Cogent Engineering

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/oaen20

MS-TR: A Morphologically enriched sentiment Treebank and recursive deep models for compositional semantics in Turkish

Sultan Zeybek, Ebubekir Koç & Aydın Seçer |

To cite this article: Sultan Zeybek, Ebubekir Koç & Aydın Seçer | (2021) MS-TR: A Morphologically enriched sentiment Treebank and recursive deep models for compositional semantics in Turkish, Cogent Engineering, 8:1, 1893621, DOI: 10.1080/23311916.2021.1893621

To link to this article: https://doi.org/10.1080/23311916.2021.1893621

Published online: 26 Apr 2021.

cogent engineering

Received: 21 November 2020
Accepted: 29 January 2021

*Corresponding author: Sultan Zeybek, Dept. of Computer Engineering Fatih Sultan Mehmet Vakif University Beyoglu, Istanbul, Turkey
E-mail: szeybek@fsm.edu.tr

Reviewing editor:
Duc Pham, School of Mechanical Engineering, University of Birmingham, Birmingham, UK

Additional information is available at the end of the article

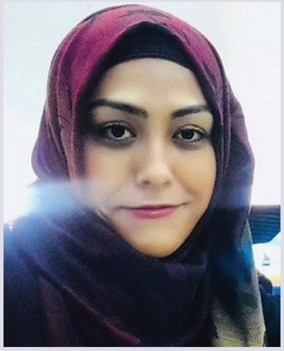
COMPUTER SCIENCE | RESEARCH ARTICLE

MS-TR: A Morphologically enriched sentiment Treebank and recursive deep models for compositional semantics in Turkish

Sultan Zeybek[1]*, Ebubekir Koç[2] and Aydın Seçer[3]

Abstract: Recursive Deep Models have been used as powerful models to learn compositional representations of text for many natural language processing tasks. However, they require structured input (i.e. sentiment treebank) to encode sentences based on their tree-based structure to enable them to learn latent semantics of words using recursive composition functions. In this paper, we present our contributions and efforts for the Turkish Sentiment Treebank construction. We introduce MS-TR, a Morphologically Enriched Sentiment Treebank, which was implemented for training Recursive Deep Models to address compositional sentiment analysis for Turkish, which is one of the well-known Morphologically Rich Language (MRL). We propose a semi-supervised automatic annotation, as a distant-supervision approach, using morphological features of words to infer the polarity of the inner nodes of MS-TR as positive and negative. The proposed annotation model has four different annotation levels: morph-level, stem-level, token-level, and review-level. Each annotation level's contribution was tested using three different domain datasets, including product reviews, movie reviews, and the Turkish Natural Corpus essays. Comparative results were obtained with the Recursive Neural Tensor

ABOUT THE AUTHOR



Sultan Zeybek

Sultan Zeybek has been working as a research assistant in the Department of Computer Engineering of Fatih Sultan Mehmet Vakif University. She was a visiting researcher at the University of Birmingham, College and Engineering Sciences between June 2019 and June 2020, at the Autonomous Remanufacturing (AUTOREMAN) Laboratory, which has been conducting state-of-art researches focusing on the theoretical implementations and the practical-industrial applications of the Robotic Remanufacturing. She is a member of the Bees Algorithm Research Group. Her research interests include Deep Learning Methods, Natural Language Processing, Computational Intelligence, Meta-heuristics, Combinatorial, and Continuous Optimisation Problems.

PUBLIC INTEREST STATEMENT

Sentiment analysis is a demanding research topic of natural language processing. It has a substantial potential impact on from academia's various areas to commercial applications, since it provides a strategical understanding of human feelings and opinions. Recently, many sentiment analysis researchers focused on the recursive compositional architectures for learning syntactic and semantic context since current semantic vector space models cannot capture the meaning of the longer phrases. With this motivation, this paper introduces a novel Morphologically Enriched Sentiment Treebank (MS-TR) for Turkish Sentiment Analysis to investigate the effectiveness of the recursive compositional deep learning models for morphologically rich languages (MRLs). To this end, Recursive deep learning models (Tree-RNNs) have been employed using composite functions over binary and fine-grained annotated parse trees of MS-TR, and compared with conventional machine learning algorithms. Benchmark results proved the effectiveness of deep recursive architectures.

Networks (RNTN) model which is operated over MS-TR, and conventional machine learning methods. Experiments proved that RNTN outperformed the baseline methods and achieved much better accuracy results compared to the baseline methods, which cannot accurately capture the aggregated sentiment information.

Subjects: Artificial Intelligence; Human Computer Intelligence; Computer Engineering

Keywords: Recursive neural networks; sentiment analysis; sentiment treebank; opinion mining; morphologically rich languages

1. Introduction

Sentiment analysis (SA) is a demanding research topic of natural language processing (NLP). It has a substantial potential impact on from academia's various areas to commercial applications since processing of different data sources from social media, movie reviews, and product reviews provides a strategical understanding of human feelings and opinions.

Sentiment analysis is based on detecting the polarity of a given text, which is generally classified as a positive, objective or negative thinking about specific domains, subjects or items (Pang & Lee, 2008). Research on sentiment analysis growing very fast with the increased interest and available resources in this field. English is the richest language in terms of availability of resources. Most sentiment analysis methods and algorithms have been first implemented for English, and{ and NLP resources, such as polarity lexicons (Cambria et al., 2012, 2014) and sentiment treebank datasets (Socher et al., 2013), parsers and other NLP tools are also essentially available for English (Oflazer, 2014). However, the accuracy level for the sentiment analysis tasks for morphologically rich languages (MRLs) is not well enough for commonly studied languages like English due to the data and source sparsity drawbacks. Hence, non-English languages should gain momentum in NLP studies, especially complementing their resources' deficiencies to reach state-of-the-art results.

Recently, sentiment analysis studies have focused on the compositional architectures for learning syntactic and semantic context since current semantic vector space models cannot capture the meaning of longer phrases (Oflazer, 2014). The recursive deep learning models have been used widely used in many NLP applications to employ composite functions over an annotated parse trees. They are tree-based (i.e. hierarchical) neural networks, which are inspired by human understanding and can learn the compositional semantics of sentence meaning by recursive compositional functions that combine the meanings of words of sub-phrases (Iyyer et al., 2014; Liu et al., 2014; Socher et al., 2012, 2010, 2011). The proposed architectures have been applied successfully for sentiment analysis task (Socher et al., 2013), language parsing (Socher et al., 2011), and political ideology detection (Iyyer et al., 2014).

Although many studies have been done for sentiment analysis in English, the research focusing on compositional models for morphologically rich languages (MRLs) is still developing. For example, current studies for Turkish have not gone far beyond the conventional machine learning approaches, and lexicon-based approaches since statistical methods cannot perform well enough for MRLs (Kaya et al., 2012). In addition, the conventional approaches have a performance bottleneck due to representing words as independent atomic units, that causes the loss of explicit information among words and reduces the accuracy.

In this work, we seek to address three main issues for Turkish SA; (i) to investigate the effectiveness of the recursive compositional models for Turkish sentiment analysis, (ii) to construct a Turkish Sentiment Treebank (a hierarchical representation of the sentences, i.e. fully labelled parse trees) to capture the semantic compositionality in a given sentence, and (iii) to contribute to the lack of sentiment analysis resources that also can be used for the other recursive deep models for future studies.

To this end, we introduced our contributions and efforts to develop a novel Morphologically Enriched Turkish Sentiment Treebank (MS-TR), which is annotated according to each word's morphological information. To the best of our knowledge, MS-TR is the first sentiment treebank that is fully labelled according to morphological structures of the Turkish words. It has been constructed to employ compositional effects of the sentiment analysis for two different subtasks: binary classification and fine-grained classification. MS-TR is constructed by using different domains, including movie reviews and multi-domain product reviews, which were annotated by Demirtas and Pechenizky (Demirtas & Pechenizkiy, 2013). The fine-grained MS-TR is constructed based on the latest updated version of BOUN Treebank (Turk et al., 2019). It was constructed by phrase-level annotation based on the morphological information of each word. The aim is investigating whether recursive neural networks can improve the classification accuracy of MRL using fully labelled parse trees of different domain dataset, which are annotated based on the morphological information. Specifically, we employ Recursive Neural Tensor Networks (RNTN) for binary and fine-grained sentiment classification over the MS-TR, since RNTN has been performed better compared to the other recursive models such as semi-supervised recursive autoencoders (RAE), matrix-vector recursive neural networks (MV-RNN) both for English (Socher et al., 2013) and for Arabic, that is also MRL like Turkish (Baly et al., 2017).

The source code for the sentiment analysis and the Turkish Sentiment Treebank dataset are available at https://tinyurl.com/y7pk4pwj. MS-TR can also be downloaded from https://data.mendeley.com/datasets/nz7vm5rchd/1 (Zeybek, 2020). The rest of the paper is organized as follows. Section 2 reviews the related work focused on Turkish sentiment analysis studies and tree-based learning models. Section 3 introducescharacteristics of Turkish language in terms of MRL, and Section 4 describes the annotation strategies for the proposed MS-TR. In Section5, RNTN architecture has been represented, and Section 6 reports the experiments employed by RNTN over MS-TR. The discussion part and conclusions have ended the proposed study.

2. Related work

Research on sentiment analysis is growingvery fast with the increased interest and available resources in this field. The accuracy level for the sentiment analysis tasks has mainly related to the availability of sentiment analysis resources and NLP tools. English is the richest language in terms of availability of resources, such as polarity lexicons (Cambria et al., 2012, 2014) and sentiment treebank datasets (Socher et al., 2013). So far, many studies have been done with the highest level of accuracy results for English sentiment analysis. However, research for morphological rich languages (MRLs) is still developing.

Recursive Neural Networks are powerful models for sentiment analysis that have already performed well in many languages for sentiment classification task, as they can capture the semantic compositionality in a given sentence (Baly et al., 2017; Socher et al., 2013). The standard version of the Recursive Neural Networks (Tree-RNNs), namely Recursive Autoencoders (RAE) was introduced by Socher et al. (Socher et al., 2011) using autoencoders in a recursive manner. A semi-supervised version of the RAE can learn phrase representations to predict sentiment distributions of the sentences with good accuracy levels; however, it is not able to capture long-phrases meanings and all types of compositions. Hence Recursive Neural Tensor Networks (RNTN) has been developed (Socher et al., 2013) to model the meaning of long-phrases by a recursive combination functions in a neural tensor network's architecture. RNTN employs fully labelled treebank, i.e. specialised tree structures to combine semantics in a bottom-up manner. Stanford Sentiment Treebank (SST) has been developed from the English classification dataset (Pang & Lee, 2005) for the binary and the fine-grained sentiment analysis. RNTN have been achieved impressive successes for sentiment analysis in English by using SST.

Dong et al. proposed an adaptive Recursive Neural Network (AdaRNN) for target-dependent sentiment analysis (Dong et al., 2014). They have been created a manually labelled twitter dataset to detect the sentiment of the given target by employing AdaRNN layer. The recursive structures were built based on the dependency parsing by employing multiple compositional functions. In terms of morphologically rich languages (MRLs), the Arabic language have been tested with Arabic Sentiment Treebank (ARSenTB) (Baly et al., 2017).

Even though compositional models and deep learning methods have been popular recently, most studies in the field of the Turkish sentiment analysis have particularly followed the two leading pathways, such as the lexicon-based (rule-based) approaches, machine learning-based approaches.

The majority of machine-learning-based studies have been used Naïve Bayes (NB), Support Vector Machines (SVM), Maximum Entropy (ME) classifiers combined with conventional feature representation methods. Erogul's thesis is the earliest Turkish sentiment analysis work based on the machine learning approach that was done in 2009 (Erogul, 2009). He built the tagged movie review dataset from the Beyazperde.com and classified them by using SVM. He investigated the effect of the stemming process and part-of-speech (POS) tags to the Turkish sentiment model performance. As a baseline, he used bag-of-words features with spellchecking and elimination process and achieved 85% F1-score. As a particular case, he focused on the different features, including using only roots of the words, parts-of-speech (POS) tags, n-grams. Using only the roots of the words as a feature decreased performance to the 83.99% F1-score. Noun + Adjective features performed better compared to the other part-of-speech tags features and achieved 83.34% F1-score. The best accuracy has been achieved as 86.16% F1-score, which has been obtained by the combination of the unigram, bigram, trigram and 4 gram.

Another remarkable study was done to classify Turkish political news by Kaya et al. (Kaya et al., 2012). They used Naïve-Bayes (NB), Support Vector Machines (SVM), Maximum Entropy (ME), and Character-based N-gram Language Model (*n*-gram LM) based on the *n*-grams, root words, adjectives and polar words features. The features are combined to represent political news in the bag-of -words (BOW) framework. Maximum Entropy and n-gram LM achieved better accuracy levels compared to the SVM and NB with the 76–77% accuracy level.

A model has been proposed for assigning polarities of the Turkish blog posts about products and services by Aytekin (Aytekin, 2013), that uses Naïve Bayes Algorithm to classify 350 positive and 350 negative comments which were collected by We Feel Fine website. A sentiment dictionary, including 4,744 synsets, was used to assign probability scores of each word in a comment. The positive and negative precision scores were measured as 72.28% and 73.14%, respectively.

Although machine learningalgorithms have reached high success rate, lexicon-based approaches are widely preferred because they are practical. These methods are simply applied by attaching a sentiment polarity score to the words or phrases based on the lexicon information. As an earlier lexicon-based work, Vural et al. proposed to use SentiStrenght as a polarity lexicon (Thelwall et al., 2012) to classify movie review dataset with unsupervised learning. They translated the SentiStrenght's polar words from English to Turkish (Vural et al., 2013) and calculated the total polarity score of the original input text to detect positive and negative polarity score. In terms of the accuracy result, their framework has reached up to 75.90% for the binary classification task.

Over the last few years, some additional efforts have been made to built Turkish polarity lexicons. Dehkharghani et al. built a polarity lexicon, namely SentiTurknet, which contains 14,795 synsets each has three-level (positive, negative, neutral) polarity score (Dehkharghani et al., 2016). The polarity scores are evaluated by complementary usage of various NLP resources, such as English Wordnet (Miller et al., 1990), Turkish Wordnet (Bilgin et al., 2004) lexicons, English SentiWordNet (Baccianella et al., 2010), English SenticNet (Cambria et al., 2012, 2014), Polar Word Set (PWS), and polarity words with Pairwise Mutual Information (PMI) polarity scores. The proposed algorithm combined manual labelling and feature extraction steps to detect the polarity of a given synset. Additional polarity features were extracted from polarity lexicons, and they classified by logistic regression (LR), neural networks, and SVM. The final estimation of the label was predicted by the combination of these three classifiers. SentiTurkNet was tested on the Turkish movie reviews from beyazperde.com and achieved between 61.3% and 66.7% accuracy for ternary sentiment classification.

In addition to the rule-based approaches based on the lexicons and polarity scores, Bag of Words (BoW) and *N*-Gram based features have become the most preferred feature modelling methods. Coban et al. (Coban et al., 2015) constructed a Twitter dataset for binary sentiment classification, and they used BOW and N-grams to extract features. The dataset annotated according to the positive and negative emoticons. Naïve Bayes, Multinom Naïve Bayes, SVM and K-Nearest Neighbourhood (k-NN) algorithms were used to classify weighted features. For all case studies, N-gram features performed better than the BOW and achieved between 62%and 66% accuracy. Similarly, the Extreme Learning Machine (ELM) was used to predict the sentiment of the tweets and customer reviews of the telecom company based on the BOW, and N-gram features compared to SVM. SVM performed better with the highest accuracy level 74%, which is not good enough again for the datasets, which has lower than 3000 entries (Ozyildirim & Coban, 2018).

Few studies have taken into account the morphological features of Turkish to detect sentiment. Yıldırım et al. (Yıldırım et al., 2014) investigated the effects of several NLP modules on the sentiment analysis of Turkish social media text. They proposed using normalisation, negation handling, morphological analysis and stemming modules, and adjectives to classify polarity into positive, negative and neutral classes by using SVM. The experiments achieved up to 79% accuracy level using normalisation and stemming features, which means that additional morphological information improved the system performance. Similarly, Turkmenoglu and Tantug (Turkmenoglu & Tantug, 2014) suggested comparing machine learning methods and lexicon-based methods for binary classification taking into account morphological features of Turkish, i.e. taking into account absence/presence suffixes to detect kind of negation in Turkish words. The SentiStreght was used as a baseline lexicon, and the sentiment score of each word is changed to the negative if the word contains negation suffixes. Additionally, the booster list was used to detect adjectives polarity score. Machine learning approach followed a similar way like the studies above, and TF-IDF was used with unigrams and bigrams features. As a classification algorithm SVM, Naïve Bayes, and Decision Trees were tested by using Twitter and Movie dataset. The lexicon-based approach combined with all modules including normalisation, negation handling, multi-word expressisons, and booster word list achieved 75.2% and 79% and accuracy levels for the Twitter and Movie dataset. Machine learning approaches reached up to 85% accuracy level by using TF-IDF and unigram and bigram features of tweets classified by SVM. SVM and Naïve Bayes achieved 89.5% accuracy level on Movie dataset with same features.

Despite the time-consuming feature engineering efforts, the accuracy level of the proposed models are still not good enough and it is necessary to improve more efficient ways. Notably, for the Turkish language, many methodologies which are successfully applied to the English language are waiting to be explored to handle the challenging nature of Turkish sentiment analysis.

3. Turkish and its challenges for sentiment analysis and solutions

In this section, we first explain the difficulties of sensitivity analysis of Turkish language from the view of morphologic richness. Additionally, we introduce the Morphologically Enriched Turkish Sentiment Treebank (MS-TR) as a first fully labelled Turkish sentiment treebank, including the binary labelled and the fine-grained labelled parse trees to employ compositional models for Turkish sentiment analysis.

3.1. Turkish as a morphologically rich language

Turkish is an agglutinative, morphologically rich language (MRL) from the Turkic family of Altaic languages. Due to its linguistic structure, Turkish has various additional difficulties in sentiment analysis. According to the world atlas of language database (Haspelmath et al., 2005), Turkish is presented as an outlier language in terms of Eurasia language's standard synthesis level. According to this, Turkish has high synthesis level, up to eight or nine categories per word on average, while other Eurasia languages have relatively low synthesis degrees (between four and five) (Bickel & Nichols, 2005). It means that one word in Turkish can take over 30 inflexions (Cotterell et al., 2016) and words are formed by adding many inflectional and derivational suffixes to the morphemes (roots) like "beads-one-a-string". These inflectional groups can completely change meaning and polarity of the word, and sometimes one word can correspond to almost a sentence in English such as in Table 1.

In an earlier work, Dehkharghani et al. 2017) pointed out the additive structure of the Turkish words, and mentioned the challenging points of the Turkish sentiment analysis. The morphological analysis of the Turkish word Turkish words is essential since it may provide sentiment information of word. For example, the polarity of the word "sabır" (patience) is changed with particular derivational suffixes, such as "-lı" and "-sız". The word "sabır-lı" (patient) has positive polarity, and "sabır-sız" (impatient) has negative polarity.

Table 1. Examples of Turkish verb "gel" with derivational and inflectional suffixes construct different sentences in English

Root: gel (come in English)	
Word Formations in Turkish	Word Formations in English
geliyorum	I am coming
geliyorsun	You are coming
geliyor	He/she/it coming
geliyoruz	We are coming
geliyorsunuz	You are coming
geliyorlar	They are coming
geldik	We came
geldikçe	As long as (somebody) comes
gelmeliyiz	We must come
gelmeden	Without coming
gelmen	Your coming
gelirken	While (somebody) is coming
gelince	When (somebody) comes
geldirmek	To cause (somebody) to cause (another person) to come
geldirttirtmek	To cause (somebody) to cause (some other person) to cause (another person) to come

Morphological features of the Turkish words are very important for accurate sentiment analysis since even suffixes of the Turkish words can have the polarity information. Particularly, negation suffixes, that are embedded within the word are crucial for the detection of the negative meaning of the word. In this study, we propose to annotate MS-TR by using the morphological features of the words to explore sentiment that hides behind Inflectional Groups (IG) of words. We parse words to the root of the word and the possible suffixes to find morphological features. We consider the following cases for a binary level annotation:

Case 1: In Turkish, the suffix -mA and its versions are used to negate the verbs. For example, the suffix -me changes the sentiment of word from positive meaning such as “sevdim” (I like it) to the negative such as “sevmedim” (I did not like it). In addition, Turkish auxiliary verbs “etmek” (to do, to make) and “olmak” (to be, to become) are used with the suffix -mA to negate the verb phrases. As it can be seen in Table 2(a), the verb “sabretmemek” was annotated as negative since it has morphological features sabret+Verb+**Neg**^DB+Noun+Inf1+ A3sg contains “Neg” keyword.

Case 2: Besides verbal negation, the adjectives, which are transformed from the noun using absence suffixes (-lı/-li) and presence suffixes (-sız/siz) can give polarity information. The suffix -lı and its versions transform names into positive adjectives, and the suffix -sız and its versions transform names into negative adjectives. For example, the name “sabır” transforms to the positive adjective with the suffix -lı, such as “sabırlı” (patient). The suffix -sız (without) negates adjectives and converts its positive meaning to the negative meaning, such as “sabırsız” (impatient). We detect the absence and presence by considering the keywords (Without/With) in the morphological analysis of the words. Since one of the morphological analysis of “sabırlı” (sabır +Noun+A3sg^DB+Adj+With) contains the keyword “With”, sabırlı annotated as positive. Similarly, “sabırsız” is annotated as negative since its morphological analysis contains “Without” (sabır +Noun+A3sgDB+Adj+Without).

Case 3: The conjunction “değil” (is/are not) is used with adjectives as a negation marker. For example, “uzun değil” means “not long” gives a negative meaning to the “uzun” (long) as an adjective. Similarly, “var” (there is) and “yok” (there is not) are also used as a presence and

Table 2. Examples of the morphological analysis of words and detecting negation within words (a). Polar words which don’t have polar suffixes (b)

(a) Word/Phrases	**(b) Word/Phrases**
Possible Morphological Analysis	**Polarity (P:Positive N:Negative)**
sabretmek (to be patient)	kazanan (winner) (P)
sabret+Verb^DB+Noun+Inf1+ A3sg	kaybeden (loser) (N)
sabretmemek (not to be patient)	asimile (assimilated) (N)
sabret+Verb+**Neg**^DB+Noun+Inf1+ A3sg	karizmatik (charismatic) (P)
sabırlı (patient)	dağınık, savruk (untidy) (N)
sabırlı+Noun+Prop+A3sg	muhteşem, harika (gorgeous) (P)
sabır+Noun+A3sg^DB+Adj+**With**	üçkağıt (fiddle) (N)
sabır+Noun+Prop+A3sg^DB+Adj+**With**	dingin (calm) (P
sabırsız (impatient)	galeyan (ebullition) (N)
sabır+Noun+A3sg^DB+Adj+**Without**	filizlenmek (sprout) (P)
sabır+Noun+Prop+A3sg^DB+Adj+**Without**	pandemi, salgın (pandemic, epidemic) (N)
sabırlı değil(without patient)	iyi (good) (P)
değil+Conj	çene çalmak, gevezelik (chitchat, chatter) (N)
değil+Verb+**Neg**+Pres+A3sg	nezaket, nazik (kindness, gentle) (P)
sabrı **yok** (have no patience)	kötümser, karamsar (pessimistic) (N)
yok+Conj	efsanevi (legendary) (P)
yok+Adj	antisosyal (antisocial) (N)
yok+Noun+Prop+A3sg	öncü (pioneer) (P)
sabrı **var** (does have a patience)	sendrom (syndrome) (N)
var+Adj	alçakgönüllü (humble) (P)
var+Verb+Imp+A2sg	intihal (plagiarism) (N)
var+Noun+A3sg	aklıselim (common sense) (P)

absence indicator for nouns. For example, “sabrı yok” (have no patience) has negative polarity, and “sabrı var” (have patience) has a positive polarity. We annotated “değil and “yok” with as a negative; and “var” as a positive.

In addition to Turkish linguistics difficulties, sentiment analysis has syntactic and semantic issues, that are encountered in almost any language. Even sentiment analysis is considered as a binary classification problem; it is hard to understand compositional meanings, rhetorical questions, negations, or ironic phrases regardless of the language (Dehkharghani et al., 2017). Additionally, when a user-generated dataset is considered, i.e., when noisy dataset from social media is used, domain-dependent issues occur. Hence, it is needed to be done time-consuming specialised pre-processing steps for cleaning a dataset. Moreover, understanding of the polarity of the domain-specific reviews requires the use of domain-specific (Dehkharghani et al., 2017) polarity lexicons, enhanced feature extraction rules, and annotation strategies. These are generally done by time-consuming manual processes, also humans can have additional bias while annotating words, and they can add additional subjectivity to the classification.

The main difficulty in terms of the semantics is capturing the meaning of longer phrases. This study mainly focused on combining morph level, stem level, word-phrase level, and sentence (review) level sentiment information, with the implementations of various datasets from different domains, including movie reviews and various product reviews. Most previous studies have been done based on a binary prediction of a given sentence due to the limited resources of a fine-grained labelled large dataset. Hence, we introduce a novel fine-grained sentiment classification dataset to contribute to the lack of Turkish NLP resources.

4. Morphologically enriched sentiment Treebank for Turkish (MS-TR)

MS-TR has been constructed as a new sentiment analysis resource to work with tree-structured Recursive Deep Learning models. The main drawback for the data-driven classification systems is that they have task-dependent well-constructed annotated data. Researchers have proposed to use emoticons, and emojis (Read, 2005; Suttles & Ide, 2013) or hashtags (Park et al., 2018) as classification labels inferring polarity of a given task as an extension of distant supervision (Mintz et al., 2009) to overcome this issue. Similar to those approaches, we propose using morphological features of each word to infer the polarity of the inner nodes of the MS-TR, which contains annotated binary-structured parse trees. Parsed trees were labelled according to fine-grained (multi-class) and binary sentiment class. The details of the proposed model are given in this section.

4.1. System architecture, resources and tools used in building MS-TR

MS-TR was constructed based on the combination of several NLP modules. Each module had been used to achieve a specified task with the combination of a series of data preparation steps. This section contains detailed information about the used dataset and NLP resources to build MS-TR.

Datasets Two different dataset formats have been used for the construction of the MS-TR. Binary-labelled MS-TR was constructed using movie reviews and multi-domain product reviews with a raw text format. A fully-labelled fine-grained MS-TR was constructed using the latest updated version of BOUN Treebank (Turk et al., 2019). BOUN Treebank includes 9,757 sentences from different application domains, such as newspapers, magazines, and essays of Turkish Natural Corpus (Aksan et al., 2012). Each sentence in BOUN Treebank was encoded in ConLL-U format, designed for Universal Dependencies (UD) framework. Movie reviews contain 5331 positive and 5330 negative reviews collected by Demirtas and Pechenizkiy (Demirtas & Pechenizkiy, 2013) from beyazperde.com. The reviews have star ratings of 0 to 5 scale. In this study, 4-star or 5-star movie reviews used as positive labelled movie reviews and 0-star or 1-star movie reviews used as negative labelled movie reviews. Multi-domain product reviews including reviews about the books, DVD, electronics and kitchen appliances. Each of the products has 700 positive and 700 negative reviews collected from Turkish commercial website hepsiburada.com. The average and maximum N-grams lengths of reviews are given in Table 3.

Figure 2 represents the system architecture of the proposed MS-TR model's annotation and binarization process. The proposed pipeline starts with pre-processing steps and sentence boundary detection. Before the annotation and morphological analysis, each review in the dataset is parsed at the sentence boundary detection phase and then cleaned by Turkish spell checking.

Table 3. Maximum and average *N*-grams length of the products datasets of MS-TR

Datasets	**Max.*N*-Gram Length**	**Avr. *N*-Gram Length**
Books	207	33.19
DVD	253	31.75
Electronic	245	37.51
Kitchen	182	32.66
Movie	1,566	33.20

Zemberek was used (Akın & Akın, 2007) for pre-processing steps, including sentence boundary detection and morphological parser. Turkish spell checking has also been done since Turkish has additional letters such as ("ç", "ğ", "ı", "ö", "ş", "ü"), that is missing in English, the raw text from the dataset usually contains non-Turkish characters because of the non-Turkish keyboards. Each word morphologically analyzed and labelled according to the negation cases, as discussed in the previous section, case 1, case 2, and case 3. In addition to the negation handling strategies, we also used a polar word list to detect sentiment-
specific adjectives and nouns in an unsupervised manner. Zemberek is only used for the movie reviews, and product reviews as a morphological analyzer since CoNLL-U format dataset has already contained FEATS field that contains morphological features for words.

In Figure 1, it can be seen the parse tree of two different sentences: "Bu bağlamda inceleyeceğimiz iki gök haritası somut örnekler sunacak." and "Zeki Müren, Müzeyyen Senar, Safiye Ayla çocukluğumdan beri tanış olduğum isimler." from the CoNNL-U dataset. The word "sunacak" is a root of the parse tree, and it can be seen POS tags (i.e. NOUN, ADJ, PUNCT) of the leaf words and root word of the parse tree. These parse trees (graphs) are examples of structural representation of the sentences that can be processed in Recursive Neural Models (Tree-RNNs).

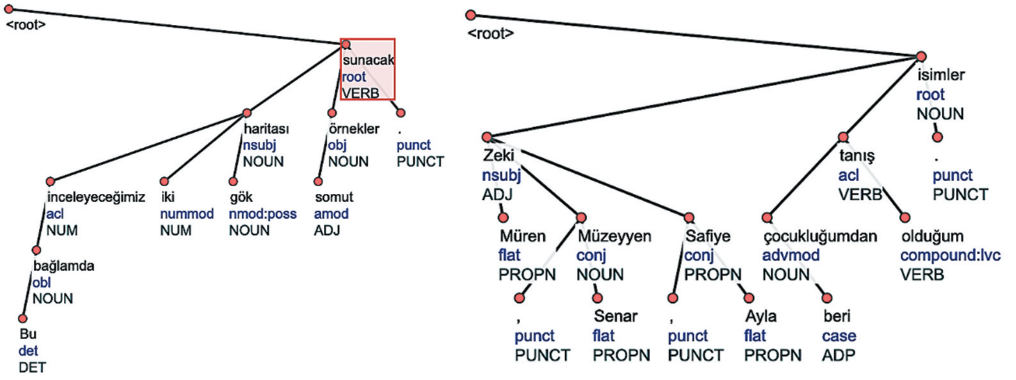
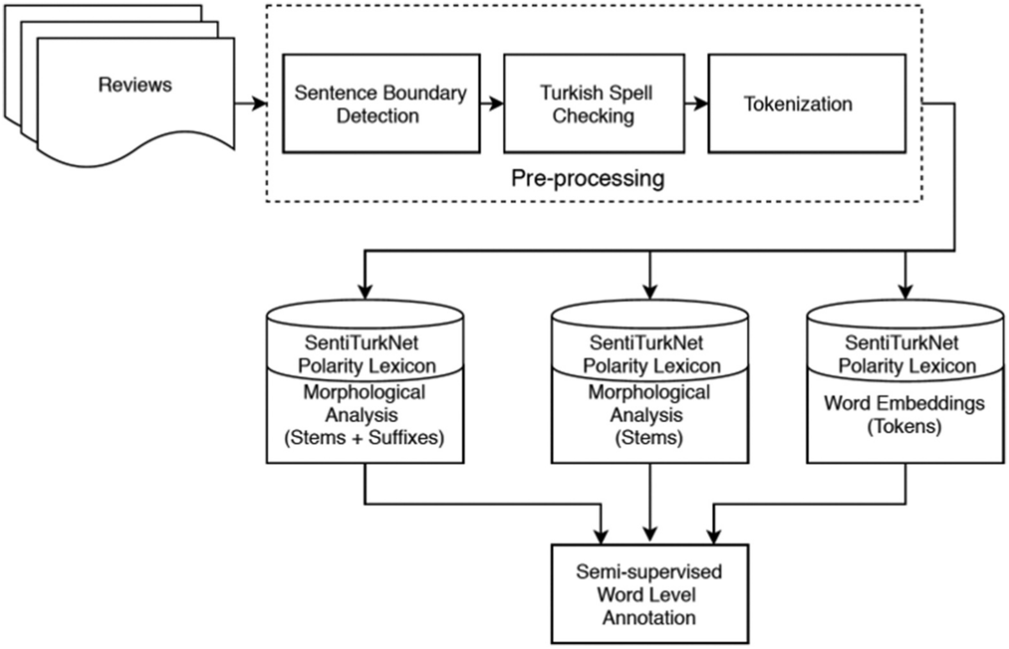


Figure 1. A constituency parse tree of two different sentences from the BOUN Treebank **(Turk et al., 2019).**

Figure 2. The pipeline of the binarization framework to construct MS-TR.


Reviews
Sentence Boundary Detection
Turkish Spell Checking
Tokenization
Pre-processing
SentiTurkNet Polarity Lexicon
Morphological Analysis (Stems + Suffixes)
SentiTurkNet Polarity Lexicon
Morphological Analysis (Stems)
SentiTurkNet Polarity Lexicon
Word Embeddings (Tokens)
Semi-supervised Word Level Annotation

Tree-RNNs have been proposed for learning from the arbitrary shape structures like trees and graphs; hence it is needed to implement a binary parse tree to train Recursive Neural Tensor Networks (RNTNs). The Stanford Core NLP tool has been used to build the binarized dataset, i.e. fully labelled parse trees at the final step of the MS-TR construction. This is a tool to set the class labels on a sentence tree dataset using the user default annotations. The expected input file for the Stanford Core NLP should contain one sentence per line. The reviews are separated by blank lines and each review is labelled with their sentiments. Before the blank line, sub-phrases of the current review can also be labelled with their own label. Consequently, all the labels on a tree are set to agiven default value. The annotation strategy of the MS-TR is different from the Stanford Sentiment Treebank (STS), which was developed by Socher et al. (Socher et al., 2013) for English. We propose to annotate MS-TR in a semi-supervised manner to construct fully labelled parse trees. The following section gives the details and pseudo-codes of the proposed annotation phase of the MS-TR.

4.2. Semi-supervised annotation strategies of the Turkish sentiment Treebank

We present two pathways as an automatic semi-supervised annotation approach for binary classification datasets. The first pathway is based on the combination of morphological features of words, and the second pathway is based on the polarity lexicon and the polar word embedding models. By following two pathways, we have constructed four different treebanks, namely morph-level (stems+suffixes) annotated MS-TR, stem-level annotated MS-TR, token-level annotated MS-TR, and review-level annotated MS-TR. Each model has been constructed for each dataset to figure out the compositional effects of the morphologically rich structure of Turkish. Each annotation level has been tested to compare the efficiency of the proposed models by feeding them into the Recursive Neural Tensor Networks.

1) Morph-Level Annotated MS-TR: In this model, the annotation has been done at morph-level, aiming to retrieve the hidden polarity of the suffixes in the morphologically rich word. We propose to construct a fully-labelled morphologically enriched treebank (Morph-Level MS-TR) with the morphological features of the words, which are used as a distant supervision method similar to the (Mintz et al., 2009; Park et al., 2018; Read, 2005; Suttles & Ide, 2013). To this end, tokens are morphologically analyzed, and they are parsed to their possible stem and suffixes. All the possible morphological results for a given input word were used for the word-level annotation. If the word contain negation suffixes as given in Section 3.1, the parsed stem, and suffixes (ending) of the word labelled with 0, if it contains positive suffixes the parsed stem and suffixes (ending) of the word have been annotated with 1. For example, sabırsız (sabır+sız) "impatient" is represented as ((0 sabır) (0 sız)) in the proposed Morph-Level Annotated MS-TR. Similarly, sabırlı (sabır+lı) "patient" is represented as ((1 sabır)(1 lı)) as a morph-level annotated tree in the MS-TR.

In addition to the morphological annotation, we propose to use polarity lexicon to capture polar words, which are rooted and do not have any morphological information. Words in Table 2(b) have been used as a case examples to elaborate on this point. Using only a morphological analysis of the word cannot provide the correct polarity information for each case. For example, the sentence "Seni yeniden görmek için sabırsızlanıyorum" (I can't wait to see you again/looking forward to seeing you again) has a positive meaning, but the morphological information of the "sabırsızlanıyorum" contain negation suffix -sız (sabır+sızlanıyorum). To handle this issue, we proposed to use sentiment polarity lexicon, SentiTurkNet (Dehkharghani et al., 2016), and we combine morphological features and polarity lexicon information to annotate each word of the review. After the tokenization step, the polarity words are controlled by the hybrid usage of morphological analysis and SentiTurkNet lexicon. This approach is used for binary annotation of the MS-TR. The detailed steps of the annotation algorithm are given in Figure 4.

Figure 3 represents the hierarchical structure of the phrase *çok eğlenceli bulamadığımız bir film*, "a movie that we could not find much enjoyable". Each token of the phrase has been parsed and annotated to learn review level sentiment recursively in a bottom-up manner. As it can be seen at the morph-level, suffix—li "with" and root word eğlence "enjoy" are composed as a distributional word vectors of left child $(c_{l,root})$ and right child $(c_{r,suffix})$ to calculate parent representation of the word eğlenceli "enjoyable". This combination combines the positive sentiments from child nodes to the parent node. The second morph-level composition is a clear example of the handling negation of the morphologically rich word. The word bulamadığımız "that we could not find", contains a negation suffix -ma "without", hence the right child $(c_{r,suffix})$ is negative. As a result the compositional parent node bulamadığımız "that we did not find" is also negative. The longest phrase *çok eğlenceli bulamadığımız bir film*, "a movie that we could not find much enjoyable" is produced recursively at the same dimension in a bottom-up manner and annotated as a negative.

Figure 3. An example for the morph-level annotated tree structure of phrase çok eğlenceli bulamadığımız bir film, "a movie that we could not find much enjoyable" from MS-TR.

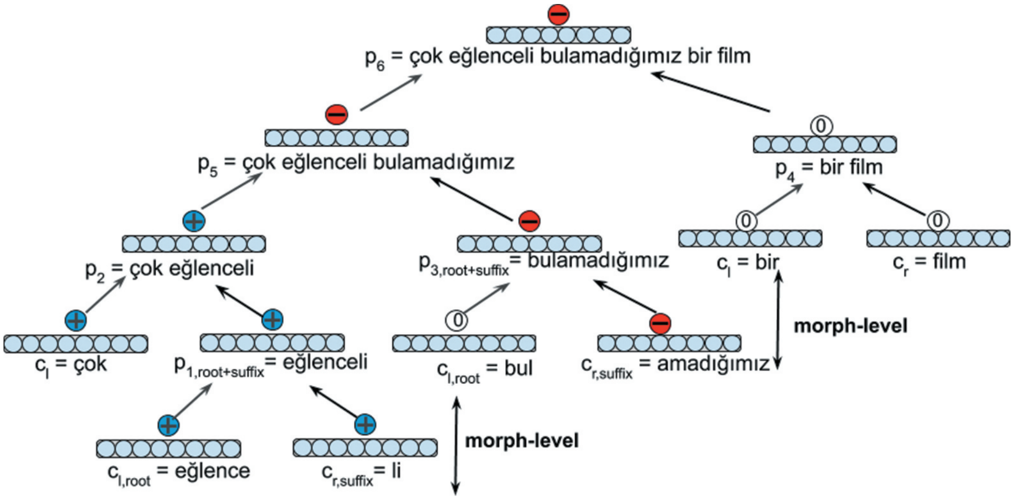
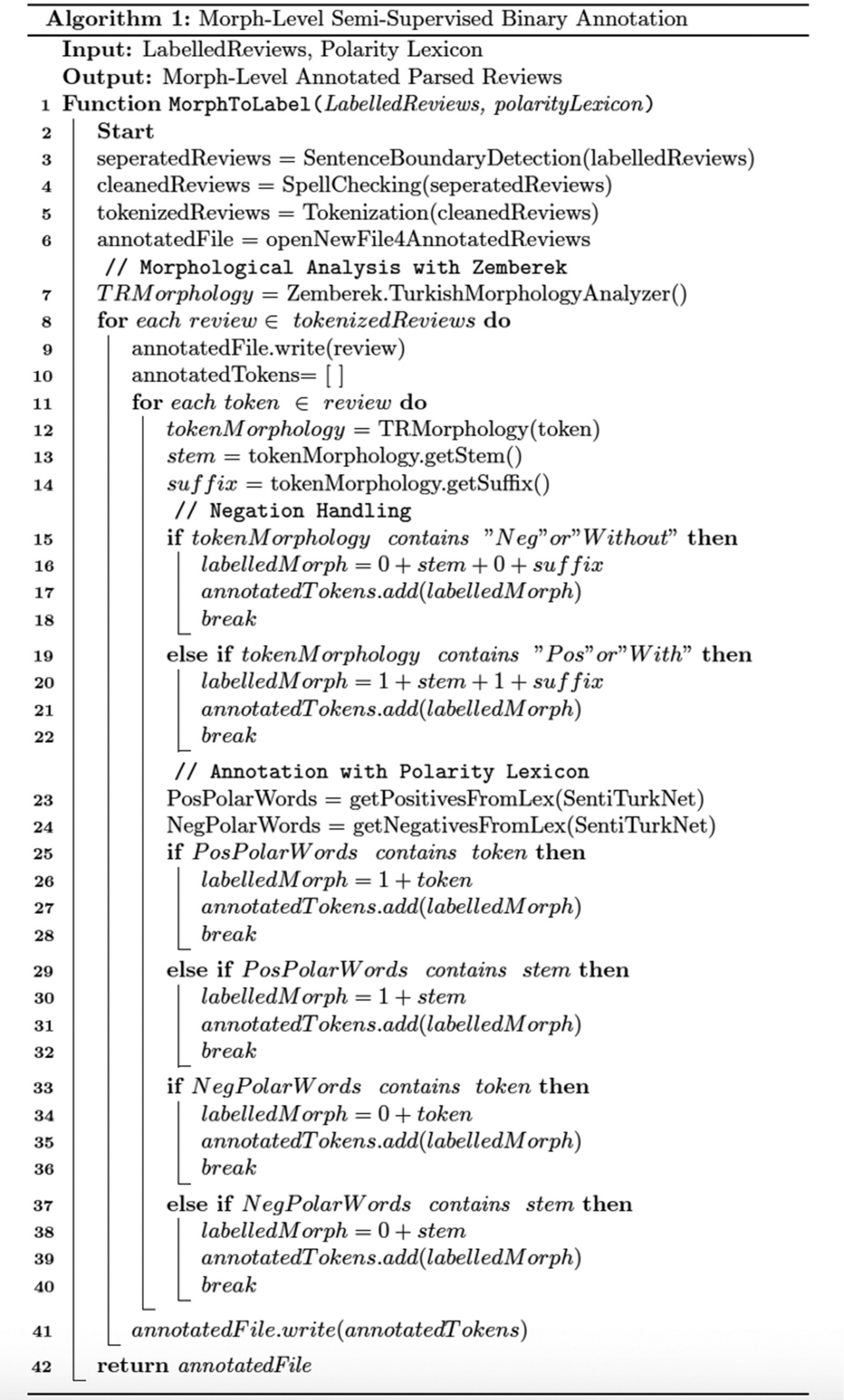

−
p_6 = çok eğlenceli bulamadığımız bir film
−
p_5 = çok eğlenceli bulamadığımız
0
p_4 = bir film
+
p_2 = çok eğlenceli
−
$p_{3,root+suffix}$ = bulamadığımız
0
c_l = bir
0
c_r = film
morph-level
+
c_l = çok
+
$p_{1,root+suffix}$ = eğlenceli
0
$c_{l,root}$ = bul
−
$c_{r,suffix}$ = amadığımız
+
$c_{l,root}$ = eğlence
+
$c_{r,suffix}$ = li
morph-level

Figure 4. Pseudo-code of the semi-supervised morph-level annotation.



```
Algorithm 1: Morph-Level Semi-Supervised Binary Annotation
Input: LabelledReviews, Polarity Lexicon
Output: Morph-Level Annotated Parsed Reviews
Function MorphToLabel(LabelledReviews, polarityLexicon)
    Start
    seperatedReviews = SentenceBoundaryDetection(labelledReviews)
    cleanedReviews = SpellChecking(seperatedReviews)
    tokenizedReviews = Tokenization(cleanedReviews)
    annotatedFile = openNewFile4AnnotatedReviews
    // Morphological Analysis with Zemberek
    $TRMorphology$ = Zemberek.TurkishMorphologyAnalyzer()
    for each review $\in$ tokenizedReviews do
        annotatedFile.write(review)
        annotatedTokens= [ ]
        for each token $\in$ review do
            $tokenMorphology$ = TRMorphology(token)
            $stem$ = tokenMorphology.getStem()
            $suffix$ = tokenMorphology.getSuffix()
            // Negation Handling
            if tokenMorphology contains "Neg" or "Without" then
                $labelledMorph = 0 + stem + 0 + suffix$
                $annotatedTokens.add(labelledMorph)$
                break
            else if tokenMorphology contains "Pos" or "With" then
                $labelledMorph = 1 + stem + 1 + suffix$
                $annotatedTokens.add(labelledMorph)$
                break
            // Annotation with Polarity Lexicon
            PosPolarWords = getPositivesFromLex(SentiTurkNet)
            NegPolarWords = getNegativesFromLex(SentiTurkNet)
            if PosPolarWords contains token then
                $labelledMorph = 1 + token$
                $annotatedTokens.add(labelledMorph)$
                break
            else if PosPolarWords contains stem then
                $labelledMorph = 1 + stem$
                $annotatedTokens.add(labelledMorph)$
                break
            if NegPolarWords contains token then
                $labelledMorph = 0 + token$
                $annotatedTokens.add(labelledMorph)$
                break
            else if NegPolarWords contains stem then
                $labelledMorph = 0 + stem$
                $annotatedTokens.add(labelledMorph)$
                break
        $annotatedFile.write(annotatedTokens)$
    return annotatedFile
```

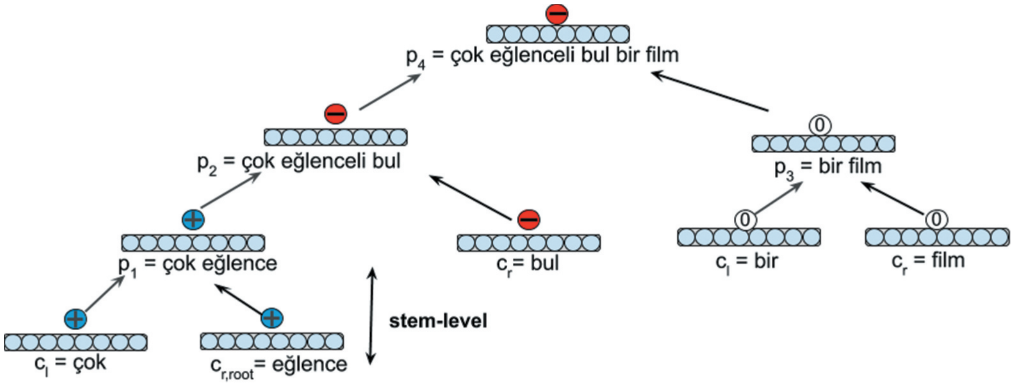


Figure 5. An example representation for the stem-level annotated tree structure of phrase *çok eğlenceli bulamadığımız bir film*, “a movie that we could not find much enjoyable” from MS-TR.

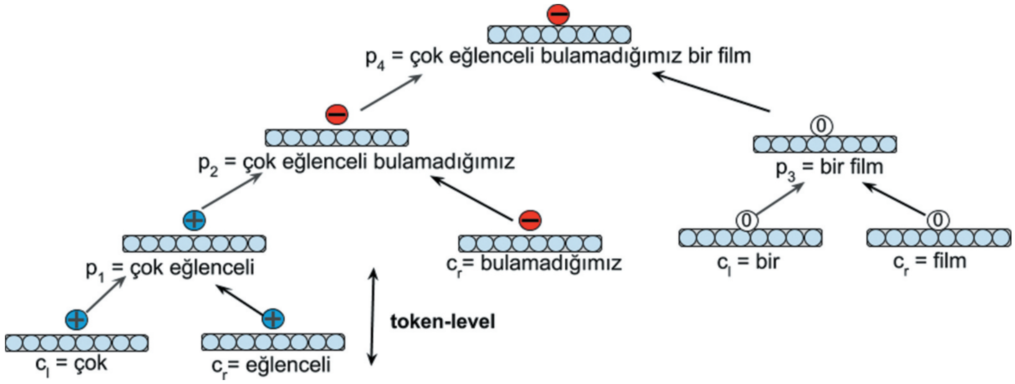
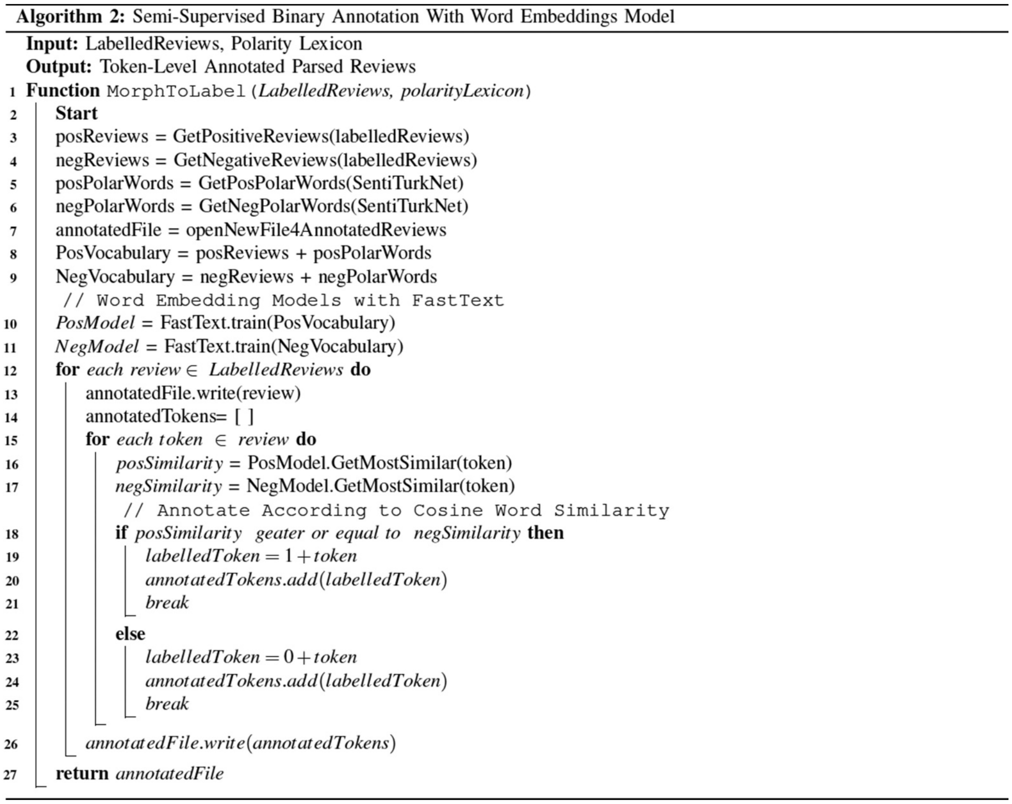


Figure 6. An example representation for the token-level annotated tree structure of phrase çok eğlenceli bulamadığımız bir film, “a movie that we could not find much enjoyable” from MS-TR.

2) Stem-Level Annotated MS-TR: Stem-Level Annotated MS-TR was constructed similar to morph-level annotated MS-TR. The only difference is using only stems of the words, and the suffixes (ending) of the words are eliminated from word after morphological parsing. The aim of constructing stem-level MS-TR is to investigate the efficiency of the using stem+suffix structure by comparing to using the only stem of the words. The flowchart of the annotation algorithm is similar to Figure 4 and the example tree structure of the stem-level annotated MS-TR is given in Figure 5.

3) Token-Level (Surface Level) Annotated MS-TR: As a third level, we propose to annotate each token of the reviews using polar embedding spaces, which are constructed by using positive and negative datasets. Related word vectors have been produced for each polarity level using the word embeddings model. FastText (Grave et al., 2019) model has been used to take its advantage of representing out-of-vocabulary word vectors for MRLs. To this end, the positive embedding model has been constructed by using positive reviews dataset and positive polar words that are taken from SentiTurkNet. Similarly, negative word embedding space has been constructed using negative reviews dataset and negative polar words taken from SentiTurkNet. The most similar word of the token has been found by using positive word vector space and negative word vector space.

Figure 7. Pseudo-code of the semi-supervised token-level annotation.

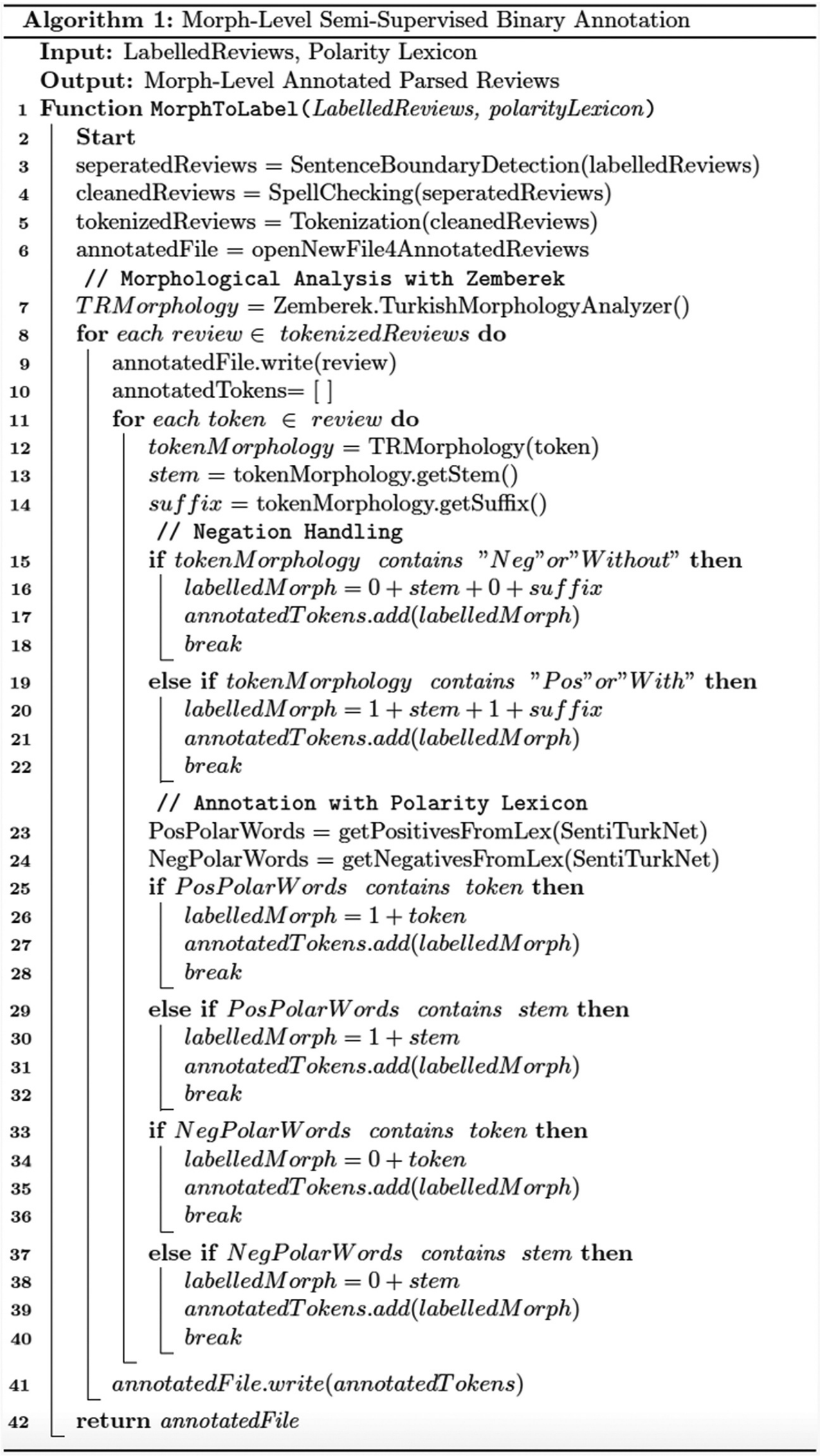


```
Algorithm 2: Semi-Supervised Binary Annotation With Word Embeddings Model
Input: LabelledReviews, Polarity Lexicon
Output: Token-Level Annotated Parsed Reviews
Function MorphToLabel (LabelledReviews, polarityLexicon)
    Start
    posReviews = GetPositiveReviews(labelledReviews)
    negReviews = GetNegativeReviews(labelledReviews)
    posPolarWords = GetPosPolarWords(SentiTurkNet)
    negPolarWords = GetNegPolarWords(SentiTurkNet)
    annotatedFile = openNewFile4AnnotatedReviews
    PosVocabulary = posReviews + posPolarWords
    NegVocabulary = negReviews + negPolarWords
     // Word Embedding Models with FastText
    PosModel = FastText.train(PosVocabulary)
    NegModel = FastText.train(NegVocabulary)
    for each review ∈ LabelledReviews do
        annotatedFile.write(review)
        annotatedTokens= [ ]
        for each token ∈ review do
            posSimilarity = PosModel.GetMostSimilar(token)
            negSimilarity = NegModel.GetMostSimilar(token)
             // Annotate According to Cosine Word Similarity
            if posSimilarity geater or equal to negSimilarity then
                $labelledToken = 1 + token$
                $annotatedTokens.add(labelledToken)$
                break
            else
                $labelledToken = 0 + token$
                $annotatedTokens.add(labelledToken)$
                break
        $annotatedFile.write(annotatedTokens)$
    return annotatedFile
```

After tokenization step of each review, the cosine similarity measure has been used to find the token-level label. If the cosine similarity of the positive most similar word and the target token is bigger than the cosine similarity of the negative most similarity word and the target token, the token is labelled as positive; else it is labelled as negative. The flowchart of the proposed algorithm is given in Figure 7, and the example tree structure of the token-level annotated MS-TR is given in Figure 6.

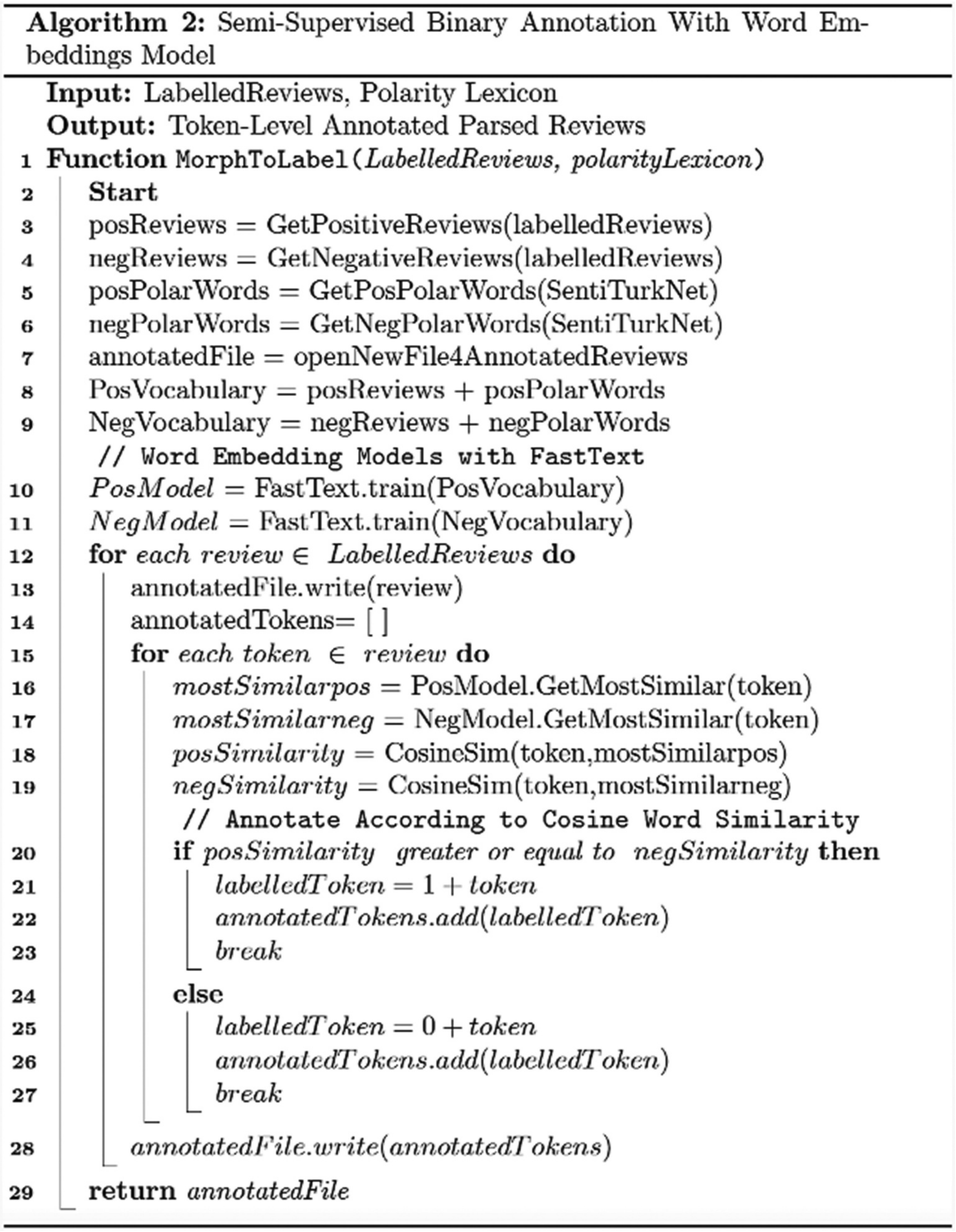
4) Review-Level Annotated MS-TR: As the last annotation level, we proposed using only review-level annotated tree structures to construct MS-TR for comparing each annotation level's performance. After the annotation process has been done, the annotated files are fed into the Standford Core NLP module to construct the binarized tree structure by parsing each review of the datasets.

In addition to the binary-labelled MS-TRs, fine-grained MS-TR has been annotated by taking into account the morphological information of the words. 1003 of BOUN Treebank sentences have been selected randomly as an initial step. Parsing the 1003 sentences has produced 63,782 nodes, including very positive, somewhat positive, neutral, somewhat negative, negative and very negative combinations of the tokens. The polarity distribution of the phrases, i.e. phrase-level to review-level labelling was realized in two stages. The first stage is labelling words according to their polarity features which are detected from the morphological analysis of the words. The morphological feature of the word was detected from the FEATS field of the CoNLL-U data format, which provides the lemma and morphological analysis of the word, including Polarity feature as proposed in the previous section. In addition to the binary annotation, we have scored words whether they are booster words. If the word is contained by positive booster words list, it was annotated with 5 for very positive class. Similarly, if the negative booster words list contains the word, it was annotated with 1 for very negative class. The total sentence-level score was calculated as follows:



Algorithm 1: Morph-Level Semi-Supervised Binary Annotation

```
Input: LabelledReviews, Polarity Lexicon
Output: Morph-Level Annotated Parsed Reviews
Function MorphToLabel(LabelledReviews, polarityLexicon)
    Start
    seperatedReviews = SentenceBoundaryDetection(labelledReviews)
    cleanedReviews = SpellChecking(seperatedReviews)
    tokenizedReviews = Tokenization(cleanedReviews)
    annotatedFile = openNewFile4AnnotatedReviews
    // Morphological Analysis with Zemberek
    $TRMorphology$ = Zemberek.TurkishMorphologyAnalyzer()
    for each review $\in$ tokenizedReviews do
        annotatedFile.write(review)
        annotatedTokens= [ ]
        for each token $\in$ review do
            $tokenMorphology$ = TRMorphology(token)
            $stem$ = tokenMorphology.getStem()
            $suffix$ = tokenMorphology.getSuffix()
            // Negation Handling
            if $tokenMorphology$ contains "$Neg$"$or$"$Without$" then
                $labelledMorph = 0 + stem + 0 + suffix$
                $annotatedTokens.add(labelledMorph)$
                $break$
            else if $tokenMorphology$ contains "$Pos$"$or$"$With$" then
                $labelledMorph = 1 + stem + 1 + suffix$
                $annotatedTokens.add(labelledMorph)$
                $break$
            // Annotation with Polarity Lexicon
            PosPolarWords = getPositivesFromLex(SentiTurkNet)
            NegPolarWords = getNegativesFromLex(SentiTurkNet)
            if $PosPolarWords$ contains $token$ then
                $labelledMorph = 1 + token$
                $annotatedTokens.add(labelledMorph)$
                $break$
            else if $PosPolarWords$ contains $stem$ then
                $labelledMorph = 1 + stem$
                $annotatedTokens.add(labelledMorph)$
                $break$
            if $NegPolarWords$ contains $token$ then
                $labelledMorph = 0 + token$
                $annotatedTokens.add(labelledMorph)$
                $break$
            else if $NegPolarWords$ contains $stem$ then
                $labelledMorph = 0 + stem$
                $annotatedTokens.add(labelledMorph)$
                $break$
        $annotatedFile.write(annotatedTokens)$
    return $annotatedFile$
```



Algorithm 2: Semi-Supervised Binary Annotation With Word Embeddings Model

```
Input: LabelledReviews, Polarity Lexicon
Output: Token-Level Annotated Parsed Reviews
Function MorphToLabel(LabelledReviews, polarityLexicon)
    Start
    posReviews = GetPositiveReviews(labelledReviews)
    negReviews = GetNegativeReviews(labelledReviews)
    posPolarWords = GetPosPolarWords(SentiTurkNet)
    negPolarWords = GetNegPolarWords(SentiTurkNet)
    annotatedFile = openNewFile4AnnotatedReviews
    PosVocabulary = posReviews + posPolarWords
    NegVocabulary = negReviews + negPolarWords
    // Word Embedding Models with FastText
    $PosModel$ = FastText.train(PosVocabulary)
    $NegModel$ = FastText.train(NegVocabulary)
    for each review ∈ LabelledReviews do
        annotatedFile.write(review)
        annotatedTokens= [ ]
        for each token ∈ review do
            $mostSimilarpos$ = PosModel.GetMostSimilar(token)
            $mostSimilarneg$ = NegModel.GetMostSimilar(token)
            $posSimilarity$ = CosineSim(token,mostSimilarpos)
            $negSimilarity$ = CosineSim(token,mostSimilarneg)
            // Annotate According to Cosine Word Similarity
            if posSimilarity greater or equal to negSimilarity then
                $labelledToken = 1 + token$
                $annotatedTokens.add(labelledToken)$
                break
            else
                $labelledToken = 0 + token$
                $annotatedTokens.add(labelledToken)$
                break
        annotatedFile.write(annotatedTokens)
    return annotatedFile
```

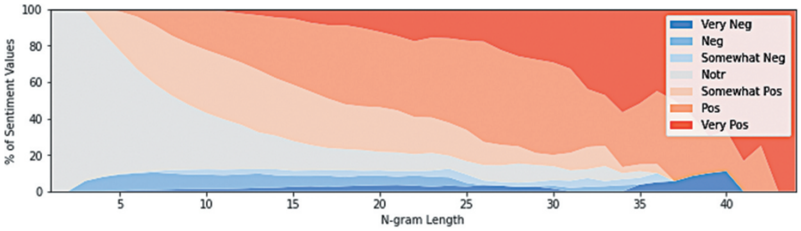


Figure 8. Normalized histogram of the annotated *n*-grams in fine-grained MS-TR. Many of the shorter *n*-grams are neutral, and many longer phrases are positive. Somewhat positive distributions are added to the positive sentiment class, and somewhat negative distributions are added to the negative sentiment class for 5 class sentiment classification.

$$senti_{Score} = -10a - 5b + 5c + 10d \tag{1}$$

Here *a, b, c*, and *d* represents the total number of the very negative, negative, positive, and very positive word in a given sentence, respectively. Booster words were taken from polarity lexicon, which are used for morph-level and stem-level annotation. After calculating the total sentiment score of the sentence, scores of −15 and less are labelled as very negative (1), those with −10 labelled as negative (2), 0 is labelled as neutral (3), 10 is labelled as positive (4), and those above 15 and 15 are labelled as very positive (5). Table 4 represents the total number of the fine-grained n-grams for each sentiment class. Somewhat positive and somewhat negative n-grams got emotion scores of 5 and −5, respectively; hence, we considered them in positive and negative, respectively. After semi-supervised annotation process, 188 reviews were labelled as class 5 (very positive), 215 reviews were labelled as positive, 230 reviews were labelled as neutral, 207 reviews labelled as negative, and 163 reviews labelled as very negative Figure 8.

We used the Stanford Core NLP (Manning et al., 2015) module to build a binarized tree structure similar to the Stanford Sentiment Treebank (STS) (Socher et al., 2013). The total inner nodes of the annotated MS-TR that have been produced by binarized parsing have been given in 6.

Table 4. Total numbers of fine-grained *n*-grams

N-Grams	Total Number
Very Positive	8,{231
Positive	7,610
Somewhat Positive	22,635
Notr	15,392
Somewhat Negative	2,471
Negative	2,559
Very Negative	214

5. Recursive Deep Models over MS-TR for Compositional Semantics

The compositional power of the Tree-RNNs is still waiting to be explored for Turkish sentiment analysis task. With this motivation, we employ MS-TR with Recursive Neural Tensor Networks (RNTN) to handle Turkish's agglutinative morphology and catch the freedom of its constituent structure for compositional sentiment analysis. The proposed model also contributes to the lack of data sources for improving understanding of semantics in Turkish.

5.1. Recursive Neural Tensor Networks with Morphologically Enriched Treebank

Recursive Neural Network (Tree-RNN) is a tree-structured model based on composing words over nested hierarchical structure in sentences. The neural network function recursively merges words to construct noun phrases until representing the entire sentence. Tree-RNNs have an extraordinary ability for mapping of phrases in a semantic space (Socher et al., 2010).

Recursive Neural Tensor Networks (RNTN) achieved good results both for English (Socher et al., 2013) and morphologically rich languages (MRL) such as Arabic (Baly et al., 2017). RNTN reached state-of-art accuracy results compare to the previous versions of the Recursive Neural Networks, such as Tree-RNNs and Matrix-Vector Recursive Neural Networks (MV-RNN). Hence we used RNTN over a morphologically enriched Turkish sentiment treebank.

The proposed model learns the distributional representations to construct not only phrases to but also morphologically rich words from their root and suffixes. Our main motivation is to learn suffixes that contain sentiment polarity, as described in the previous section. To this end, we first described the shallow versions of the Tree-RNNs, and then the applied Recursive Neural Tensor Network over MS-TR for the sentiment classification task.

1) Recursive Neural Networks (Tree-RNNs): Tree-RNNs are the simplest version of the Recursive Deep Models. In essence, they are designed to process tree-structured datasets, and they generalize the sequential models from chain-structures to the tree-structures. Assume that $x^{(s)} = (x_1, x_2, \ldots, x_k)$, is the sequence of d-dimensional word vectors of a given sentence for $x_i \in \mathbb{R}^d$, $1 \le i \le k$. $x^{(s)} = (x_1, x_2, \ldots, x_k)$ has been learned using binary tree structure by evaluating parent word vectors with following recursive function:

$$\begin{aligned} x_p &= \varphi\left(W_{emb}\left[x_l; x_r\right],\ b_{emb}\right) \\ &= \varphi\left(W_{emb}\begin{bmatrix} x_l \\ x_r \end{bmatrix},\ b_{emb}\right) \end{aligned} \tag{2}$$

Here φ represents the tanh function, and x_l, x_r, and x_p are $\mathbb{R}^{dx1}$ dimensional word vectors of left child word, right child word, and parent word, respectively. The concatenation of the x_l, x_r is represented by $[x_l; x_r] \in \mathbb{R}^{2dx1}, W_{emb} \in \mathbb{R}^{dx2d}$, and $b_{emb} \in \mathbb{R}^{dx1}$.

2) Recursive Neural Tensor Networks (RNTN): Recursive Neural Tensor Networks (RNTN) is the enhanced version of the tree-based recursive deep learning models, aiming to aggregate the polarity of child nodes to the parent root node using one direct composing relation by tensor algebra (Socher et al., 2013). The learning architecture of the model is the same as the Tree-RNN. The modification has been done using the following aggregation:

$$h_{pi} = \begin{bmatrix} x_l \\ x_r \end{bmatrix}^{\mathrm{T}} \mathrm{T}^{[i]} \begin{bmatrix} x_l \\ x_r \end{bmatrix} \tag{3}$$

$$h_p = \begin{bmatrix} x_l \\ x_r \end{bmatrix}^{\mathrm{T}} \mathrm{T}^{[1:d]} \begin{bmatrix} x_l \\ x_r \end{bmatrix} + W_{emb} \begin{bmatrix} x_l \\ x_r \end{bmatrix} \tag{4}$$

$$x_p = \varphi\left(\begin{bmatrix} x_l \\ x_r \end{bmatrix}^{\mathrm{T}} \mathrm{T}^{[1:d]} \begin{bmatrix} x_l \\ x_r \end{bmatrix} + W_{emb} \begin{bmatrix} x_l \\ x_r \end{bmatrix} \right) \tag{5}$$

Tensor defined as $T^{[1:d]} \in \mathbb{R}^{2dx2dxd}$ and $[x_l; x_r]^{\mathrm{T}} \in \mathbb{R}^{1x2d}$, where x_l, x_r, h_p are $\in \mathbb{R}^{dx1}$ dimensional word vectors of left child word, right child word and hidden parent word which is the output of the tensor-layer, respectively. $T^{[i]}$ is a slice of a tensor $\in \mathbb{R}^{dxd}$. This process is done recursively as defined in the Tree-RNN and φ represents the same tanh function. The primary motivation to use RNTN is to satisfy the direct association between input vectors. The quality of the parent vector is calculated by reconstruction layer as follows:

$$[x_l{}'; x_r{}'] \quad = \quad W_{rec}x_p + b_{rec} \tag{6}$$

Here $W_{rec} \in \mathbb{R}^{dx2d}$, $b_{rec} \in \mathbb{R}^{dx1}$. The aim is minimizing the reconstruction error, which is defined as follows:

$$\begin{aligned} E_{EMB} \quad &= L\ ([x_l; x_r],\ [x_l{}'; x_r{}']) \\ &= argmin \parallel [x_l; x_r] - [x_l{}'; x_r{}']\|^2 \end{aligned} \tag{7}$$

The parent word vector x_p scored positively or negatively at every internal, using softmax function for the binary classification task. The phrases are merged recursively to predict sentiment class of the root vector as follows:

$$\begin{aligned} y^{x_p} \quad &= softmax\ (W_{label}x_p) \\ &= \frac{exp(W_{label}x_p)}{\sum_{i=1}^{c} exp(W_{label}\ x_{p_i})} \end{aligned} \tag{8}$$

y^{x_p} is the predicted sentiment, $W_{label} \in \mathbb{R}^{cxd}$, and c is the number of sentiment class. Here $y^{x_p} = softmax\ (W_{label}x_p)$ can be interpreted as a conditional probability $y^{x_p} = p(c|[x_l; x_r])$ and $\sum_{i=1}^{c} y^{x_{pi}} = 1$ and for a given target (true) label t^{x_p} of the parent vector, the total cross-entropy loss is defined as follows:

$$\begin{aligned} E_{CE} \quad &= -\sum_{i=1}^{c} E(x_{pi}, t^{x_{pi}}, \theta) \\ &= -\sum_{i=1}^{c} t^{x_{pi}} \log(y^{x_{pi}}) \end{aligned} \tag{9}$$

The total objective function is defined by the weighted sum of the reconstruction error of the compositional embeddings is E_{EMB}, and the cross-entropy error of the sentiment labels E_{CE} as follows for a given pair (x^s, t^{x_s}) from the dataset:

$$L_{total} = -\frac{1}{N_{out}} \sum_{i=1}^{N_{out}} E(x_i, t^{x_i}, \theta) + \frac{\lambda}{2} \parallel \theta\|^2 \tag{10}$$

Learning happens using backpropagation through structure with updating the learnable parameters of the model $\theta = (T,\ W_{emb},\ W_{rec},\ W_{label},\ t^{x_s})$.

6. Experiments

6.1. Experimental Setup

The reviews are split into train set, dev set, and test set. Detailed information for each dataset is given at Table 5. The hyper-parameters of the model were set following up from previous studies. Socher et al. pointed the RNTN model achieved promising results for English when the dimension of the word embeddings was set between 25 and 35. We chose 30 as a dimension of the word embeddings, which is also used as a dimension of the suffix embeddings (Socher et al., 2013). Similarly, the recommended batch size was between 20 and 30. We used the train and dev set to set batch size, learning rate. We observed that the performance of the model was decreasing for larger batch size; hence we used 20 as batch size. The learning rate was set to 0.01, and AraGrad was used as an optimizer with 0.001 weight decay regularization. The model was trained over 100 epoch. The predictions were evaluated at the sentence-level and phrase-level and compared with

Table 5. Root and inner node counts different level annotated MS-TR

Annotation	Datasets	Train/Inner	Dev/Inner	Test/Inner
Level		**Node Count**	**Node Count**	**Node Count**
Morph-Level	Books	978/98,182	141/13,507	280/27,252
	DVD	978/89,490	141/13,644	280/27,950
	Electronics	978/111,588	141/15,208	280/32,126
	Kitchen	978/95,064	141/14,501	280/28,376
	Movie	8,000/405,964	1,001/50,717	1,657/84,469
Stem-Level	Books	978/86,580	141/9,672	280/21,1819
	DVD	978/78,156	141/9,783	280/21,368
	Electronics	978/90,230	141/12,607	280/24,034
	Kitchen	978/78,309	141/10,489	280/21,629
	Movie	8,000/378,768	1,001/45,639	1,657/77,460
Token-Level	Books	978/73,316	141/10,489	280/19,878
	DVD	978/70,570	141/9,979	280/19,882
	Electronics	978/79,434	141/12,783	280/23,516
	Kitchen	978/70,631	141/9,158	280/20,332
	Movie	8,000/336,008	1,001/43,717	1,657/73,617
Review-Level	Books	978/76,917	141/10,647	280/21,562
	DVD	978/75,836	141/14,120	280/20,468
	Electronics	978/88,550	141/16,454	280/22,604
	Kitchen	978/78,309	141/10,489	280/21,629
	Movie	8,000/379,074	1,001/45,618	1,657/80,738

the baseline methods that were described in the following section. The best accuracy results were obtained over cross-validation of dev set.

6.2. Baselines

For the sake of fair comparison, MS-TR is constructed over datasets that were classified by both machine-

learning and lexicon-based approaches. The conventional machine learning algorithms including Naive Bayes (NB), Support Vector Machine (SVM) and Maximum Entropy (Max-Ent) were used as baseline methods to compare performances. Feature representation methods for each baseline models are as follows:

- **NB/SVM BoW**: BoW features combined with cross-lingual machine translation feature set from Turkish to English (Demirtas & Pechenizkiy, 2013).

- **NB/SVM BA**: SentiTurkNet and polarity lexicon, to use the average of words as a feature, basic approach (BA) (Gezici & Yanıkog˘lu, 2018).

- **NB/SVM BA-Neg**: Basic approach combined with handling negation (Gezici & Yanıkoglu, 2018).

- **NB/SVM BA-Booster**: Basic approach combined with booster words (Gezici & Yanıkoglu, 2018).

Table 6. Performance comparisons of books and DVD dataset

Dataset	Models	Accuracy %
Books	RNTN Morph-Level MS-TR	82.49
	RNTN Stem-Level MS-TR	81.84
	RNTN Token-Level MS-TR	82.80
	RNTN Review-Level MS-TR	**86.08**
	NB BA	67.49
	SVM BA	67.61
	NB BA-Neg	68.34
	SVM BA—Neg	68.92
	NB BA-Booster	69.18
	SVM BA-Neg	69.78
	NB BA-Seed Words	74.28
	SVM BA-Seed Words	75.52
	NB BoW	72.40
	NB BoW TR-MT	72.90
	SVM BoW	66.60
	SVM BoW TR-MT	66.90
	MaxEnt BoW	68.70
	MaxEnt BoW TR-MT	70.50
DVD	RNTN Morph-Level MS-TR	81.04
	RNTN Stem-Level MS-TR	**82.95**
	RNTN Token-Level MS-TR	80.84
	RNTN Review-Level MS-TR	82.42
	NB BA	67.49
	SVM BA	67.61
	NB BA-Neg	68.34
	SVM BA—Neg	68.92
	NB BA-Booster	69.18
	SVM BA-Neg	69.78
	NB BA-Seed Words	74.28
	SVM BA-Seed Words	75.52
	NB BoW	76.00
	NB BoW TR-MT	74.90
	SVM BoW	70.30
	SVM BoW TR-MT	67.60
	MaxEnt BoW	71.80
	MaxEnt BoW TR-MT	72.90

Table 7. Performance comparisons of electronics and kitchen appliances dataset

Dataset	Models	Accuracy %
Electronics	RNTN Morph-Level MS-TR	83.61
	RNTN Stem-Level MS-TR	82.65
	RNTN Token-Level MS-TR	81.87
	RNTN Review-Level MS-TR	**86.66**
	NB BA	67.49
	SVM BA	67.61
	NB BA-Neg	68.34
	SVM BA—Neg	68.92
	NB BA-Booster	69.18
	SVM BA-Neg	69.78
	NB BA-Seed Words	74.28
	SVM BA-Seed Words	75.52
	NB BoW	73.00
	NB BoW TR-MT	64.40
	SVM BoW	72.40
	SVM BoW TR-MT	64.40
	MaxEnt BoW	74.00
	MaxEnt BoW TR-MT	66.30
Kitchen	RNTN Morph-Level MS-TR	81.68
	RNTN Stem-Level MS-TR	80.73
	RNTN Token-Level MS-TR	**81.73**
	RNTN Review-Level MS-TR	79.86
	NB BA	67.49
	SVM BA	67.61
	NB BA-Neg	68.34
	SVM BA—Neg	68.92
	NB BA-Booster	69.18
	SVM BA-Neg	69.78
	NB BA-Seed Words	74.28
	SVM BA-Seed Words	75.52
	NB BoW	75.90
	NB BoW TR-MT	69.60
	SVM BoW	70.00
	SVM BoW TR-MT	67.30
	MaxEnt BoW	72.40
	MaxEnt BoW TR-MT	70.20

- **NB/SVM BA-Seed Words**: Basic approach combined with handling negation (Gezici & Yanıkoglu, 2018).

- **Max-Ent BoW**: BoW features combined with cross-lingual machine translation feature set from Turkish to English (Demirtas & Pechenizkiy, 2013).

Aforementioned baseline studies did not report the training and test splits ratio. Hence, in this study, we prefer to split train/dev/test dataset, as suggested in Socher et al.(2013) (Socher et al., 2013) for STS dataset.

7. Results and discussion

In terms of the experimental results, RNTNs have sentence level (root) and total node accuracy. For the sake of fair comparison, we report the first experiments results in terms of the micro average of the sentence (root) level accuracies of the train set, since the previous baseline studies reported the train dataset results. Tables 6 and Table 7 present the binary classification accuracy results of the }product reviews, and Table 8 presents the binary classification accuracy results of the movie reviews.

Although Socher et al. (Socher et al., 2013) mentioned that RNTN performed better for shorter reviews, we observed that compared to the traditional methods RNTN performed better even for longer reviews of Turkish movie dataset. In addition to models comparison, we also investigated the effect of different annotation levels to the accuracy results. As it can be seen in Tables 6, Table 7, and Table 8, we observed that there is not much difference between the accuracy rates of RNTN over morph-level annotated MS-TR, stem-level annotated MS-TR, token-level annotated MS-TR and review-level annotated MS-TR. However, as shown in Table 9, in terms of the inner node accuracy,

Table 8. Performance comparisons for movie reviews dataset

Models	Accuracy %
RNTN Morph-Level MS-TR	89.60
RNTN Stem-Level MS-TR	**89.84**
RNTN Token-Level MS-TR	89.71
RNTN Review-Level MS-TR	89.57
NB BA	67.49
SVM BA	67.61
NB BA-Neg	68.34
SVM BA-Neg	68.92
NB BA-Booster	69.18
SVM BA-Neg	69.78
NB BA-Seed Words	74.28
SVM BA-Seed Words	75.52
NB BoW	69.5
NB BoW TR-MT	70.0
SVM BoW	66.0
SVM BoW TR-MT	66.5
MaxEnt BoW	68.2
MaxEnt BoW TR-MT	68.6

Table 9. Performance comparisons for the total node and sentence-level accuracies of the test dataset over RNTN

Datasets	Model Level	All (Node)	Sentence-Level
(Test)		Accuracy%	(Root) Accuracy%
Movie	Morph-Level	68.22	89.60
	Stem-Level	68.28	**89.84**
	Token-Level	**68.96**	89.71
	Review-Level	68.48	89.57
Books	Morph-Level	68.86	82.49
	Stem-Level	68.34	81.84
	Token-Level	**73.55**	82.80
	Review-Level	62.14	**86.08**
DVD	Morph-Level	66.92	81.04
	Stem-Level	67.24	**82.95**
	Token-Level	**73.03**	80.84
	Review-Level	59.02	82.42
Electronics	Morph-Level	71.40	83.61
	Stem-Level	70.28	82.65
	Token-Level	**76.24**	81.87
	Review-Level	63.63	**86.66**
Kitchen	Morph-Level	70.83	81.68
	Stem-Level	67.70	80.73
	Token-Level	**75.17**	**81.73**
	Review-Level	68.03	79.86
BOUN 5-class	Review-Level	46.66	43.50

i.e. all node accuracy, we observed that RNTN performed better with the token-level annotated MS-TR.

8. Conclusions

This paper introduces a Morphologically Enriched Sentiment Treebank (MS-TR) for compositional semantics in Turkish. MS-TR was constructed based on the four different annotation levels, including morph level, stem level, token level, and review level using morphological features of the words as a semi-supervised annotation approach. Each annotation level of binary and fine-grained fully labeledparse trees has been constructed as a novel sentiment treebank. Experiments have been done using different domain datasets with Recursive Neural Tensor Networks (RNTN) and compared to conventional baseline methods, including Naïve Bayes(NB), Maximum Entropy (ME), and Support Vector Machines (SVM). The effect of using labeled stems and suffixes in MS-TR has also been investigated for each dataset. According to the experimental results, RNTN has outperformed conventional baseline methods for each MS-TR annotation level. It has been shown that our semi-supervised distant annotation approach can practically be used to construct a fully labeled sentiment treebank without the need for human labor while keeping sentiment information of words to construct structured input for Tree-RNNs. This study can be improved using enhanced tree-based deep learning architectures, including constituency or dependency parse trees. Particularly, the fine-grained train dataset for Turkish can be expanded for future studies.

Acknowledgements

This project was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK), 2214-A International Research Fellowship Programme (for PhD Students), and Fatih Sultan Mehmet Vakif University (FSMVU).

Funding

This research supported by The Scientific and Technological Research Council of Turkey (TUBITAK) [1059B141800193]; and Fatih Sultan Mehmet Vakif University.

Author details

Sultan Zeybek[1]
E-mail: szeybek@fsm.edu.tr
ORCID ID: http://orcid.org/0000-0002-1298-9499
Ebubekir Koç[2]
ORCID ID: http://orcid.org/0000-0002-9069-715X
Aydın Seçer[3]
ORCID ID: http://orcid.org/0000-0002-8372-2441
[1] Dept. of Computer Engineering Fatih Sultan Mehmet Vakif University Beyoglu, Istanbul, Turkey.
[2] Dept. of Biomedical Engineering Fatih Sultan Mehmet Vakif University Beyoglu, Istanbul, Turkey.
[3] Dept. of Mathematical Engineering Yildiz Technical University Davutpasa, Istanbul, Turkey.

Citation information

Cite this article as: MS-TR: A Morphologically enriched sentiment Treebank and recursive deep models for compositional semantics in Turkish, Sultan Zeybek, Ebubekir Koç & Aydın Seçer*, Cogent Engineering* (2021), 8: 1893621.

References

Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source nlp framework for Turkic languages. *Structure, 10*, 1–5.

Aksan, Y., Aksan, M., Koltuksuz, A., Sezer, T., Mersinli, Ü., Demirhan, U. U., Yilmazer, H., Kurtoglu, Ö., Atasoy, G., Ö Z, S., & Yildiz, I., "Construction of the Turkish national corpus (TNC)," in Proceedings of the 8th international conference on language resources and evaluation, Istanbul, Turkey LREC 2012, 2012

Baccianella, S., Esuli, A., & Sebastiani, F., "SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining," in *Proceedings of the seventh international conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), May 2010.

Baly, R., Hajj, H., Habash, N., Shaban, K. B., & El-Hajj, W. (2017). A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in Arabic. *ACM, Trans. Asian Low-Resource Lang. Inf. Process.* https://doi.org/10.1145/3086576

Bickel, B., & Nichols, J. (2005). Inflectional synthesis of the verb. In *The World Atlas of language structures.* (pp. 94–97). Max Planck Institute for Evolutionary Anthropology.The World Atlas of Language Structures Online

Bilgin, O., Cetinoglu, Ö. C., & Oflazer, K. (2004). Building a Wordnet for Turkish. Romanian Journal of Information Science and Technology

C. Aytekin, "An Opinion Mining Task in Turkish Language: A Model for Assigning Opinions in Turkish Blogs to the Polarities ," Journal. Mass Commun., vol. 3, pp. 179–198, 2013

Cambria, E., Havasi, C., & Hussain, A., "SenticNet 2: A semantic and affective resource for opinion mining and sentiment analysis," in *Proceedings of the 25th international Florida artificial intelligence research society conference, FLAIRS-25*, 2012, pp. 202–207.

Cambria, E., Olsher, D., & Rajagopal, D., "SenticNet 3: A common and common-sense knowledge base for cognition- driven sentiment analysis," *Proceedings of the national conference on artificial intelligence*, vol. 2, pp. 1515–1521, 2014.

Cotterell, R., Schütze, H., & Eisner, J., "Morphological smoothing and extrapolation of word embeddings," in *54th annual meeting of the Association for Computational Linguistics, ACL 2016 - long papers*, Berlin, Germany, 2016.

Dehkharghani, R., Saygin, Y., Yanikoglu, B., & Oflazer, K. (2016). SentiTurkNet: A Turkish polarity lexicon for sentiment analysis. Language Resources and Evaluation, 501, 667–685. https://doi.org/10.1007/s10579-015-9307-6

Dehkharghani, R., Yanikoglu, B., Saygin, Y., & Oflazer, K.. (2017). Sentiment analysis in Turkish at different granularity levels. Natural Language Engineering, 23 (4), 535–559. https://doi.org/10.1017/S1351324916000309

Demirtas, E., & Pechenizkiy, M., "Cross-lingual polarity detection with machine translation," in *Proceedings of the 2nd International Workshop on Issues of Sentiment Discovery and Opinion Mining, WISDOM2013 - held in conjunction with SIGKDD*, Chicago, USA, 2013.

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., & Xu, K., "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *52nd annual meeting of the Association for Computational Linguistics, ACL 2014 - proceedings of the conference*, vol. 2. Association for Computational Linguistics, Baltimore, Maryland, USA, 2014, pp. 49–54. [Online]. Available: https://www.aclweb.org/anthology/P14-2009

Erogul, U., "*Sentiment analysis in Turkish*," *Middle East Technical University, MSc Thesis, Computer Engineering*, 2009.

Gezici, G., & Yanıkoglu, B. (2018). Sentiment analysis in Turkish. In K. Oflazer & M. Saraclar (Eds.), *Turkish natural language processing* (pp. 255–271). Springer Publishing.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T., "Learning word vectors for 157 languages," in *LREC 2018- 11th international conference on language resources and evaluation*, Miyazaki, Japan, 2019.

Haspelmath, M., Dryer, M. S., Gil, D., & Comrie, B. (2005). *The World Atlas of Language Structures*. Oxford University Press.

Iyyer, M., Enns, P., Boyd-Graber, J., & Resnik, P., "Political ideology detection using recursive neural networks," in *52nd annual meeting of the Association for Computational Linguistics, ACL 2014 - proceedings of the conference*, Baltimore, Maryland 2014.

Kaya, M., Fidan, G., & Toroslu, I. H., "Sentiment analysis of Turkish political news," *Proceedings - 2012 IEEE/WIC/ACM international conference on web intelligence, Macau, China, WI 2012*, pp. 174–180, 2012.

Liu, S., Yang, N., Li, M., & Zhou, M., "A recursive recurrent neural network for statistical machine translation," in *52nd annual meeting of the Association for Computational Linguistics, ACL 2014 - proceedings of the conference*, 2014.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2015). The Stanford coreNLP natural language processing toolkit.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: an on-line lexical database . *International Journal of Lexicography*, *3*(4), 235–244. https://doi.org/10.1093/ijl/3.4.235

Mintz, M., Bills, S., Snow, R., & Jurafsky, D., “Distant supervision for relation extraction without labeled data,” 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*. Suntec, Singapore: Association for Computational Linguistics, August. 2009, pp. 1003–1011. [Online]. Available: https://www.aclweb.org/anthology/P09-1113

Coban, Ö. C., Özyer, B., & Özyer, G. T., “Sentiment analysis for Turkish Twitter feeds,” in *2015 23nd signal process. commun. appl. conf.*, Malatya, Turkey, May 2015, pp. 2388–2391.

Oflazer, K. (2014). Turkish and its challenges for language processing. *Language Resources and Evaluation*. https://doi.org/10.1007/s10579-014-9267-2

Ozyildirim, B. M., & Coban, O. (2018). An empirical study of the extreme learning machine for Twitter sentiment analysis. *International Journal of Intelligent Systems and Applications in Engineering*, *3*(6), 178–184. https://doi.org/10.18201/ijisae.2018644774

Pang, B., & Lee, L., “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” in *ACL-05-43rd annual meeting of the association for computational linguistics, proceedings of the conference, Ann Arbor, Michigan*, 2005.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends{ in Information Retrieval*. https://doi.org/10.1561/1500000011

Park, J. H., Xu, P., & Fung, P., “PlusEmo2Vec at SemEval-2018 Task 1: Exploiting emotion knowledge from emoji and #hashtags,” in *Proceedings of the 12th international workshop on semantic evaluation (SemEval-2018)*, Association for Computational Linguistics, 2018, pp. 264–272.

Read, J., “Using emoticons to reduce dependency in machine learning techniques for sentiment classification,” in *Proceedings of the ACL student research workshop*, ser. ACLstudent ‘05. USA: Association for Computational Linguistics, 2005, p. 43–48.

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y., “Semantic compositionality through recursive matrix-vector spaces,” in *EMNLP-CoNLL 2012-2012 joint conference on empirical methods in natural language processing and computational natural language learning, proceedings of the conference, Jeju Island, Korea*, 2012, pp. 1201–1211.

Socher, R., Manning, C. D., & Ng, A. Y., “Learning continuous phrase representations and syntactic parsing with recursive neural networks,” *Proceedings of the NIPS-2010 deep learning and unsupervised feature learning workshop*, 2010.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D., “Semi-supervised recursive autoencoders for predicting sentiment distributions,” in *EMNLP 2011 - conference on empirical methods in natural language processing, proceedings of the conference*, no. ii, 2011, pp. 151–161.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C., “Recursive deep models for semantic compositionality over a sentiment treebank,” in *EMNLP 2013-2013 conference on empirical methods in natural language processing, proceedings of the conference*, 2013, pp. 1631–1642.

Suttles, J., & Ide, N. (2013). Distant supervision for emotion classification with discrete binary values. In A. Gelbukh (Ed.), *Computational linguistics and intelligent text processing* (pp. 121–136). Springer Berlin Heidelberg.

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol*, *63*(1), 163–173. https://doi.org/10.1002/asi.21662

Turk, U., Atmaca, F., Ozates¸, S. B., Koksal, A., Ozturk, B., Gungor, T., & Ozgur, A., “Turkish treebanking: Unifying and constructing efforts,” in *LAW 2019-13th linguistic annotation workshop, proceedings of the workshop*, 2019.

Türkmenoglu, C., & Tantug, A. C., “Sentiment analysis in Turkish media,” in *ICML 2014, international conf. mach. learn. WISDOM’14, workshop issues sentim. discov. opin. mining, June 25th, Beijing*, 2014.

Vural, A. G., Cambazoglu, B. B., Senkul, P., & Tokgoz, Z. O., “A framework for sentiment analysis in Turkish: application to polarity detection of movie reviews in Turkish,” in *Computer and information sciences III: 27th international symposium on computer and information sciences*, E. Gelenbe & R. Lent, Eds. London: Springer London, 2013, pp. 437–445.

Yıldırım, E., Etin, F. S. C., Eryigit, G., & Temel, T. (2014). The impact of NLP on Turkish sentiment analysis. *TBV Journal of Computer Science and Engineering*.

Zeybek, S., “*MS-TR: A Morphologically enriched Turkish sentiment Treebank*,” vol. 1, 2020. [Online]. Available: https://data.mendeley.com/datasets/nz7vm5rchd/1