speech is not unambiguous and that linguistic structure can vary depending on numerous complex variables, including pragmatic and sociolinguistic factors.

Current perspectives on NLP are machine learning-based, that is a type of artificial intelligence that examines and uses patterns in the to develop the system's own understanding.

In addition, the term 'Natural Language Processing' is not always used in one way. It sometimes includes signal processing or speech recognition, context reference issues, and discourse planning and generation, as well as syntactic and semantic analysis and processing. The other usage is more narrow and only includes syntactic and semantic analysis and processing.

The interpretation of languages that NLP systems work are examined in several levels which can be listed as "Mor-phological Level", "Lexical Level", "Syntactic Level", "Se-mantic Level", "Discourse Level", "Pragmatic Level"; and common NLP tasks are "Sentence Segmentation, POS (Part-of-speech) tagging and parsing", "Shallow Parsing", "Named Entity Recognition", and "Word Sense Disambiguation". This study focuses on Word Sense Disambiguation task at the the "Semantic Level" that focuses on how the context of words within a sentence helps to determine the meaning of words at an individual level.

NLP systems capture the meaning from an input of words within a text (sentences, paragraphs, pages etc.) and transform it into a structured output. This operation is generally problematic due to linguistic complexities and ambiguous senses of input words. The ultimate goal of NLP is for computer systems to achieve a human-like understanding of text/speech/language. When this goal is reached, computer systems can be aware of summarizing, translating and generating natural human text and language.

Senses are the meaning(s) of the words located in a dictionary (or rather the lexicon) of a specific language. There can be a one-to-one-mapping between the word and its sense but also, there can be multiple meanings for a single word. By that reason, in order to make reasonable outputs, one of the senses of each word in a text or speech should be selected properly to prevent confusion of the system for interpreting the linguistic data. "Word Sense Disambiguation" (WSD) is one of the common tasks of NLP that is trying to address this

978-1-5386-0930-9/17/$31.00 ©2017 IEEE

issue by using various approaches and implementations.

In this research, we generated seven different models in order to apply Word Sense Disambiguation task on our dataset that consists of 1400 sentences and 13187 words. We created these models by using a machine learning approach. After that, we compared the accuracy rates of these models to obtain the most accurate model in order to optimize the sense selection for each word onto the sentences in the dataset.

This paper is organized as follows: We define WSD problem in Section II and give the previous work in Section III. In continuous models, we represent words with continuous vectors, namely word embeddings. A brief introduction to word embeddings is given in Section IV. The details of our dataset and how it is constructed are given in Section V. We give our experiment methodology in Section VI and results in Section VII. Lastly, we conclude in Section VIII.

## II. WORD SENSE DISAMBIGUATION

The task of choosing the correct sense for a word is called word sense disambiguation (WSD). WSD algorithms take an input word w in its context with a fixed set of potential word senses $S_w$ of that input word and produce an output chosen from $S_w$. In the isolated WSD task, one usually uses the set of senses from a dictionary or theasurus like WordNet. Table I shows an example for the word 'yüz', which can refer to the number '100', to the verb 'swim' or to the noun 'face'.

Table I. POSSIBLE DEFINITIONS FOR THE SENSE TAGS FOR YÜZ

| Sense | Definition | Example |
|---|---|---|
| $yüz^1$ | Doksan dokuzdan sonra gelen sayı | **Yüz** kişi toplandı. |
| $yüz^2$ | Su içinde ilerlemek, durmak | Dün Altınoluk'ta **yüzdüm**. |
| $yüz^3$ | Sima, çehre, surat | Bu sanatçının **yüzü** çok güzel. |

In the literature, there are actually two variants of the generic WSD task. In the lexical sample task, a small selected set of target words is chosen, along with a set of senses for each target word. For each target word w, a number of corpus sentences (context sentences) are manually labeled with the correct sense of w. In all-words task, systems are given entire sentences and a lexicon with the set of senses for each word in each sentence. Annotators are then asked to disambiguate every word in the text.

In all-words WSD, a classifier is trained to label the words in the text with their set of potential word senses. After giving the sense labels to the words in our training data, the next step is to select a group of features to discriminate different senses for each input word. Table II shows the sample text represented with sense labels and three possible features, namely the root form of the word, the part of speech (POS) tag of the word, and a boolean feature for checking the capital case.

Given such a training data, a classifier like neural network or decision tree can be trained to label new words. Figure 1 shows the operation of such a classifier at the point where the word *yüzdükten* is next to be labeled. For this classifier, the window size is 2, that is, we assume a context window that includes two preceding words and two succeeding words.

Table II. ALL-WORDS WSD AS A CLASSIFICATION PROBLEM

| Word | Features | | | | Label |
|---|---|---|---|---|---|
| | Root | Pos | Capital | ... | |
| Yüzündeki | yüz | Noun | True | ... | $yüz^3$ |
| ketçap | ketçap | Noun | False | ... | $ketçap^1$ |
| lekesi | leke | Noun | False | ... | $leke^2$ |
| yüzdükten | yüz | Verb | False | ... | $yüz^2$ |
| sonra | sonra | PCAbl | False | ... | $sonra^1$ |
| çıkmış | çık | Verb | False | ... | $çık^{10}$ |
| . | . | Punctuation | False | ... | $.^1$ |

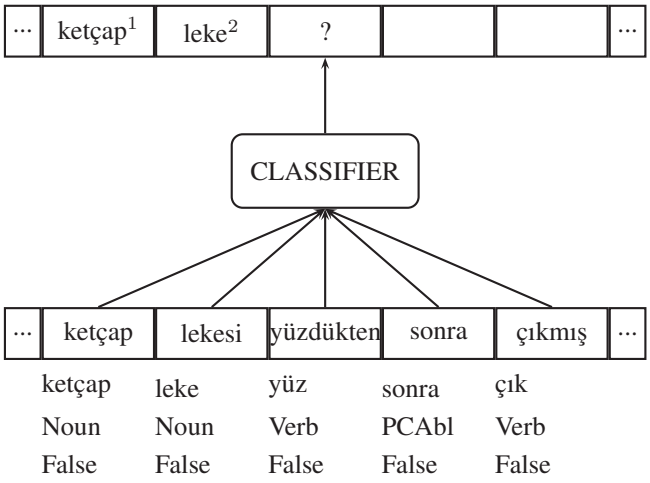


Figure 1. Classifier based approach to all-words WSD. The tagger slides a context window over the sentence, classifying words as it proceeds. At this point, the classifier is attempting to label *yüzdükten*. Features derived from the context typically include the words, POS tags, etc.

## III. PREVIOUS WORK

### A. Linguistic Background

One of the central foci of literature on Turkish linguistics has been the complexity of Turkish morphology. Turkish is a textbook example for an agglutinative language, i.e. words in their surface form may contain various morphemes, especially suffixes, each of which have a semantic and/or syntactic contribution to the sentential meaning.

In general, language in its written form gives us orthographic clues about word boundaries but not about the internal makeup of words in their surface form. Moreover, orthography is misleading when multi-word expressions are taken into consideration. In Turkish, the problem of parsing and disambiguation is even more advanced, as not only content words in their base forms but also functional morphemes may be the source of ambiguity which cannot be resolved without the context.

The presence of an excessive amount of suffixes in the Turkish lexicon is closely related to the problem of the storage vs. computation trade-off. In the linguistics literature, there are several approaches to the size of the mental lexicon - the load on memory - and the amount of work to be performed by the computational mechanism - the burden on the 'processor' -. When the parallels among the human brain and computers were drawn at early stages of developments in computer science and linguistics, it was believed by most linguists that human mind has a limited storage and an efficient compu-

tational mechanism. Only morphemes, i.e. morphologically simplex items such as roots or affixes, were assumed to be stored in lexicon [5]. Developments in computer science made linguists question this belief and most scientists nowadays hold the view that some morphologically complex items are stored rather than assembled in the brain. Therefore, in an agglutinative language like Turkish, considering some morphologically complex word forms as inseparable chunks may help us reduce the rate of ambiguity.

Not only the morphology but also the syntactic structure of Turkish is a fertile source for disambiguity. Turkish is typically treated as a subject-object-verb (SOV) language yet it has a rather free word order. A particular order is chosen mainly on pragmatic grounds [9]. The problem arises due to the fact that, in predicting the correct sense of a given word, WSD needs to take into account the frequency of neighboring words, i.e. to use the word-order as a clue for disambiguation.

Another issue in Turkish syntax that leads to further complications for WSD is related to the fact that Turkish is a pro-drop language, i.e. pronouns may be omitted when they are predictable from the pragmatic and/or grammatical context. This property of Turkish leads to cases where the antecedent of an agreement relationship (in terms of person and number) is missing on the surface and can only be retrieved through context.

In short, many linguistic features in Turkish, such as in which sense a word is used, what the functions of the affixes are, word-order and the antecedent of some agreement relationships are predictable through discourse only. Thus, it is safe to assume that Turkish is a challenging case for WSD tasks.

### *B. Computational Background*

WSD has some weaknesses about selectable differences between ambiguous words, feature selection, algorithms and evaluation criteria. In their paper, Orhan and Altan investigate feature selection strategies for word sense disambiguation task in Turkish. The paper explains that Turkish verbs can be affected by many different factors on sense disambiguation process. According to the results, selecting the correct and effective features are more important than the algorithms. Therefore, increasing the size of the feature set does not directly affect the performance due to some features being irrelevant [7].

İgen et al. investigate the effect of different windowing schemes on word sense disambiguation accuracy in Turkish language. They use Turkish lexical sample dataset in their experiments. They run the experiments with different window sizes on different machine learning algorithms. According to their studies, Naive Bayes and Functional Tree have better accuracy results with +-5 window size for verbs and nouns than other algorithms. Finally, it can be argued that smaller window scores are more effective for the disambiguation process. Best performing window size on average is 5 on Turkish WSD data [12].

Altintas et al. look into the effects of windowing on success rates in WSD. Authors use a modified version of Maximum Relatedness Disambiguation algorithm to test the performance of several weighting functions. According to their results, the accuracy of WSD with this approach is increased to 4.24. Therefore, distance of neighboring words is a significant factor for WSD [2].

Ilgen et al. aim to find out the best sets of collocational features for WSD in Turkish language. They use a lexical dataset which includes polysemous nouns and verbs. The different feature sets are tested by several machine learning algorithms (Naïve Bayes, IBk, Star, J48 and FT) to find the most effective features. According to the results, preceding and following words of the target word are more effective than other words in the sentence, by increasing the accuracy performance of both verbs and nouns [11].

## IV. WORD EMBEDDINGS

Distributed representations (DR) of words (i.e., word embeddings) are used to capture semantic and syntactic regularities of the language by analyzing distributions of word relations within the textual data. Modeling methods generating DRs rely on the assumption that “words that occur in similar contexts tend to have similar meanings” (distributional hypothesis) which stems from the nature of language itself. Due to their unsupervised nature, these modeling methods do not require any human judgement input to train, which allows researchers to train very large datasets in relatively low costs.

SkipGram is a prediction based distributional semantic model (DSM) consisting of a shallow neural network architecture inspired from neural language modeling (LM) intuitions. It is commonly known for its open-source implementation library word2vec. SkipGram acts like a log-linear classifier maximizing the prediction of the surrounding words of a word within a context (center window). Probabilistic word and sentence prediction by local neighbors of a word has been successfully applied on LM tasks under Markov assumption. SkipGram leverages the same idea by considering the words within the window as positive and negative instances and learning weights (for k contexts) which maximizes word predictions. In the training process, each word vector starts as a random vector, and then iteratively shifts to the neighboring vector.

## V. DATA

We used Penn Treebank corpus as our dataset which is gathered from the web, and each sentence was translated to Turkish [10]. Then, 1400 randomly chosen sentences from the dataset, containing 13194 words, were manually annotated.

### *A. Morphological Disambiguation*

Turkish is an agglutinative language in which words are formed by attaching derivational and inflectional suffixes to the roots. Morphemes added to a word can change its part of speech, i.e., for instance, convert a noun to a verb - or vice-versa - or can create adverbs from adjectives. Moreover, during word formation, some letters can change or undergo deletion. Hence, without determining the lemma of a word from its surface form, based on its intended meaning, it is not possible to identify the word correctly and extract candidate senses from a dictionary.

Following the translation, the corpus has been morpholog-ically disambiguated. In that work, human annotators selected

the correct morphological parsing among multiple possible analyses returned from the automatic parser (See Figure 2 for the morphological disambiguation tool used). The tag set and morphological representation were adopted from the study [6]. Each output of the parser comprises the root of the word, its part-of-speech tag and a set of morphemes, each separated with a "+" sign.

Figure 2. Morphological disambiguation tool

### B. Word Sense Annotation

In the data, there are 13194 class instances, 3127 of which are distinct. 57 items were eliminated due to some errors in the data. The most frequently used class labels, their counts and meanings are given in (Table III). There is no transposed sentences in dataset. In the annotation, we used the NLP Tool of I¸sık University.

Table III. THE MOST USED CLASS LABELS

| Label | Count | Sense |
|---|---|---|
| TUR10-1081860 | 1282 | nokta |
| TUR10-0000000 | 987 | özel isim |
| TUR10-0820240 | 418 | virgül |
| TUR10-0775480 | 268 | tırnak işareti |
| TUR10-0000010 | 235 | bir tam sayı |
| TUR10-0105580 | 171 | herhangi bir varlığı belirsiz olarak gösteren sayı |
| TUR10-0816400 | 159 | bir bağ olduğunu anlatan bir söz |
| TUR10-0000020 | 157 | reel sayı |
| TUR10-0178920 | 146 | da / de |
| TUR10-0120760 | 92 | bu |

In the annotation screen, there are some possible meanings under each word of each sentence. For tagging the words, the person (tagger) needs to choose the appropriate meaning for each word considering the items from combobox as illustrated in (Figure 3). When the tagger presses the "next" button, his/her annotation is saved and the tagger can pass to the next one to annotate it. The sentences were annotated by 7 people (200 sentences per person).


Figure 3. Annotation screen for WSD

The data was stored in text file as text format, and it consists of some parts which are morphological analysis, named entity recognition and shallow parsing as shown in (Figure 4).

Annotated dataset and source codes are freely accessible[1].

## VI. METHODOLOGY

The eight known classification algorithms we use, are:

[1]http://haydut.isikun.edu.tr/nlptoolkit.html

2501.train ×

```
{turkish=Ama}{morphologicalAnalysis=ama+CONJ}{metaMorphemes=ama}
{semantics=TUR10-0031730}{namedEntity=NONE}{propbank=NONE}
{shallowParse=HİÇBİRİ} {turkish=reklam}{morphologicalAnalysis=reklam
+NOUN+A3SG+PNON+NOM}{metaMorphemes=reklam}{semantics=TUR10-0644960}
{namedEntity=NONE}{propbank=NONE}{shallowParse=ÖZNE} {turkish=geliri}
{morphologicalAnalysis=gelir+NOUN+A3SG+P3SG+NOM}{metaMorphemes=gelir
+sH}{semantics=TUR10-0289950}{namedEntity=NONE}{propbank=NONE}
{shallowParse=ÖZNE} {turkish=yetersiz}{morphologicalAnalysis=yetersiz
+ADJ}{metaMorphemes=yetersiz}{semantics=TUR10-0852240}
{namedEntity=NONE}{propbank=NONE}{shallowParse=YÜKLEM} {turkish=.}
{morphologicalAnalysis=.+PUNC}{metaMorphemes=.}
{semantics=TUR10-1081860}{namedEntity=NONE}{propbank=NONE}
{shallowParse=HİÇBİRİ}
```

Figure 4. Data content as text file

- Dummy: All test instances are assigned to the class with the maximum prior.
- C45: The archetypal decision tree method [8].
- Knn: K-Nearest Neighbor classification algorithm that uses the Euclidean distance.
- Rocchio: Nearest-mean classification algorithm that uses the Euclidean distance.
- Lp: Linear perceptron with softmax outputs trained by gradient-descent to minimize cross-entropy [3].
- Mlp: Well-known multilayer perceptron classification algorithm [3].
- Nb: Classic Naive Bayes classifier where each feature is assumed to be Gaussian distributed [1] and each feature is independent from other features.
- Rf: Random Forest method improves bagging idea with randomizing features at each decision node [4] and called these random decision trees as weak learners. In the prediction time, these weak learners are combined using committee-based procedures.

### A. Discrete Model

In this section, first, features that are used in this work are explained in detail. Then, six different feature sets and the learning algorithms employed are provided and feature set - learning algorithm pairs are stated.

*1) Features:* There are several features that are used in this work. There are different feature sets created from these features. In this section, all features, regardless of the feature sets, will be explained in detail.

- CaseAttribute (C): This is a discrete attribute for a given word. If the last inflectional group of the word contains case information, the attribute will have that case value. Otherwise the attribute will have the value null.
- Is Capital Letter (IC): This a binary valued feature. Its value returns true if the word starts with a capital letter, and false otherwise. Words starting with a capital letter are proper nouns unless the word is not the first word in the sentence. Being proper noun is important for meaning because proper nouns do not have specific meanings.

- LastIGContainsPossessiveAttribute (LIP): This is a binary attribute for a given word. If the last inflectional group of the word contains possessive information, the attribute will be true, otherwise it will be false.
- Main POS (MP): This is a discrete valued feature. The value of the feature consists of the main part-of-speech (POS) tag of the word.
- NerTag (NT): This is a discrete valued feature, which finds the NER (Named Entity Recognition) tag of the word. This is a strong feature especially for detecting subjects and objects in the sentence. For Instance, if the word has a "PERSON" NER tag, it is generally the subject or the object of the sentence.
- Root Form (RF): This is a discrete valued feature and stores the word's root as its value. A root is a word without any affixes. A word takes its meaning mainly from the root. Therefore, this features works well for words having different roots. But when it comes to different surface forms created from the same root, it fails. Both the surface form and the root of a word are directly connected to the sense of the word and these features are crucial for WSD.
- RootPos Attribute (RP): This attribute is a discrete attribute for a given word. It returns the part of speech of the root word.
- Surface Form (SF): This is a discrete valued feature and stores the word's itself as its value. It is a strong feature because the meaning of a word's root can be modified through affixation. In a word's surface form, the word is considered as a whole with all the affixes attached, and, when it is used as a feature, it matches with the corresponding meaning.
- ShallowTag (ST): This is a discrete valued feature, which finds the Shallow Parse tag of the word.
- Is Percentage (IP): This is a discrete valued feature. It checks each character in the string. If the character equals to "%", it returns true, otherwise false.

*2) Methods:* We give the details of the methods, such as the attributes, the classifier and its parameters, in Table IV.

### B. Continuous Model

In our continuous model experiments, we learnt continuous word features (vectors in C dimensions) from 1 million (M) word news corpus by using Mikolov et al.'s SkipGram and CBOW DSMs in an unsupervised fashion. We used our news corpus in three separate sizes (100K, 500K, 1M) in order to measure effect of the corpus size to the model performances. Hyper-parameters of the SkipGram and CBOW models we used are listed below along with their default values:

- Context window (win): Word context window size where co-occurrence information is gathered. Default is 5.
- Dimension (d): Size of a C context dimensions of vectors for word-context dense matrices (W × C). Default size is 100.
- Deleting infrequent words (del): A threshold number for model to exclude words where their frequencies are below the value. Default setting is no threshold (0).

We kept out model case sensitive and did not exclude punctuations. Distributions of punctuations are also learnt along with the words. Thus, every punctuation type is also represented by a C dimensional real valued vector where its context is learnt from its neighboring words. Model's ability to represent punctuation-word relations could provide extra information to classifiers for WSD task. For an example, model represents question mark "?" with a vector which is spatially very close to the vectors of the common question words in Turkish (e.g., "ne", "nasıl", "niye", "kim").

We trained our DR models in two kinds of word forms:

- Surface Form (SU): Natural form of a word which appeared in a text as it is. Ex: *Kitapları kim sevmez?*
- Root Form (RO): Lexical root of a word gathered by morphological disambiguation in the sentence level. Ex: *Kitap kim sev?*

## VII. EXPERIMENTS

### A. Inter-annotator Agreement

For evaluation of the annotated dataset, we used inter-annotator-agreement measure. Two different group of annotators annotated same sentences. Due to a lack of time, we could only re-annotate (by a different annotator) 100 of the total of 1400 sentences and got %78.5 of inter-annotator aggrement. As a comparison, the expected inter-annotator agreement, assuming the annotators annotated completely randomly, is %54.5.

### B. Discrete Model

In this study, we experimented on 6 different methods with stratified 10-fold cross-validation. The obtained error rates are given in Table V.

### C. Continuous Model

As a continuous model baseline, we set SkipGram (SG) hyper-parameters to win=5, d=100, del=0 (SG-win5-d100-del0). We trained 10 models in total with various configurations (9 SkipGram and 1 CBOW). We grouped our experiments in 3 for each corpus sizes (100K, 500K, 1M), and for 2 different word forms (Surface, Root). We ran our models with Dummy, Lp with learning rate: 0.1, Mlp with learning rate 0.1, hidden nodes = 50, Knn with k = 3, and Nb classifiers with 10-fold cross-validation on data. We ran all classifiers with the window size (w) of 1 except one (1M-RO-w2). Table VI shows classifier error rates and OOV percentages of our experiments:

Our continuous model experiments show that:

- Continuous word representation trained from unsupervised DR models can provide valuable information as features of words for WSD classification. Although DR model continuous features did not perform well

Table IV. DETAILS OF METHODS USED IN DISCRETE MODEL

| Method | Attributes | Window | Classifier | Parameters | Author |
|---|---|---|---|---|---|
| METHOD1 | IC, MP, RF, SF | 3 | Rocchio | | B.Ö. |
| METHOD2 | IC, MP, NT, RF, SF, ST | 1 | Rf | | B.E. |
| METHOD3 | C, IC, LIP, MP, RF, RP, SF | 1 | Knn | $k$ = 1 | A.B.K. |
| METHOD4 | IC, LIP, MP, RF, SF | 1 | Mlp | learning rate = 0.1, number of hidden nodes = 5, 8 | O.T. |
| METHOD5 | C, IP, LIP, MP, RP | 5 | Mlp | learning rate = 0.1, number of hidden nodes = 10 | A.T.G. |
| METHOD6 | IC, MP, RF | 1 | C45 | prune = true | O.A. |

Table V. ERROR RATES OF METHODS USING STRATIFIED 10-FOLD CROSS-VALIDATION

| Classifier | DUMMY | METHOD1 | METHOD2 | METHOD3 | METHOD4 | METHOD5 | METHOD6 |
|---|---|---|---|---|---|---|---|
| Error Rate | 29.42 | 26.44 | 21.65 | 29.13 | 25.14 | 21.34 | 20.37 |

Table VI. ERROR RATES OF DIFFERENT METHODS USING 10-FOLD STRATIFIED CROSS-VALIDATION OF CONTINUOUS MODEL EXPERIMENTS

| | OOV% | Dummy | Lp | Mlp | Nb | Knn |
|---|---|---|---|---|---|---|
| SURFACE (SU) | | | | | | |
| 100K | 32.35 | 33.19 | 33.16 | 33.37 | 39.22 | 32.68 |
| 500K | 22.97 | 33.1 | 32.45 | 32.97 | 37.03 | 32.03 |
| 1M | 20.22 | 32.59 | 32.23 | 32.85 | 37.17 | 32.16 |
| ROOT (RO) | | | | | | |
| Discrete Baseline | - | 29.41 | 28.83 | 29.63 | 35.65 | 28.1 |
| 100K | 20.65 | 32.76 | 32.58 | 33.02 | 41.32 | 31.68 |
| 500K | 17.37 | 32.24 | 31.13 | 32.28 | 40.52 | 31.06 |
| 1M | 16.22 | 32.17 | 31.1 | 31.67 | 40.26 | 30.6 |
| OTHER CONF. | | | | | | |
| 1M-RO-CBOW | 16.22 | 32.17 | 32.14 | 31.9 | 40.26 | .30.58 |
| 1M-RO-d300 | 16.22 | 32.17 | 31.11 | 31.78 | .40.09 | 30.57 |
| 1M-RO-d20 | 16.22 | 32.17 | 31.57 | 31.88 | 40.19 | 30.63 |
| 1M-RO-w2 | 23.83 | 32.37 | 31.86 | 32.34 | 40.33 | 31.11 |
| MIN | 16.22 | 29.41 | 28.83 | 29.63 | 35.65 | 28.1 |

standalone as manually handcrafted discrete features, they can potentially increase performance of existing models when used along with other discrete features.

- Compared to discrete single attribute baselines, almost all continuous models performed worse on all classifiers.
- Word embeddings with basic morphological root form enrichment did not provide any significant gain than simple surface form word embeddings for a WSD task.
- Increasing corpus size did not affect WSD classification task performances.
- SkipGram performed better than CBOW model almost on all experiments.
- Increasing or decreasing vector dimensions (d) of word embeddings did not affect WSD performances significantly.
- Increasing window size did not increase model per-formances.

## VIII. CONCLUSION

Word Sense Disambiguation is one of the common tasks in NLP. In this paper, we presented our studies on WSD in Turkish by using variate approaches and implementations, and by comparing the performances of various learning algorithms for WSD using classifiers like DecisionTree (C45), Random Forest, Rocchio, Naive Bayes, Knn, Linear Perceptron, and Multilayer Perceptron.

We also used Mikolov et al.'s SkipGram and CBOW models for learning continuous representation of words. Then we used our trained continuous representation of words to feed our classifiers as continuous features of words without doing any feature enhancement. Our results show that, these continuous models for WSD classification tasks can perform as good as discrete models with the same input.

## ACKNOWLEDGEMENTS

This work was supported by I¸sık University BAP projects 14B206 and 15B201. All authors contributed equally to this work. O. A., ˙I. Ç., B. E., A. T. G., A. B. K., B. Ö., O. T. designed and implemented discrete model experiments. They also labeled the data and wrote the manuscript. G. E. designed and implemented continuous model experiments. B. A. wrote the manuscript. O. T. Y. supervised the project, gave conceptual advice and wrote the manuscript.

## REFERENCES

[1] E. Alpaydın, *Introduction to Machine Learning*, 3rd ed. The MIT Press, 2010.

[2] E. Altıntas¸, E. Karslıgil, and V. Cos¸kun, "The effect of windowing in word sense disambiguation," in *Computer and Information Sciences*, vol. 3733, 2005.

[3] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[5] D. Mackay, "Lexical insertion, inflection, and derivation: creative processes in word production," *Journal of Psycholinguistic Research*, vol. 8, pp. 477–498, 1979.

[6] K. Oflazer, B. Say, D. Hakkani-Tür, and G. Tür, "Building a Turkish treebank," in *Building and exploiting syntactically-annotated corpora*. Dordrecht: Kluwer, 2003.

[7] Z. Orhan and Z. Altan, "Impact of feature selection for corpus-based wsd in Turkish," in *MICAI 2006: Advances in Artificial Intelligence*, vol. 4293, 2006.

[8] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Meteo, CA: Morgan Kaufmann, 1993.

[9] E. E. Taylan, *The function of word order in Turkish Grammar*. Berkeley: University of California Press, 1984.

[10] O. T. Yildiz, E. Solak, O. Gorgun, and R. Ehsani, "Constructing a Turkish-English parallel treebank," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 112–117.

[11] B. ˙Ilgen, E. Adalı, and A. C. Tantug˘, "The impact of collocational features in Turkish word sense disambiguation," in *16th International Conference on Intelligent Engineering Systems*, 2012.

[12] B. ˙Ilgen, E. Adalı, and A. Tantug˘, "A comparative study to determine the effective window size of Turkish word sense disambiguation systems," in *Information Sciences and Systems*, vol. 264, 2013.