DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEVELOPING A NEW APPROACH IN NATURAL LANGUAGE UNDERSTANDING TO DETECT DEFECTIVE EXPRESSIONS FOR TURKISH

by

Atilla SUNCAK

June, 2022

İZMİR

DEVELOPING A NEW APPROACH IN NATURAL LANGUAGE UNDERSTANDING TO DETECT DEFECTIVE EXPRESSIONS FOR TURKISH

A Thesis Submitted to the
Graduate School of Natural And Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering

by

Atilla SUNCAK

June, 2022
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"DEVELOPING A NEW APPROACH IN NATURAL LANGUAGE UNDERSTANDING TO DETECT DEFECTIVE EXPRESSIONS FOR TURKISH"** completed by **ATİLLA SUNCAK** under supervision of **ASSIST. PROF. DR. ÖZLEM AKTAŞ**, and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

.....

Assist. Prof. Dr. Özlem AKTAŞ

Supervisor

.....	.....
Assist. Prof. Dr. Kökten Ulaş BİRANT	Assoc. Prof. Dr. Mete EMİNAĞAOĞLU
Thesis Committee Member	Thesis Committee Member
.....	.....
Prof. Dr. Ayşegül ALAYBEYOĞLU	Assoc. Prof. Dr. Kemal AKYOL
Examining Committee Member	Examining Committee Member

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Assist. Prof. Dr. Özlem AKTAŞ, for not only her excessive knowledge, great mentorship, guidance and invaluable suggestions along with this route, but also for her continuous support and encouragement even when I have desperate and hopeless times. It is a great luck and honor to be her Ph.D. student. Moreover, I would like to thank my thesis committee members, Assist. Prof. Dr. Kökten Ulaş BİRANT and Assist. Prof. Dr. Mete EMİNAĞAOĞLU for their sincere advices and endless supports.

I would like to express my special thanks and gratitude to my dear friends Hakan CAN and Sinan GÖKER for their endless patience and priceless helps for this study. Without their supports, this study would not be a reality. I also wish to convey my appreciations to my colleagues in Kastamonu University and to my friends for their support.

Besides, I would like to thank my family for their endless love, support and encouragement.

Atilla SUNCAK

DEVELOPING A NEW APPROACH IN NATURAL LANGUAGE UNDERSTANDING TO DETECT DEFECTIVE EXPRESSIONS FOR TURKISH

ABSTRACT

Defective expression is a grammatical term that refers both semantic and morphologic ambiguities in Turkish sentences. They are generally caused by misusing of a suffix in addition to absence or unnecessary use of an element in a sentence such as object, subject and etc. Having analyzed several studies related to this issue, it is found out that they are mostly performed by linguists by means of student questionnaires, tests or manual analysis by researchers. The absence of Natural Language Processing (NLP) studies related to this issue directed us to deal with this subject using computer technologies. However, grammatically demanding languages such as Turkish generally require rule-based and language-specific solutions especially in semantic problems. Rule-based systems have some major obstacles such as efficiency in processing, time consumption while development and intolerance for alteration in language. Machine learning models have made great advances in recent years, which led to unprecedented boost in NLP applications in terms of performance. In this thesis, we propose deep learning models of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) in addition to machine learning classifiers of k-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest (RF) to detect defective expressions in Turkish sentences. Experimental trials show that deep neural approaches come into prominence for detection in comparison to traditional classifiers. The study also reflects that due to having learning capability of long term dependencies, LSTM architecture will provide more promising results when amount of dataset is increased and more optimized. By being an original study in this field, this study is considered to make a great contribution to Turkish NLP and provides an excellent source for other researchers studying this area.

Keywords: Defective expression, deep learning, machine learning, NLP, semantic ambiguity, text classification, Turkish, word vector

TÜRKÇE İÇİN DOĞAL DİL ANLAMADA ANLATIM BOZUKLUKLARININ TESPİTİ İÇİN YENİ BİR YAKLAŞIM GELİŞTİRİLMESİ

ÖZ

Anlatım bozukluğu, Türkçe cümlelerdeki anlamsal ve biçimsel belirsizlikleri ifade eden dil bilgisel bir terimdir. Genelde cümledeki özne, yüklem, nesne gibi ögelerin gereksiz kullanımından veya hiç kullanılmamasından ya da eklerin yanlış kullanımından kaynaklanırlar. Literatürdeki çeşitli çalışmalar incelendiğinde, bu konu ile alakalı çoğunlukla dilbilimcilerin gerçekleştirdiği öğrenci anketleri ve kompozisyon analizleri ya da araştırmacıların yaptığı manuel analizler ortaya çıkmaktadır. Konu ile alakalı doğal dil işleme çalışmalarının olmaması, bizi bu konuyu bilgisayar teknolojileri kullanarak analiz etmeye yöneltmiştir. Ancak, Türkçe gibi dil bilgisel anlamda zorlu diller, özellikle anlamsal problemlerde kural tabanlı ve dile özgü çözümler gerektirir. Kural tabanlı sistemlerin ise işlem sırasındaki etkinliği, geliştirme sırasındaki zaman tüketimi ve dildeki değişime karşı adaptasyon problemleri gibi büyük engelleri mevcuttur. Makine öğrenmesi modelleri, son yıllarda büyük gelişmeler göstermiştir, bu gelişmeler ise doğal dil işleme uygulamalarında eşi görülmemiş bir performans artışı sağlamıştır. Bu tezde, anlatım bozukluklarının tespitinde derin öğrenme modellerinden LSTM ve CNN; makine öğrenmesi sınıflandırıcılarından da KNN, SVM ve RF modelleri önerilmiştir. Deneysel çalışmalar, derin öğrenme yaklaşımlarının, anlatım bozukluğu tespitinde makine öğrenmesi sınıflandırıcılarına göre daha ön plana çıktığını göstermektedir. Ayrıca bu çalışma, veri seti artırıldığında ve daha uygun hale getirildiği takdirde, uzun dönem bağımlılıklarının da öğrenme kabiliyetine sahip olduğundan LSTM mimarisinin daha iyi sonuçlar verebileceğini yansıtmaktadır. Tezin, bu alanda yapılan orijinal bir çalışma olması, Türkçe doğal dil işlemeye büyük bir katkı sağlayacak ve alanda çalışma yapan diğer araştırmacılara da iyi bir kaynak olacağı düşünülmektedir.

Anahtar kelimeler: Anlatım bozukluğu, derin öğrenme, makine öğrenmesi, doğal dil işleme, anlamsal belirsizlik, metin sınıflandırma, Türkçe, kelime vektörü

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 General

Language is incontrovertibly the most essential means for communication with others as well as expressing them our feelings and ideas (Sirbu, 2015). Improper use of language results in narrowness in meaning or ambiguities during communication, thus the opinions or ideas cannot be addressed clearly. Moreover, a proper interaction among society and being aware of the developments in this global world significantly rely on a proper communication (Usur Hızlı, 2004). In addition to these issues, recent innovations in social media and rapid increase in its use led to an acceleration in Natural Language Processing (NLP) studies such as sentiment analysis, machine translation, text normalization, robotic respond and knowledge extraction (Janiesch et al., 2021; LeCun et al., 2015).

In addition to being a method of artificial intelligence, NLP is a data mining technique that is benefited for the purpose of knowledge discovery from text or voice data. The main idea of NLP is to analyze, understand and evaluate the data with regard of the grammar rules of the natural language either semantically or morphologically using computer technologies. Associated with the rapid development of NLP studies, the interaction between human and computer has been increased significantly.

However, natural language is ambiguous because the language itself and its components such as words, contexts, spelling rules, grammar issues and etc. tend to alter frequently. That is why natural language is considered by linguists to be an alive existence and as a result, ambiguities which are either semantic or morphologic can be occurred during the use of the natural language, which makes NLP studies more compelling. On the other hand, it is important that a sentence must be in an understandable, open and clear form with having no unnecessary components in order to avoid ambiguity during narration our senses and thought. During generating a

computational NLP model, morphologically rich languages have their own language difficulties in terms of grammar rules (Sak et al., 2011; Vylomova et al., 2016). That kind of languages such as Turkish mostly have an agglutinative complexity (Özçift et al., 2021) which means a stem word can take multiple suffixes and generate new words whose meanings might be completely different from the original stem (Yıldırım et al., 2015). Those morphological features of Turkish lead to come across various ambiguities as well. In Turkish, all those morphological and semantic ambiguities are grammatically named as 'Defective Expressions'.

1.2 Problem Definition

Defective Expression is a grammatical issue which addresses both morphological and semantic ambiguities in Turkish sentences. This is a significantly crucial issue for Turkish people in the fields of mass media such as newspapers or TVs, literal publications such as books or magazines, educations at schools from primary schools to universities and even almost any competitive exams held in Turkey for attending high schools or universities (Çetinkaya & Ülper, 2015). All the aforementioned issues clearly show the importance of practicing Turkish language without defective expressions. In Turkish grammar, defective expressions can be separated in two categories (Yiğit, 2009) as given in Table 1.1 below.

Table 1.1 Defective expression types in Turkish

Semantic Defective Expressions	**Morphological Defective Expressions**
Using redundant word	Disagreement in subject and verb
Using semantically opposed word	Using incorrect suffixes
Using semantically incorrect word	Absence of element in the sentence
Using word in the wrong place	Missing verb
Using semantically incorrect idiom	Errors in determinative group
Uncertainty in the meaning	Using incorrect conjunction
Error in logic and order	

Misusing of the elements in a sentence such as subject, object or suffixes causes defective expressions, which are generally the types of morphological defective expressions. On the other hand, semantic defective expressions in the sentence are caused by misuse of words in terms of context or using them in the wrong places. In the following, there are two examples of defective expressions with their disambiguation correspondences, each of which are semantic and morphological.

- Mehmet, siyah pantalonunu giydi. *(Mehmet has worn **[his or your]** black trousers.)*
- **Elbette** Ayşe de sınavdan kalmış **olabilir**. *(**Definitely** Ayşe **might** have failed in the exam, too.)*

The first sentence has a morphologic defective expression of 'missing element in the sentence', because the possessed item of trousers is unclear. In order to disambiguate the sentence, the adequate possessive pronoun must be added to the sentence as follows:

- Mehmet, **senin** siyah pantalonunu giydi. *(Mehmet has worn **your** black trousers.)*
- Mehmet, **onun** siyah pantalonunu giydi. *(Mehmet has worn **his** black trousers.)*

The second sentence has a semantic defective expression due to using contrast words, because a properly made-up sentence cannot express the context of both definiteness (of course) and possibility (might) together.

- **Elbette** Ayşe de sınavdan kalmıştır. *(**Definitely** Ayşe has failed in the exam, too.)*
- Ayşe de sınavdan kalmış **olabilir**. *(Ayşe **might** have failed in the exam, too.)*

The studies related to defective expressions in Turkish are mostly performed by linguists or researchers of education field of study with questionnaires and

composition papers of students; some researchers analyzed newspapers in terms of defective expressions and all of these studies have been performed using manual analyze methods. On the other hand, the ambiguity studies of other natural languages are mostly related to semantic extraction of texts such as the documentations of requirement analysis in software engineering, however even if these studies have been performed using computer technologies such as rule-based NLP techniques and machine learning, these ambiguities are not a grammatical issue as in Turkish; they are generally caused by the misunderstanding of customer expressions in the document as customers are incapable of technical terms. To conclude, it is understood from the literature that other language ambiguities are not related to specific defective expressions of Turkish language, therefore the absence of studies related to Turkish defective expressions led us to apply NLP and deep learning approaches for this subject.

1.3 Objectives of the Thesis and Novel Contribution

For many years, rule-based techniques that define text using a series of linguistic guidelines into ordered categories have been used in many studies, however these techniques require extreme knowledge of grammar of the language for implementation. Moreover, it is mostly difficult to create rules for a complex structure due to the requirement of a lot of research and checking which also results in difficulties in maintaining and scaling as the pre-existing grammar rules tend to be altered by the new ones in time (Dogra et al., 2021). Furthermore, gradually increasing data causes loss in performance as handling them becomes a big problem for rule-based approaches.

Machine learning and artificial neural network (ANN) techniques have made unprecedented advances in recent years, which led to significant performance boost in NLP applications (Lauriola et al., 2021). They tend to learn meaningful relationships and patterns from previous examples and observations rather than relying on manually designed rules and codifying knowledge into computers. One of these

advancements is the evolution of deep neural network approaches having upgraded capabilities of learning, named as deep learning. Early models of NLP techniques were based on grammar rules, ontologies, or dictionaries, but today, deep learning approaches such as convolutional neural network (CNN) and long short-term memory (LSTM) have replaced them gradually to achieve better performance (Bahdanau et al., 2014).

This study suggests an approach to detect defective expressions in Turkish using deep learning techniques. However, as mentioned before, there has not been any specific 'disambiguation' study for Turkish in the literature which means that there is no specific dataset for the purpose of this study. Because a deep learning technique requires thousands of data to train the model, a new specific dataset needed to be created by manual sentence collection.

First of all, a dataset which consists of 9710 Turkish sentences that are collected in person from online sources has been created and all sentences are labelled as "NON-DEF" (not-defective) and "DEF" (defective) whether they have defective expression or not by manual analyze of each sentence. However, that amount of sentence is inadequate for a deep learning model to train, therefore the dataset was augmented up to 29,756 sentences by using the Turkish Synonym Dictionary (Aktas et al., 2013). After preprocessing the data which includes some basic NLP operations such as tokenization, omitting stop words, removing punctuations or numbers and text normalization, we created a corpus of embedding Turkish word vectors from this dataset using word2vec technique (LeCun et al., 2015). This is because recurrent neural network (RNN) models perform more accurately with word vectors in comparison to the sentences themselves. As for the code implementation, python programming language is used with the other essential NLP libraries such as Keras, Tensorflow and etc.

In this thesis, the main goal is to develop machine learning approaches to detect defective expressions in Turkish. For this purpose, deep learning approaches of LSTM and CNN in addition to traditional machine learning classification algorithms such as

support vector machine (SVM), random forest (RF) and k-nearest neighbor (KNN) have been implemented. The collected and augmented sentences have been used to create both the word embedding corpus for feature extraction and the dataset for both training and testing the models. Consequently, the semantic knowledge of Turkish is aimed at combining with the artificial intelligence techniques in order to generate better performance in comparison to rule-based approaches. Since being an original research, this study will be a useful resource for other researchers working in this field and make a crucial contribution for Turkish semantic NLP analysis, therefore we will leave behind one more step for the purpose of creating Turkish WordNet.

1.4 Organization of the Thesis

This thesis consists of six chapters and the organization of the thesis is explained in the following paragraphs.

In Chapter Two, literature review has been performed and recent studies related to defective expressions in both Turkish and other languages have been analyzed in order to gather more information for the purpose of this study.

In Chapter Three, defective expressions in Turkish grammar have been explained in details by providing defective Turkish sentence examples in each type with their disambiguated correspondences.

In Chapter Four, background information related to machine learning techniques is given briefly. Furthermore, dataset collection, data augmentation and word2vec technique for feature extraction is explained. Moreover, deep learning approaches (LSTM and CNN) and machine learning classifiers (KNN, SVM and RF) to be used in this thesis have been discussed excessively.

In Chapter Five, the flow of the classification algorithm has been described in details. After that, experimental trials for all the proposed models have been depicted separately and in the light of trials, the results have been analyzed in terms of model

performances.

Finally, in Chapter Six, concluding remarks of this thesis and possible future recommendations for more efficient learning model architectures have been presented.

CHAPTER TWO
RELATED WORKS

Defective expressions have always been a grueling issue especially for Turkish students during both their educational durations at school and almost any attendance exams of upper-level schools. Moreover, publishing a literal book, a magazine or even a newspaper requires a proper use of language which is free of defective expression. The literature analysis so far shows that apart from computer scientists, researchers of education science have analyzed this field of study by means of student questionnaires, book or newspaper analyzes, essays of students and etc., which are entirely based on manual inspections in person. On the other hand, there are several NLP studies related to the ambiguities of other languages analyzed by computer scientists, however these ambiguities are grammatically dissimilar in comparison to the sui generis defective expressions of Turkish. For this reason, the related works are categorized in two sections: Turkish defective expressions and other language ambiguities.

2.1 Turkish Defective Expressions

The MSc. Thesis of Büyükikiz (2007) analyzes the writing skills of 105 8^{th}-class students of elementary school, whose socio-economic levels vary among each other, in terms of syntax and defective expression in Turkish. According to the study, the students are required to write essays about one of the given eleven topics in order to inspect the structure of sentences and defective expressions. After the manual analyze of each sentence, it is found out that 311 sentences out of 1360 in total were determined to consist defective expressions of twelve separate kinds. Several suggestions were proposed in the light of findings for Turkish teachers to the development of writing skills of students such as teaching the meaning of a word considering its context in books, teaching the grammar should be cooperated including essay writing, encouraging student to read not only course books but also other literal books and etc.

The MSc. Thesis of Bahar (2006) aims at determining whether there exists any relationship between theoretical grammar knowledge of Turkish and written defective expressions. For the purpose of this study, 45 male and 45 female 8^{th}-class students from three separate elementary schools located in the city center of Uşak have been participated. First, they have been applied an exam of Turkish grammar to determine their grammar knowledge, then they have been asked to write an essay to measure their capability of using the grammar knowledge. After that, these each sentence of all essays have been analyzed in terms of defective expressions. As a result of the evaluation, the researcher proposes several beneficial suggestions related to grammar teaching such as simplification of the grammar topics from complex ones during basic education, teaching grammar topics not theoretically but by giving examples from text books, a course at universities for teacher candidates to provide them how to teach grammar in more efficient and enjoyable manner and etc.

The MSc. Thesis of Özdem (2012) deals with the local newspapers of Çanakkale in terms of use of Turkish language by analyzing the grammar mistakes and defective expressions in order to point out a proper use of Turkish in local newspapers and provide awareness about the significance of using the language plain and free of defections. For the purpose of this study, total 250 copies of nine daily and two weekly newspapers published on December in 2009 have been investigated in terms of defective expressions. After analyzing each sentence of these copies, the sentences including defective expressions have been determined. After that the frequency of defective expressions have been listed according to both their types and the newspapers they have. As for the result of this study, several suggestions in order to decrease the use of defective expressions have been proposed such as providing a better Turkish education to the student, providing an advanced language education to Turkish teacher candidates, establishing a language commission for the approval of all publications of mass communication tools and etc.

In the MSc. Thesis of Saydam (2016), The local newspapers of 2015 – 2016 years in Giresun have been analyzed in terms of defective expressions, punctuation errors and misspelling for the purpose of take awareness of using a proper Turkish in mass

media. For the purpose of the study, fourteen different local newspapers are read by the research group and the aforementioned situations have been assorted in order to specify their reasons. As for the result, all the mistakes have been corrected and the defective expressions have been disambiguated manually. As for the main goal, significant suggestions related to the misspellings and defective expressions have been proposed such as reading habit development in family, school and society, writing skill development for student, being more sensible about teaching Turkish as a whole for teachers and etc.

2.2 Ambiguities in Other Languages

The research of Ferrari & Esuli (2019) analyzes language-specific ambiguities in requirement analyses between stakeholders of a technical project. Stakeholders with the different background and skills need to have an efficient communication for understanding or addressing the problem in order to avoid linguistic ambiguity which can be occurred due to terminology conflicts. Furthermore, these ambiguities may lead to distrust between stakeholders and result in problems in later phases of development. To deal with this problem, two different NLP approaches of Language Model Generation and Cross-Domain Term Selection have been proposed to identify ambiguous terms using seven separate elicitation scenarios within five domains of interest and rank them by ambiguity score. To perform these approaches, a domain-specific language model is built to identify and word embeddings have been used to measure the ambiguity, depicted as Figure 2.1. The evaluations show that despite some acceptable accuracies of 81% or 88% in a few elicitations, the model application was not successful in general, therefore researchers point out further analysis requirement for future between natural language ambiguities and domain knowledge.

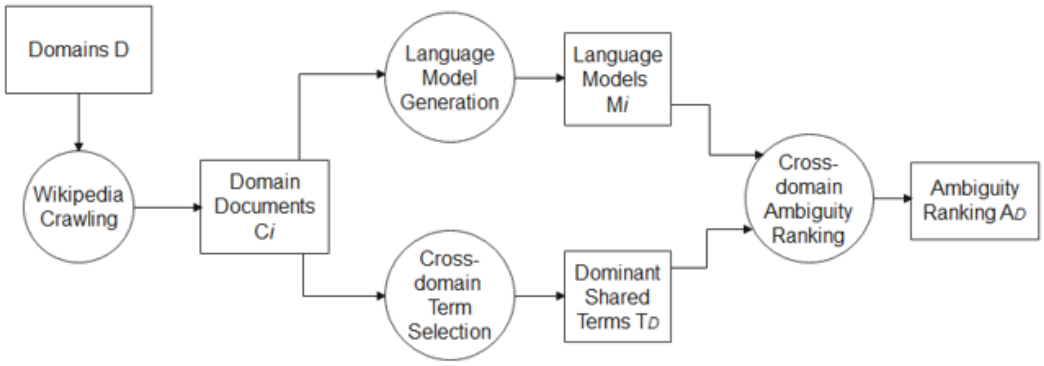

Figure 2.1 Cross-domain ambiguity measurement approach

Bano (2015) performed a review study related to ambiguities of natural language requirements in requirement engineering. To start with, the researchers of requirement engineering have been asked how to challenge the ambiguities in the documents, then a mapping study has been performed that focuses on NLP techniques for ambiguity detection. 174 studies published between 1995 – 2015 have been systematically reviewed and 28 of the have been selected to be analyzed for the purpose of the study. The analysis results of the papers show that 81% of the papers focused on ambiguity detection in addition to 4% of them for reducing and 5% of them deals with removing ambiguities. The results have also showed that there are still some significant gaps in empirical results which is also interpreted by the author that in requirement engineering, lack of empirical evaluation in NLP techniques for the purpose of addressing ambiguities is still an important issue to be handled.

Hoceini et al. (2011) proposed a method for disambiguation of non-vowel Arabic text data as a word may represent multiple meaning due to the ability of taking several forms. The goal of the study is to combine NLP with MCDA (Multiple Criteria Decision-Aid) by integrating textual data analysis of no-vowel Arabic texts into a decision making system when an ambiguity detected, shown in Figure 2.2. The approach is based on decision theory with NLP techniques for disambiguation such as probabilistic Hidden Markov Model, rule-based linguistic constraints, n-grams and etc. One of the advantages of this study is to reduce the dominant candidates of

ambiguities for a non-vowel word and rank the others by various criteria evaluations.

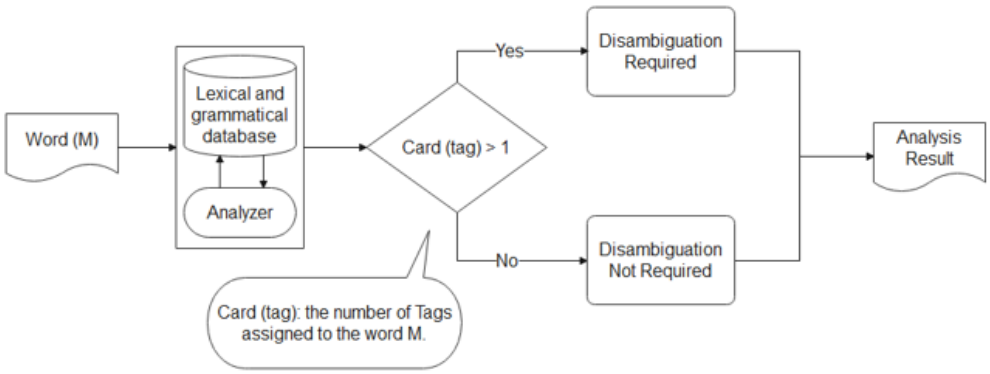

Figure 2.2 Disambiguation decision for a word in the model

The research of Staron et al. (2018) dealt with the ambiguities occurred in relative clauses and proposed an approach which combines NLP techniques that uses grammar knowledge with external data which provides further knowledge into account. According to the researchers, regular disambiguation techniques benefits only the language specific grammar knowledge to handle the problem, which generally results with unreliable solutions. The main idea is to generate a dependency trees by the combination of graph-based and grammar based dependency parsers. The combination of text with visual corresponding scene have proven the hypothesis of the researchers by analyzing the corpus of Language and Visual Ambiguities (LAVA) (Berzak et al., 2016) which contains 237 ambiguous English sentences with their corresponding short videos or visual images. The example sentence of “The woman carves the head of the bed, which the man paints.” is ambiguous due to being interpreted as the man is either painting ‘the head of the bed’ or ‘the bed’. The dependency trees and visual scenes for both meanings are presented respectively in Figure 2.3 and Figure 2.4.

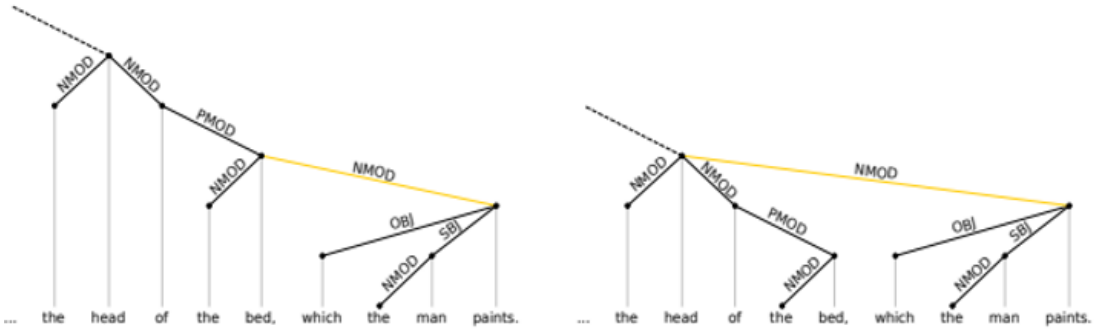

Figure 2.3 Dependency trees for the two interpretations (Staron et al., 2018)

Figure 2.4 The visual scenes for both interpretations (Staron et al., 2018)

In the study of Shirin et al. (2018), an approach that detects idioms in the sentence and replaces it with its correspondent figurative phrase has been proposed using an online idioms dictionary. According to the authors, idioms are categorized as ambiguous due to involving different meaning in terms of context than their literal meaning, therefore this situation makes idioms a great challenge during NLP operations such as machine translation. The approach includes five sequential phases, shown in Figure 2.5. The first phase of the approach is 'Idiom Extraction' using a python package 'ICE (Idiom Collocation Extractor) by several NLP techniques such as tokenization of the input sentence into bigrams or trigrams in order to determine whether they are idiomatic expressions or not. If the meaning of the individual words of each phrase is different in terms of context, then they are accepted as idioms. The next step is 'Usage identification' which determines whether that idiom is figurative of literal by using a couple of similarity measurements such as lexical similarity and

topical similarity. After that, 'Definition Extraction' step is performed, which is the meaning extraction of the idioms using an online idioms dictionary. Next step is 'Substitution Generation' which extracts the relevant replacement for the idiom using rule-based and equal-POS methods. As for the final step, 'Post Editing' completes the approach by generating meaningful and compatible paraphrase of the original input sentence when necessary.

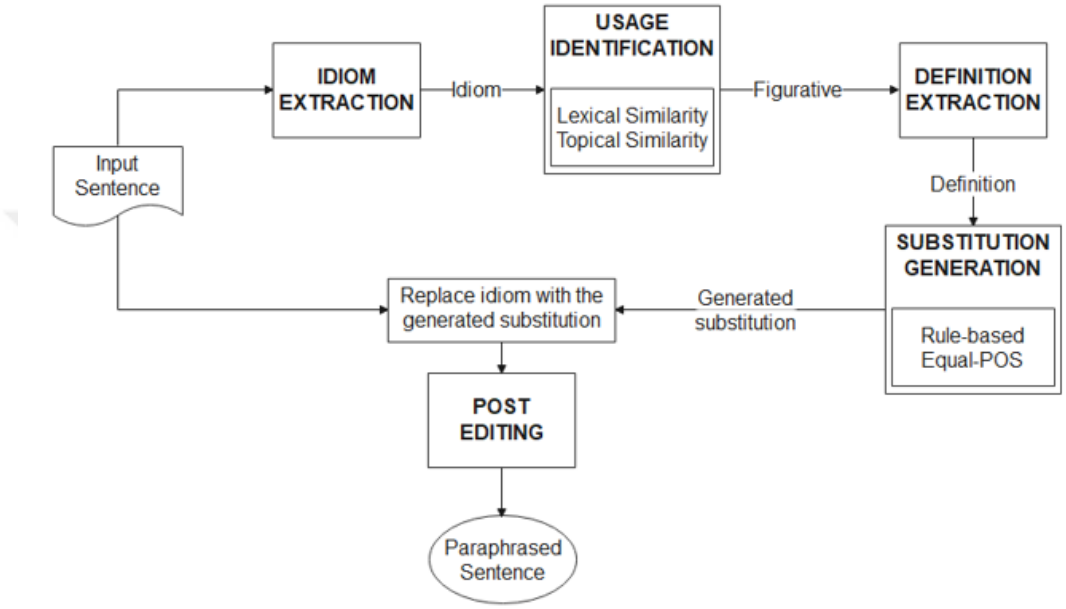

Figure 2.5 The flow of the idiom detection approach

The study of Elkahky et al. (2018) focuses on POS Tagging in the English words which are literally both verb and noun such as 'mark', 'flies (conjugated verb, plural noun)' and etc. According to the authors, most of the treebanks used for POS tagging consider words itself without the context they have, which results in ambiguities. This study proposes a data collection of noun-verb ambiguity having more than 30,000 manually labeled examples, thus the POS-tagging using this dataset can be handled by taking into account the context of the word. As a result of this study, this improvement in the POS-tagging is intended to increase the efficiency on disambiguation tasks for text-to-speech.

The study of Kuchta et al. (2018) analyzes the software requirements for concept extraction in order to automate software development processes. Expressing the

requirement specifications is a challenging issue because natural language can be ambiguous, therefore the rule-based methods that have been used before such as statistical analysis, ontological matching expressions and etc. were not accurate enough to handle the problem. This study proposes an approach to extract concepts and class structure of requirements in English by analyzing them grammatically without language specific ontology. The basic idea is to divide texts into paragraphs, then into sentences and then into tokens. After that, POS-tagging is handled using maximum entropy model, however this operation is ambiguous as the verification results showed a lot of errors. Therefore, an online English dictionary is used to disambiguate the results. Next, phrase detection is performed, because the concepts are not only nouns, they may be occurred by phrases as well. For this reason, all nouns and adjective-noun sequences are extracted and converted into their basic forms (root) for phrase detection by using WordNet. To classify, aggregate and disambiguate the meanings of phrases, the authors also benefited hypernym-hyponym, holonym-meronym (whole-part) and domain-member relationships.

Mahadzir et al. (2018) proposed two path-based and three information-content based semantic similarity measurements in order to disambiguate the words belonging to two separate languages of Malay and English literally. In bilingual society, the two languages are spoken together during conversations, moreover a sentence can be built by mixed words from either languages, which causes ambiguities for NLP operations. This complication makes NLP studies even more difficult as the word becomes ambiguous due to the determination of its belonging language. For handling this problem, the researchers generated a test collection of word pairs with their labels of belonging language. The first word is the one that has to be determined, and the second word is the related word of the first one contextually. After that, those five measurements have been applied to these pairs, and the dominant results are compared with the language labels.

CHAPTER THREE
DEFECTIVE EXPRESSIONS IN TURKISH GRAMMAR

In this chapter, the defective expression types in Turkish grammar are explained briefly with sample sentences and their possible disambiguated correspondences. As having explained shortly in the first chapter, even though they are written correctly in terms of spelling, defective expressions causes ambiguities in Turkish sentences either semantically or morphologically. Despite the fact that defective expressions are not categorized as same among the sources, one of the suitable categorization, which this study obeys as well, is depicted as given in Table 3.1 below.

Table 3.1 Defective expression types in Turkish

Semantic Defective Expressions	**Morphological Defective Expressions**
Using redundant word	Disagreement in subject and verb
Using semantically opposed word	Using incorrect suffixes
Using semantically incorrect word	Absence of element in the sentence
Using word in the wrong place	Missing verb
Using semantically incorrect idiom	Errors in determinative group
Uncertainty in the meaning	Using incorrect conjunction
Error in logic and order	

3.1 Semantic Defective Expressions

3.1.1 Using Redundant Words

This defectiveness is occurred in case of using two words that are synonyms or the meaning of a word can also be consisted by another word. To disambiguate, one of the words must be omitted from the sentence.

- *Arkadaşım kulağıma **sessizce fısıldadı**. (My friend **whispered** me **quietly**.)*

  - The verb ‘whisper’ consist the meaning of ‘quiet’, therefore the disambiguated sentence should be as follows:
  - *Arkadaşım kulağıma fısıldadı. (My friend whispered me.)*

- *Babam **henüz hala** işe gitmedi. (My father has not **still** gone to work **yet**.)*
  - The adverbs ‘hala [still]’ and ‘henüz [yet]’ are synonym words for this sentence, therefore one of them should be omitted.
  - *Babam henüz işe gitmedi. (My father has not gone to work yet.)*
  - *Babam hala işe gitmedi. (My father has still not gone to work.)*

3.1.2 Using Semantically Opposed Word

In order to avoid conflicts, contrast words must not be used together to express the intended idea in a sentence. Omitting the correct word disambiguates the sentence.

- ***Şüphesiz** Ahmet de Zonguldak’a gitmiş **olabilir**. (**There is no doubt** that Ahmet **might** have gone to Zonguldak too.)*
  - The two expressions have contrast meaning and spoil the semantics of the sentence.
  - *Şüphesiz Ahmet de Zonguldak’a gitti. (There is no doubt that Ahmet have gone to Zonguldak too.)*
  - *Ahmet de Zonguldak’a gitmiş olabilir. (Ahmet might have gone to Zonguldak too.)*

3.1.3 Using Semantically Incorrect Word

Confusing the words which are not semantically the same but similar in terms of meaning causes defectiveness in a sentence.

- *Yetkililerin yanlış yatırımları, şirketin iflasını **sağladı**. (Wrong investments of the authorities **provided** bankruptcy of the company.)*
  - The positive and negative results are expressed by different words in cause-and-effect-relation sentences in order to avoid ambiguity. The above sentence, a negative result is intended to expressed, however the verb 'provide' is used in a positive manner. To disambiguate the sentence, the verb 'caused' can be replaced with 'provide'.
  - *Yetkililerin yanlış yatırımları, şirketin iflasına neden oldu. (Wrong investments of the authorities caused bankruptcy of the company.)*

3.1.4 Using Word in the Wrong Place

Disordered words in a sentence causes defective expressions.

- *Dünden beri **çok** karnım ağrıyor. (I have **severely** stomachache since yesterday.)*
  - The word 'çok [severely]' is intended to expressed the severity of the pain, however it is used such an adjective that does not describe the severity.
  - *Dünden beri karnım çok ağrıyor. (I have stomachache severely since yesterday.*

3.1.5 Using Semantically Incorrect Idiom

Idioms (and proverbs as well) are phrased expressions which provide metaphoric meanings and ideas differently from their literal ones. Misusing them in terms of meaning or replacing one or more words with their synonyms or similar ones cause defective expressions.

- *Semih'in bana yaptığı iyiliklere daima **göz yumdum**. (I always **ignored/tolerated** all the favors Semih did to me.)*

- ‘Göz yummak [Literally ‘to close eye’]’ is an idiom which has the meaning of ‘ignoring/tolerating a bad action of someone’. However, this sentence describes a positive idea, therefore the idiom is misused in the sentence.
- *Semih’in bana yaptığı iyiliklere daima **minnettarım**. (I always have been **grateful** for all the favors Semih did to me.)*

3.1.6 Uncertainty in the Meaning

In the literature, this type of defectiveness can also be expanded as ‘comparison mistakes’, ‘punctuation related mistakes’ and etc. In Turkish, the pronouns of the actions (subjects) and possessive pronouns of nouns are not needed to be used while writing or speaking, because suffixes of the verbs and nouns are capable of providing them. Sometimes not using them causes ambiguities in the sentences. On the other hand, misusing or not using the punctuation (mostly the comma) results in defectiveness as well.

- *Zeynep **cüzdanını** okulda unuttu. (Zeynep forgot **his-her/your purse** at school.)*
  - **Cüzdan** (root) + **ı** (accusative) + **n** (possession) + **ı** (accusative)
  - The suffix of ‘-n’ provides possession of the noun ‘cüzdan [purse] in correspondences to the both possessive pronouns of ‘your’ and ‘his-her’. To disambiguate the sentence, the intended possessive pronoun must be added to the sentence.
  - *Zeynep **senin** cüzdanını okulda unuttu. (Zeynep forgot **your** purse at school.)*
  - *Zeynep **onun** cüzdanını okulda unuttu. (Zeynep forgot **her** purse at school.)*

3.1.7 Error in Logic and Order

- *Bilgisayar kullanmayı bırak, Ali daha kod bile yazamıyor. (Forget about using a computer, Ali cannot still implement a code.)*

  - Coding is harder or more complicated than using a computer, however it was described on the contrary.
  - Ordering Error

- *Gösterinin ilk gününde rekor düzeyde katılım gerçekleşti. (The attendance level of the show broke a record on the first day.)*
  - Logically, breaking record requires comparison with something.
  - Logic Error

3.2 Morphological Defective Expressions

3.2.1 Disagreement in Subject and Verb

Inconsistency between verb and subject in terms of singularity-plurality or positivity-negativity leads to defective expressions in sentences.

In Turkish, when the subject is 'human', then the verb can be either singular or plural according to the subject. On the other hand, 'non-human' subjects always require singular verbs no matter if the subject is singular or plural.

- ***Biz*** *okula gidiyoruz. (**We are going** to school.)*
  - The sentence is correct.
- ***Ben*** *okula gidiyorum. (**I am going** to school.)*
  - The sentence is correct.
- ***Kuşlar*** *gökyüzünde **uçuyorlar**. (**The birds are flying** in the sky.)*
  - Because 'birds' are non-human existences, the verbs must be singular.
  - ***Kuşlar*** *gökyüzünde **uçuyor**. (**The birds is flying** in the sky.)*

While some indefinite pronouns and some conjunctions require only positive verbs, other ones require only negative verbs similarly with English grammar.

- ***Ne** Ayşe **ne** de Fatma bugün okula **gelmediler**. (**Neither** Ayşe **nor** Fatma **have not come** to school today.)*
  - ***Ne** Ayşe **ne** de Fatma bugün okula **geldiler**. (**Neither** Ayşe **nor** Fatma **have come** to school today.)*
- ***Herkes** yardımcı oluyor; şikayet etmiyordu. (**Everyone** was helping, was not complaining.)*
  - The indefinite pronoun 'herkes [everyone]' requires only positive verbs. When a negative action follows the first expression, then another proper indefinite pronoun must be added to the sentence.
  - *Herkes yardımcı oluyor, **kimse** şikayet etmiyordu. (Everyone was helping; **no one** was complaining / **anyone** was not complaining.)*

3.2.2 Using Incorrect Suffixes

Because of being an agglutinative language, using suffixes for forming new words depict significant importance in Turkish, determining the case of the word such as accusative or dative, generating singularity or plurality, conjugation of possession and etc. For this reason, use of wrong suffix results in defective expression in the sentence.

- *Öğrenciler kitap **okumasını** çok severler. (Students like **reading** of book a lot.*
  - oku (verb) + ma (noun_making_suffix) + sını (possession + accusative)
  - The suffix of 'sını' is intended to specify the love of reading book, however misusing the suffix resulted as if 'reading' is possessed by the book. To disambiguate the sentence, the proper suffix use must be as follows:
  - *Öğrenciler kitap **okumayı** çok severler. (Students like **reading** book a lot.)*

3.2.3 Absence of Element in the Sentence

A missing element which must be occurred in sentence causes defectiveness due to semantic restriction. The defectiveness caused by subject and verb is analyzed in separate sections, therefore this section deals with direct / indirect objects, adverbs and etc. It can be also said that this kind of defective expressions are occurred in joint and ordered sentences in general.

- *Hakan **derslerini** oldukça **önemser** ve çok **çalışır**. (Hakan **cares** about **his lessons** pretty much and **studies** a lot.)*
  - Ordered sentence (Two sentence is combined with a punctuation mark.)
  - According to Turkish grammar, while the verb 'önemsemek [care]' requires accusative case of the noun, the verb 'çalışmak [study]' requires dative case. However, the commonly used noun 'dersler [lessons]' for both verbs is in the case of accusative and only suitable for the verb 'önemsemek [care]', but not 'çalışmak [study]'. To disambiguate the sentence, we must separately add the dative form of the noun 'dersler [lessons]'.
  - *Hakan **derslerini** oldukça önemser ve **derslerine / onlara** çok çalışır. (Hakan cares about **his lessons** pretty much and studies **his lessons / them** a lot.)*
- *İnsanlar **gazetelere** inanmıyor, bu nedenle de çok az okuyorlar. (People do not believe **the newspapers**, therefore they read very little.)*
  - Joint sentence. (Two sentence is combined with a conjunction word.)
  - The same situation is happened with the previous sentence. The verb 'inanmak [believe] requires dative noun, while the verb 'okumak [read]' requires accusative one. As the noun 'gazeteler [newspapers]' is in the dative form, it is only suitable for the verb 'inanmak [believe]'. In order to disambiguate the sentence, the accusative form of the noun must be added to the second sentence as follows:

– *İnsanlar **gazetelere** inanmıyor, bu nedenle de **gazeteleri / onları** çok az okuyorlar. (People do not believe **the newspapers**, therefore they read **the newspapers / them** very little.)*

3.2.4 Missing Verb

For every language, verb is the essential element of the sentence. The defective expression of missing verb expresses the missing verbs in the first sentences of ordered and joint sentence types.

- *Sen kendi kitabını, ben de kendi kitabımı **okuyorum**. (You **am reading** your own book, and **I am reading** my own book.)*
  - In the parenthesis, 'You am reading' is intentionally translated like that, because two sentences which are joint with a comma have only one verb (okuyorum [am reading]) in common and it is conjugated for the subject of 'ben (I)'. In order to disambiguate the sentence, a properly conjugated additional verb must be added to the first sentence.
  - *Sen kendi kitabını **okuyorsun**, ben de kendi kitabımı **okuyorum**. (You **are reading** your own book, and I **am reading** my own book.)*

3.2.5 Error in Determinative Group

Determinative groups refer to 'noun + noun' or 'subject + noun' expressions. This kind of defective expressions generally occur when there are two consecutive determinative groups in a sentence where their second noun (determinated noun) is in common.

- *Kamu ve özel **kuruluşlar**..... (Public and private **institutions** .....)*
  - According to Turkish grammar, each determinative groups are written such as 'kamu kuruluşları' and 'özel kuruluşlar'. As seen in the sentence, the

determinated word of ‘kuruluşlar’ is only compatible with the determining word of ‘özel’, but not with the noun ‘kamu’.

- *Kamu kuruluşları ve özel kuruluşlar……. (Public institutions and private institutions ……)*

3.2.6 Using Incorrect Conjunction

In the sentences, conjunction words must be used according to their semantics and misusing them cause ambiguities in the sentence.

- *Ders çalışmayı çok önemserim **ama** ödevlerimi hep zamanında yaparım. (I care about my lessons very much; **however**, I always do my homeworks on time.)*
  - The two sentences have cause-and-effect relationship, however using a conjunction which comprises contrast meaning resulted in defective expression in the sentence. To disambiguate the sentence, the conjunction word must be replaced with a proper one.
  - *Ders çalışmayı çok önemserim **bu yüzden / ayrıca / bundan dolayı** ödevlerimi hep zamanında yaparım. (I care about my lessons very much; **therefore / moreover / thus**, I always do my homeworks on time.)*

CHAPTER FOUR
MATERIALS AND METHODS

In this section, the technical basics and components of the thesis are explained in details. To start with, a brief background information is given about the approach and requirements. After that the structure of dataset and word embedding is explained briefly. Finally, the model approaches used for this study is mentioned in details.

4.1 Background

Deep learning is one of the most popular state-of-art technique which is benefited for several purpose of studies due to the capability of training the artificial intelligence model. In order to estimate the output from the input data, deep learning models require no rule-based configuration and implementation for NLP operations. This situation made a significant increase of deep learning models by NLP researchers since a natural language is an alive existence and the grammar rules inevitably tend to alter from generation to generation.

One of the main constraints of deep learning is the amount of data for training and validating (testing) the model. In spite of the fact that there are several sources that provides datasets having a big diversity of study areas, yet the amount of data in the dataset might not be sufficient enough for the model, therefore researchers augment the data in order to increase the model performance.

Another important issue for a deep learning model to get better accuracy results is the feature extraction from the input data. There are several techniques to extract beneficial features from data depending on the data type. Apart from regular NLP techniques such as POS-tagging, Named Entity Recognition or n-grams; one of the most popular and beneficial feature extraction technique for NLP purposed deep learning studies is vectorising the words, named as word embeddings. They are basically the weighted vector correspondences of words, which provides semantic context information of each word with its surrounding ones in a dataset.

In the following sections, the dataset collection and augmentation operations, feature extraction technique, deep learning models and machine learning classifiers used in this study are explained in details.

4.2 Dataset

Input data are the most crucial factor for a deep learning technique for training and validating the model. Moreover, the amount of data is also another issue in order to get the optimum accuracy, because even if the model is configured perfectly by adjusting the optimum hyper-parameters, the inadequacy of train data would affect the model performance negatively. For the purpose of this study, a set of labelled input sentences have been used as dataset which includes 29756 sentences. In the following, the processes of data collection, augmentation and preparation are explained in details.

4.2.1 Data Collection

In NLP studies, despite being tough, time consuming and grueling, data collection is one of the most significant process for a better development and performance (Kumhar et al., 2021). Moreover, the success of a deep learning model relies on the quantity and quality of the input data, thus a dataset including a great number of sentences with their labels was the inevitable need. For this reason, a comprehensive research was held, however it was understood that in the literature there has never been performed a dataset collection before that includes Turkish sentences having defective expressions. Therefore, we had to collect all the sentences and label them individually whether they have defective expressions or not by analyzing each of the sentences.

To begin with, an approximate number of fifty separate sources such as open-access websites of education centers, schools and courses in addition to the official exam center of Turkey (Center for Assessment, Selection and Placement - ÖSYM) have been analyzed in a three-month duration since defective expressions are one of the main subjects of Turkish tests in almost any kind of attendance exams. As the result of the

analysis, 9710 sentences related to defective expressions have been collected one by one. After that, all the sentences were determined individually one by one whether they have defective expressions or not; 4299 of the sentences have been labelled as 'DEF' as they have defective expressions in addition to 5411 of them as 'NON-DEF' because they are proper sentences having no defective expressions, as depicted in Table 4.1 below.

Table 4.1 Sample sentences in dataset

Sentence	**Label**
Ekonomi ile alakalı yapılan haberler, her şeyi ortaya koymaktadır.	NON-DEF
Yarın kesinlikle bu görüşme gerçekleşebilirmiş.	DEF
Herkes işini doğru yaparsa, ortada problem kalmaz.	NON-DEF
Derslerindeki başarıları, onun okul birincisi olmasına neden oldu.	DEF
Ortada, karamsar olmayı gerektirecek bir durum yoktu.	NON-DEF
Özel ve kamu kuruluşları yarın tatil olacak.	DEF
Öğrencilerin okula gelmesiyle koridorlarda şen bir hava esmeye başladı.	NON-DEF

Due to the fact that deep learning models require an excessive number of input data for the sake of a better performance and learning capability and the amount of collected sentences is definitely insufficient for train and test operations, a data augmentation operation is held before preprocessing the input data. This must be strongly pointed out that since there has never been any dataset study related to defective expressions performed previously in the literature, this study makes a great contribution for this field of study.

4.2.2 Data Augmentation

It is controvertible that the success of a learning model highly relies on the input data that trains it. In spite of being low quality, the sufficient number of input data provides better results in comparison to the high quality but insufficient number of

data. Moreover, augmenting the data is one of the main techniques that reduces or prevent overfitting, which means that the model stops learning at some time of the training process and starts memorizing that must be strictly avoided. In the literature, it is easily said that lots of the studies performed a data augmentation for several reasons such as better performance, avoiding overfitting, collecting insufficient data and etc.

In NLP purposed machine learning studies, there have been several data augmentation techniques performed such as back translation, random swap, random insertion, random deletion and etc. However, when studying the semantics of a sentence, these techniques may cause problems since the context of each word in sentence must be preserved as much as possible. Whereas, synonym replacement technique is a better approach for data augmentation as a word and its synonym correspondence have the same context in terms of meaning. Therefore, we used Turkish Synonym Dictionary (Aktas et al., 2013) to get the synonym correspondences of the words in our dataset, as depicted in Figure 4.1 below.

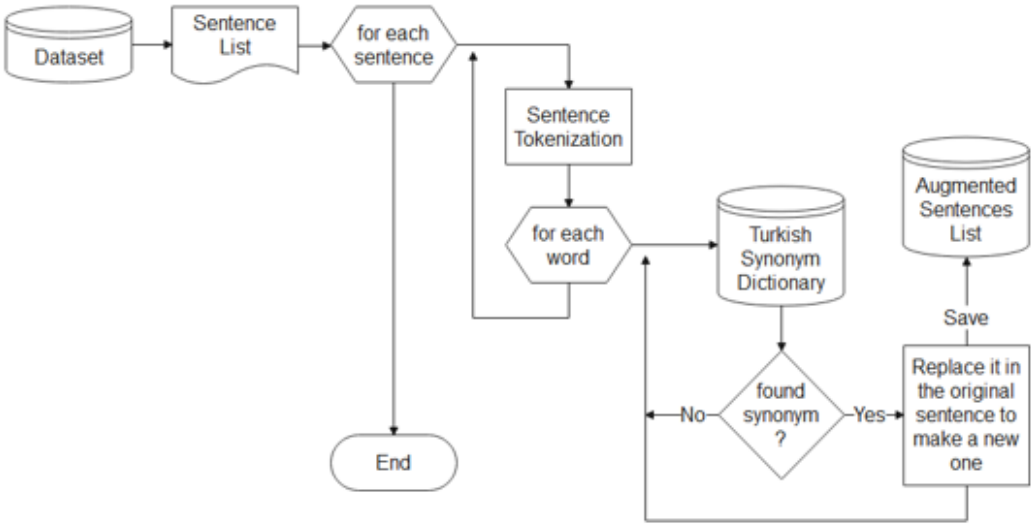

Figure 4.1 The flow of data augmentation using Turkish Synonym Dictionary

The flow of augmentation algorithm is explained as follows:

- Each sentence from dataset is split into words, which is called tokenization.

- The sentence itself is hold as a whole in a variable before tokenization.
- Each word of the sentence is searched for its correspondent synonym one in the Turkish Synonym Dictionary.
- When found, the synonym word in the dictionary is replaced with the original word in the sentence that is hold, and added to a file.
- Then these operations are applied to other words of the sentence.
- Then these operations are applied to other sentences of the dataset.
- As a result, a unique sentence can generate at least 3 to 5 new sentence depending on the word count.

The augmentation process is shown on an example sentence, *"Bu güz harika bir tatil yapacağız. (We are going to have a great holiday in this fall.)"* below, the synonym words are listed in Table 4.2:

Table 4.2 Sample of the words in the sentence with their correspondent synonyms taken from Turkish Synonym Dictionary

Word	**Synonym**
güz (fall)	sonbahar (autumn)
harika (great)	muhteşem (magnificent)
tatil (holiday)	dinlence (vacation)

- Bu **sonbahar** harika bir tatil yapacağız. *(We are going to have a great holiday in this autumn.)*
- Bu güz **muhteşem** bir tatil yapacağız. *(We are going to have a magnificent holiday in this fall.)*
- Bu güz harika bir **dinlence** yapacağız. *(We are going to have a great vacation in this fall.)*

At the end of the loop of each sentence, the label of that sentence has been applied to the correspondent augmented sentence so that the new list of augmented sentences would fit to the original dataset perfectly. As a result of the entire operation, the sentences to be used as input data as a whole increased up to 29756; 13398 of them are tagged as DEF, which means they have defective expressions and the rest of the 16358 sentences are tagged as NON-DEF, that is they are proper sentences in terms of defectiveness.

4.2.3 Data Preparation

Data preparation or data preprocess is the process that before training the model with input data, some NLP techniques are applied to them in order to clean the text and increase the quality for the sake of better performance of the deep learning approach. In this study, in order to prepare the sentences for training, the first operation was removing the punctuations and numbers. After that, all the sentences have been turned into lowercase. Then, same sentences have been removed from the list. Moreover, stop-words have been omitted from each sentence and finally normalization is performed such as multiple blank spaces, removal of non-Turkish words and etc. The flow is depicted in Figure 4.2 below.

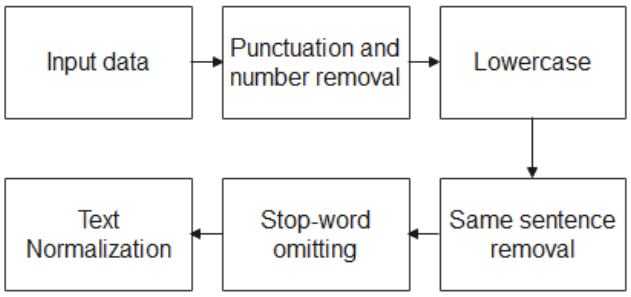

Figure 4.2 The flow of data preparation

On the other hand, this issue must be pointed out that we did not perform a stemming operation intentionally. Since being an agglutinative language; in Turkish, a word can be generated from a root or stem when they get suffix(es) and the

generated word can gain a completely different meaning in comparison to its root or stem. Furthermore, a defective expression may be occurred by using wrong suffix. In the light of these issues, stemming a word may result in losing the meaning of the word, therefore stemming has not been performed in this dataset.

4.3 Word Embedding for Feature Extraction

Word embedding is a term that refers to using vectors to represent document vocabulary (Muhammad et al., 2021). Word2vec is one of the techniques that vectorizes the words, introduced by Mikolov et al. (2013). This technique creates word vectors by considering the context of the reference words in the sentence by two separate algorithms: skip-gram (SG) and continuous bag-of-words (CBOW). CBOW algorithm estimates the target word considering the surrounding words (context) and skip-gram, which notably performs better for infrequent words, predicts the context using the target word (Fang et al., 2021). The architectures of CBOW and skip-gram algorithms are shown in Figure 4.3 below.

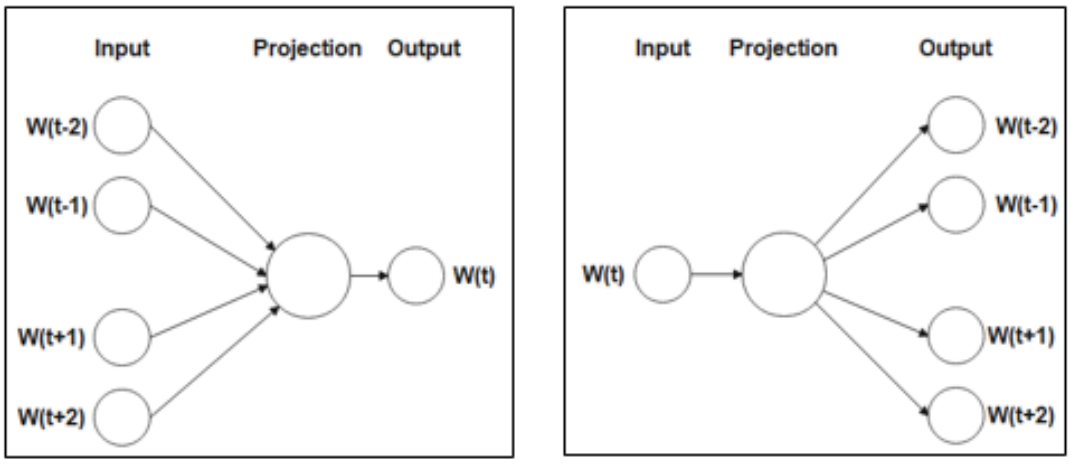

Figure 4.3 The architectures of CBOW (left) and Skip-Gram (right) algorithms

With word2vec technique, the relationship of words in a document can be extracted by transferring each word into a fixed-size numeric-value vector; named as word embedding. Word embeddings boost the deep learning model to measure the

distance relations of each words in terms of semantics or morphological in the document using cosine similarity. Moreover, they are capable of shortening training time, which leads to study more dimensions in vectors and bigger corpuses. The bigger the corpus is and the more dimension a vector has, the more accurate a vector can extract the context and relationships. To sum up, word embeddings are one of the most beneficial feature extraction technique in NLP studies. In the literature, word embeddings have been used in several NLP studies such as recommendation systems, machine translation, text classification and etc.

In this thesis, word2vec technique has been used as a feature extraction method of the input data due to providing successful semantic relationships among words in a sentence. First of all, all the words in the dataset have been tokenized, and then maximum length of the sentence in the dataset is calculated. After that, word2vec approach is applied to each word using CBOW algorithm to generate the corpus file with 200-dimension vectors. The window size is adjusted as 5, which means during calculation of a word vector, the surrounding before and after 5 words of the target word is determined as context and handled. The resulted corpus used for feature extraction provides the embedding matrix of the study which trains the deep learning models.

4.4 Deep Learning Approaches

This thesis proposes two deep learning approaches for detecting defective expressions in Turkish sentences; LSTM and CNN which is generally benefited when the input data is image files, however using 1-dimensional CNN (Conv1D) performs successfully with text data. These two approaches are explained in the following sections in details.

4.4.1 Long Short-Term Memory (LSTM)

Deep learning architectures can process complex patterns of time series and change their internal variables using back propagation techniques while conventional artificial neural networks can only performs data process in a raw form, which makes deep learning a state of the art method (Nourani & Behfar, 2021). Recurrent neural network (RNN) is a kind of deep neural architecture and RNNs are widely used due to the capability of predicting time-series since they are capable of using previous time steps for predicting current information. LSTM is the most appropriate network in terms of handling long sequence of data among RNNs (Zaytar & El Amrani, 2016; Zhang et al., 2018), introduced by Schmidhuber and Hochreiter (1997).

LSTM avoids long terms dependencies and vanishing gradient problems, which is major obstacles of regular RNNs, since it can decide whether to forget or remember the information by using its own forget gate and memory cells (Gers et al., 2000). That is why LSTM is used extensively for NLP tasks with time-variant data such as handwriting recognition (Graves et al., 2008; Graves & Jaitly, 2014), machine translation (Schmidhuber, 2015; Wu et al., 2016) and etc. The architecture of LSTM is depicted in Figure 4.4, where 'H' refers to hidden state vector, and 'X' for input vector of the LSTM unit.

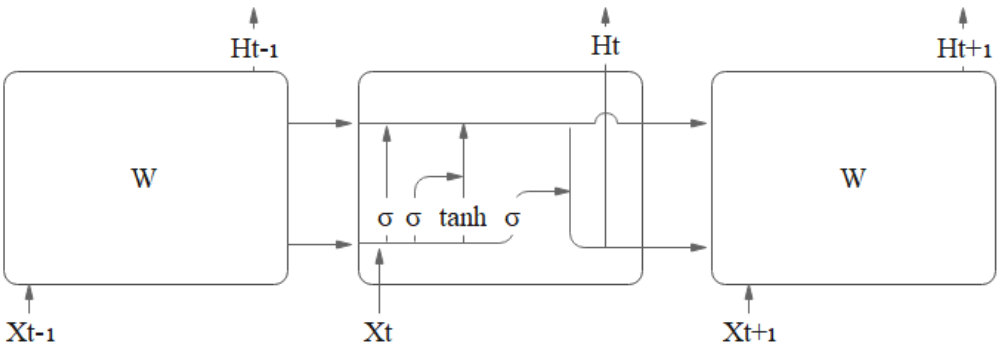

Figure 4.4 The architecture of LSTM

In this study, LSTM model is designed using three separate layers of Embedding

Layer, Subsequent LSTM Layer and Softmax Activation Function. Each LSTM unit whose number is the length of the input sentence is designed in such manner. The words of the sentences are the inputs of the embedding layer. After their corresponding word embeddings are determined using the corpus, then these embeddings become the input of the subsequent LSTM layer. The output of LSTM layer then feeds the softmax output layer and the LSTM cell for the next word in the sequence. The model design of an LSTM is given in Figure 4.5 below.

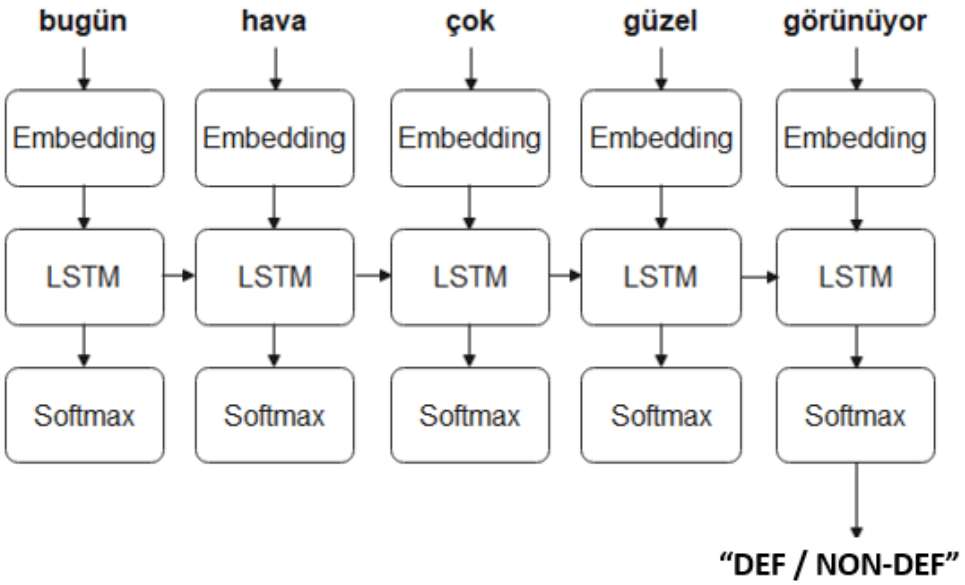

Figure 4.5 The model design of LSTM

4.4.1.1 LSTM Hyper-parameters

Specified hyper-parameters of our LSTM model with their corresponding values are depicted in Table 4.3 below.

Table 4.3 Hyper-parameters of the LSTM model with the adjusted values

Parameters	**Values**
Hidden layer	24,32,64,96,128,192,256
Learning rate	0.01,0.001
Epochs	50
Activation Function	Softmax
Optimizer	Adam
Loss function	MSE
Dropout	0.3,0.5

'Hidden Layer' is the number of LSTMs that are stack on top of each other in an LSTM layer, therefore the output of the LSTM can become the input of the next one and so on.

'Learning Rate' is the value of Adaptive Learning Rate of the model that is benefited for minimizing the loss function by controlling how much to change the model each time the model weights are updated.

'Epoch' is the number of iteration the model will perform, however 'Early Stopping' is adjusted on the model which means that the iteration will end when the loss value increases in comparison to the previous iteration. Since the 'Patience' hyper-parameter is set to 3, then the model will only tolerate the loss increase for three iterations at most.

'Activation Function' is basically decision mechanism that computes the weighted sum of the input and adds bias, thus determines whether the neuron should be activated or not. As short, it transforms an input of the neuron to produce output in the layer. We have chosen 'Softmax' activation function as the most suitable for this task since the produced output is a vector of values that sum to 1.0 such as the probabilities of class distribution. The formula of softmax activation function is shown in Equation 4.1.

$$\sigma(Z)_i = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} \tag{4.1}$$

where Z is the input vector of softmax function and all Z_i values are the elements of input. K is number of class of the model. The above of fraction ensures positive values on inputs and the below is the normalization that ensures all values are fixed in the range of 0 and 1.

'Optimizer's are the method for changing the attributes such as weights or learning rate of the model in order to provide lower losses. It is for sure that the lower the loss is; the better performance the model provides. Due to the capability of better working with learning rates, we applied 'Adam' optimizer on the LSTM model. As mentioned in Section 4.5.4, the loss function is adjusted as 'MSE'.

'Dropout' is the method that prevents overfitting by ignoring some random neurons of a certain number during training, therefore these neurons are not the part of the forward or backward pass. Thus, the network will not too dependent on the dataset for making satisfactory predictions.

4.4.2 Convolutional Neural Network (CNN – Conv1D)

CNN and its derivatives are one of the most benefited model among deep learning models in terms of visual tasks where input data is processed in the form of 2-dimension such as computer vision operations, semantic segmentation, object detection, image classification and etc. (Hussain et al., 2018; Liu et al., 2021; Teng et al., 2019). CNN model have shown notable efficiency for image diagnostics (Anavi et al., 2015; Banerjee et al., 2018; Cho et al., 2015; Hua et al., 2015; Shin et al., 2016), however it is possible that by processing the data in the form of 1-dimension, CNN is also applicable on word sequence, called 'Conv1D' (Banerjee et al., 2019). Conv1D approach has great performance and success in NLP tasks such as text classifications and sentiment analysis (Radhika et al., 2018; Yoon, 2014), text categorizations (Hughes et al., 2017; Mo et al., 2018) and many others.

For text classification, CNN tries to pick up pattern information on sequence data using filters with specific kernel sizes. In each convolutional layer (Conv1D), a patch

of input with the size of kernel is taken from the input vector sequence. Using this patch, the dot product of the multiplied vectors weights of filter is calculated. This operation is performed through the input sequence till the end of it in order to provide a pattern to train the model. The model architecture of our CNN model is depicted in Figure 4.6 below.

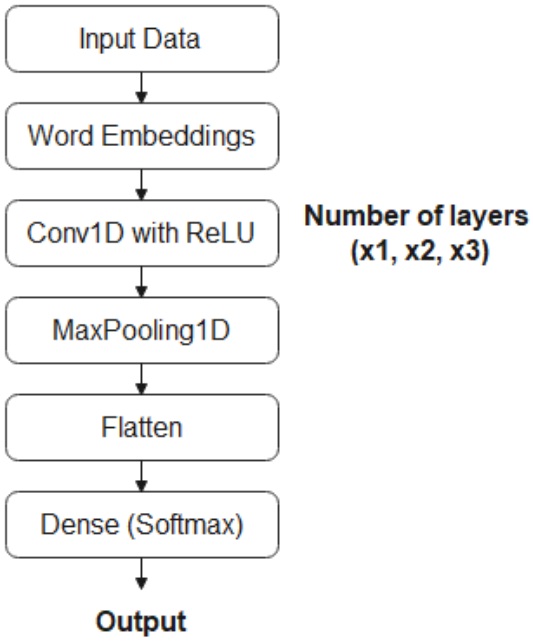

Figure 4.6 The model architecture of CNN

4.4.2.1 CNN Hyper-parameters

The hyper-parameters adjusted for CNN model with their corresponding values are listed in Table 4.4 below.

Table 4.4 Hyper-parameters of the CNN model with the adjusted values

Parameters	**Values**
Filters	(64,128,256)
Kernel size	(2,3)
Pool size	(2,3)
# of Conv1D layers	(1,2,3)
Epochs	50
Conv1D activation	Relu (Rectified Linear Unit)
Model activation	Softmax
Optimizer	Adam
Loss function	MSE
Dropout	0.5

'Filter' and 'Kernel size' are related hyper-parameters of a CNN model. In a sequence data such as voice or text, filter refers to a window which slides on the input sequence in order to pick up pattern information. The width of this sliding window is called kernel size. To give an example, when kernel size is set to 5 with 50 filters, then the convolutional layer creates 50 different sliding windows, each of them with length 5, therefore the result will bring 50 different convolutions.

'Pooling' is the operation of a dimensionally reduction of output in the convolutional layer. This operation is benefited due to the reducing the number of computation during training, therefore prevents overfitting. The pool size determines the output dimension of the pooling operation.

'ReLU' is the activation function of each convolutional layer that applies non-linearity on input data since convolutional process is linear, however real-time values such as texts or images contain non-linear components. ReLU sets all negative values to zero for preparing it to the next layer, as seen in Equation 4.2.

$$f(x) = max(0, x) \tag{4.2}$$

'Flatten' is used when the output is multidimensional, however it is desired to be linear to transfer to the Dense layer. This operation basically reshapes the input from the previous layer into 1-dimensional vector for the task of this study.

4.5 Machine Learning Classifiers

This thesis proposes three traditional machine learning classifier approaches for detecting defective expressions in Turkish sentences; KNN, SVM and RF. Even though their performances fall behind in general in comparison to deep learning approaches, some of them provides acceptable results. These three approaches are explained in the following sections in details.

4.5.1 K-Nearest Neighbor (KNN)

KNN is known to be one of the easiest approach, which is an instance-based or non-parametric classifier, in data mining and machine learning (Qin et al., 2013; Zhang et al., 2017a,b). As likely with other traditional classifiers, KNN tries to find the most similar instances of the same class which are supposed to have high probability. KNN first finds the k number of nearest neighbors in the training set with distance calculation of all train instances for each test data in the process (Zhu et al., 2014). Then it predicts the test data with the major class among k nearest training data (Deng et al., 2016). KNN has been chosen as one of the most effective algorithm in the field of data mining in comparison to the other supervised learning approaches such as Centroid-Based Classifier (CB), SVM, Decision Trees, Naive Bayes, Winnow, Voting and etc. (Wu et al., 2008).

KNN approach is considered to be beneficial for the purpose of this study as the main goal is to classify Turkish sentences whether a sentence has defective expression or not. KNN algorithm requires only the adjustment of 'k' parameter and sentence vectors which are calculated by averaging word vectors of each word in the sentences.

4.5.2 Support Vector Machine (SVM)

SVM is a kind of supervised learning approach for the tasks of classification, regression and outlier detection, introduced by Vapnik (1999). SVMs are derived from a robust theory of structural risk minimization, which aims at minimizing the structural risk, instead of the training error (Vapnik, 1998; Suykens & Vandewalle, 1999). This classifier is widely used in text classification and clustering tasks with acceptable efficiency results (Fayed & Atiya, 2021; Garla et al., 2013; Sun et al., 2009).

Support vector machines have several advantages; they are excessively efficient in high dimensional spaces and tasks where there is a great number of dimensions in comparison to the number of samples. Moreover, SVM provides support vectors which are basically the subset of training points in the decision function, therefore this classifier works memory efficient during classification.

4.5.3 Random Forest (RF)

Random Forest is widely used due to being an ensemble learning and an advanced decision tree method, introduced by Breiman (2001) that is excessively benefited for classification and regression operations (Aria et al., 2021). Ensemble learning means that it combines more than one decision algorithm for classifying objects.

Since the high variance makes a decision tree model unstable, random forest is highly preferred due to creating several decision trees which are extracted from random subsets of training set (Liang & Zhao, 2019). Then, it combines the results from those decision trees for determining the classes of test set, therefore it provides more accurate results by averaging the scores of each tree.

4.5.4 Evaluation Metrics

In this research, the performances of learning models have been measured using the accuracy (validation accuracy) metric to determine the success and loss (mean squared error [MSE]) metric to analyze whether there exists overfitting or not, defined as Equations 4.3 and 4.4.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{4.3}$$

The calculation of MSE is held by averaging result of the square value of subtracting the original values of the data from the predicted ones, defined as follows:

$$MSE = \frac{1}{N}\sum_{i=0}^{n}(actual\ values\ -\ predicted\ values)^2 \tag{4.4}$$

where N is the number of sentences in dataset. The symbol of sigma calculates the difference between original and estimated values which are taken on each i value in the range from 1 to n.

In addition to accuracy metric; precision, recall and f-score metrics have been applied for a better analyse in the models. The equations of these metrics are given in Equations 4.5 to 4.7 respectively. The abbreviations of TP, FP, TN and FN stand for True-Positive, False-Positive, True-Negative and False-Negative respectively.

$$Precision = \frac{TP}{TP\ +\ FP} \tag{4.5}$$

$$Recall = \frac{TP}{TP\ +\ FN} \tag{4.6}$$

$$F-score = \frac{2\ x\ precision\ x\ recall}{precision\ +\ recall} \tag{4.7}$$

4.6 Programming Tools and Libraries

Python is an open-source programming language which is highly preferred for machine learning studies due to providing great convenience during implementation and variety of libraries such as gensim, keras, tensorflow, numpy and etc. Therefore, python programming language is preferred for implementing the study of this thesis by using the IDE of PyCharm.

For the purpose of this study, several libraries of python have been benefited for exact operations such as ‘gensim’ for word2vec implementation, ‘tensorflow’ and ‘keras’ for learning models and evaluation metrics, ‘numpy’ and ‘scipy’ for scientific calculations, ‘NLTK (Natural Language Toolkit)’ for several NLP operations including Turkish stop-word list and etc.

CHAPTER FIVE
MODEL DEVELOPMENT AND EXPERIMENTAL RESULTS

5.1 Model Implementation

The models that are proposed for the purpose of this study basically performs a classification task for Turkish sentences to determine whether they have defective expression or not. Two separate approaches, which are deep neural techniques and traditional machine learning classifiers, have been analyzed in brief and as the result, the models to be used have been selected for each approach. In conclusion, LSTM and CNN with one-dimension (Conv1D) have been found the most beneficial as deep neural approaches in addition to KNN, SVM and RF as traditional machine learning classifiers for the purpose of this study, shown in Figure 5.1 below.

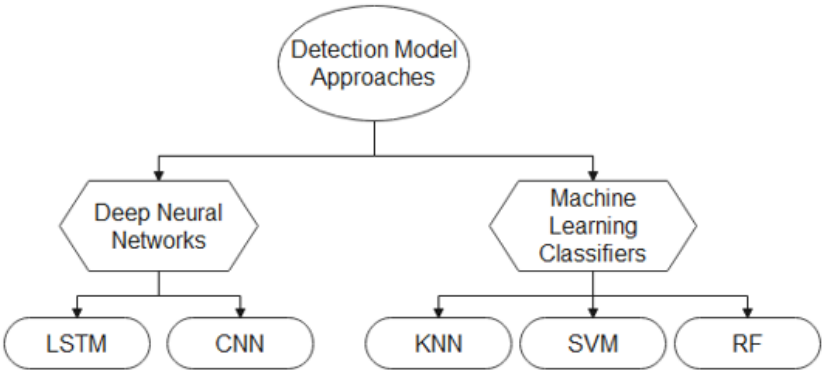

Figure 5.1 Detection model approaches

All the models have been implemented using Python programming language due to the capability of providing several libraries for the complicated tasks of machine learning, deep learning, NLP and etc. Thus, the code implementations become less suffering which leads to the researcher to be able to spend more time on the model configuration and optimization.

The detection models for each proposed approach is almost in the same flow; as for the step, the input sentences are taken from the dataset, then they are preprocessed

using NLP techniques, after that feature extraction is performed using word embeddings. The next step is to train the model with data to generate the detection model, therefore the input vectors split into two separate set of train data to feed the learning model and test data for validation, shown in Table 5.1 below.

Table 5.1 Train-test data split ratio

	Train	**Test**	**Total**
# of sentences	22,317	7,439	29,756
Percentage	75%	25%	100%

In order to provide better performances for traditional machine learning techniques, sentence vectors which are the average word vectors of each sentence have been calculated to train them while deep learning techniques do not require such operation due to the capability of using embedding matrix of words for training. After the split operation, the model is fed using training set. Finally, test set is used to validate the model whether it is successful in terms of detection or not. The entire flow can be partitioned into two separate phases of NLP phase and model development phase, depicted in Figure 5.2 below.

When the results are not satisfactory for the purpose, then the model is tried to be optimized using several operations such as adjusting different values on model hyper-parameters or altering them with slight increases or decreases. Sometimes, adding or removing layers during deep neural model implementation may result in better accuracies.

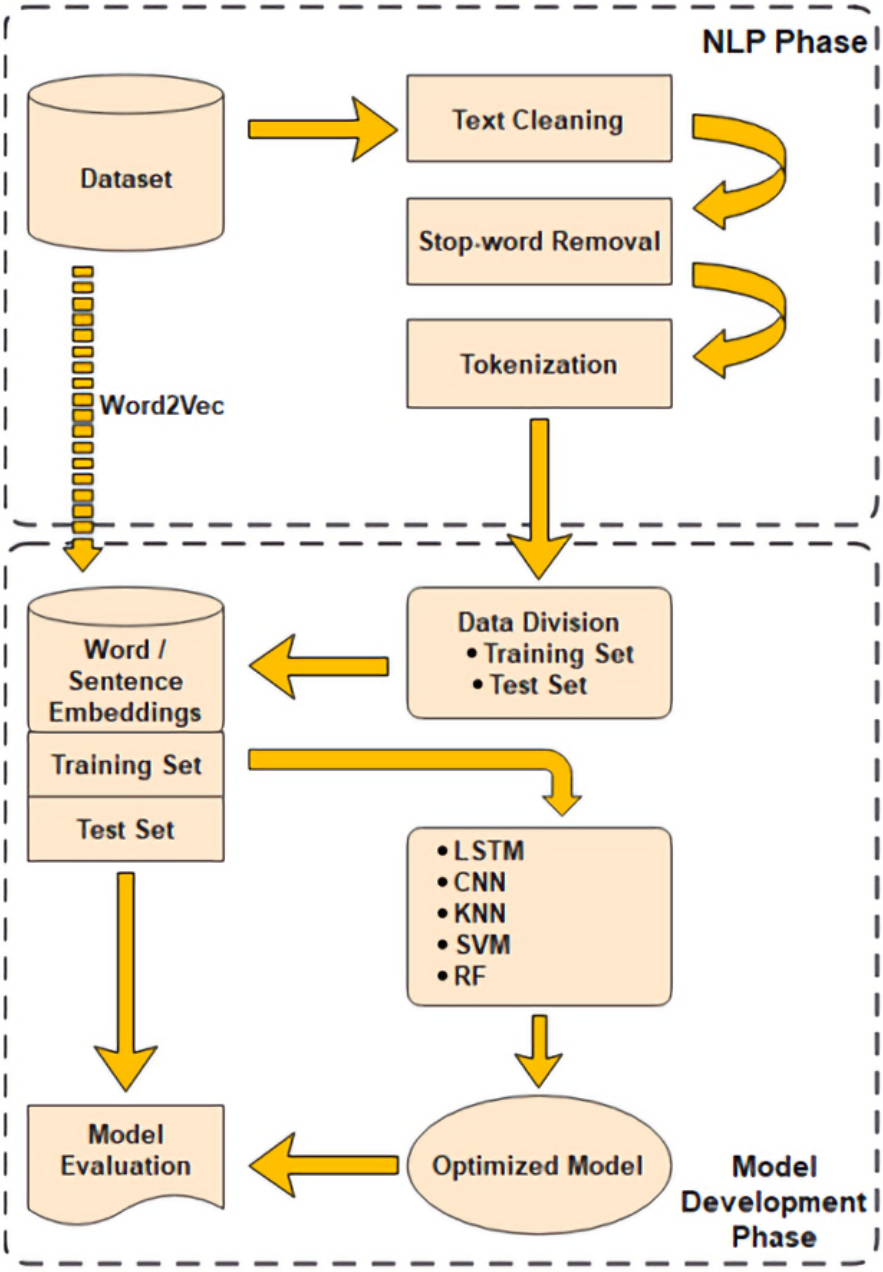

Figure 5.2 The general flow of detection algorithm

5.2 Model Optimization and Experimental Results

Machine learning or deep learning implementation requires a great effort and an excessive research in order to provide high performance for the task. In contrast to

the easiness of code implementation of the models thanks to Python and its beneficial libraries, generating the most optimum model is one of the most challenging issue since there is a complex structure with notable mathematics behind the scene, therefore analyzing and configuring a model for a specific task to optimize it requires plenty of experience and great knowledge.

Model optimization refers to adjusting the best possible hyper-parameters of each model in order to provide higher performance and accuracy for the task. Configuring these hyper-parameters has a significantly positive effect on the model for a better learning capability, therefore there has been an excessive number of empirical trials performed for each model for the sake of adjusting the best possible hyper-parameters.

5.2.1 LSTM

The empirical trials of LSTM model by several adjustments of the essential hyper-parameters have been performed to generate the optimized model. The results are shown in Table 5.2 below.

Table 5.2 The experimental results of LSTM model

Experiment (E)	**Hidden Layers**	**Dropout**	**Learning Rate**	**Validation Accuracy**	**Validation Loss**
E1	24	0.3	0.001	0.7933	0.1456
E2	24	0.3	0.01	0.7830	0.1549
E3	24	0.5	0.001	0.7778	0.1535
E4	24	0.5	0.01	0.7705	0.1615
E5	32	0.3	0.001	0.8074	0.1383
E6	32	0.3	0.01	0.8091	0.1433
E7	32	0.5	0.001	0.7982	0.1448
E8	32	0.5	0.01	0.7838	0.1496
E9	64	0.3	0.001	0.8387	0.1233
E10	64	0.3	0.01	0.8236	0.1355

Table 5.2 continues

E11	64	0.5	0.001	0.8338	0.1245
E12	64	0.5	0.01	0.8305	0.1279
E13	96	0.3	0.001	0.8529	0.1155
E14	96	0.3	0.01	0.8379	0.1252
E15	96	0.5	0.001	0.8437	0.1226
E16	96	0.5	0.01	0.8168	0.1392
E17	128	0.3	0.001	0.8625	0.1101
E18	128	0.3	0.01	0.8248	0.1338
E19	128	0.5	0.001	0.8593	0.1138
E20	128	0.5	0.01	0.8133	0.1412
E21	192	0.3	0.001	0.8605	0.1100
E22	192	0.3	0.01	0.8320	0.1331
E23	192	0.5	0.001	0.8666	0.1055
E24	192	0.5	0.01	0.8126	0.1376
E25	**256**	**0.3**	**0.001**	**0.8794**	**0.0992**
E26	256	0.3	0.01	0.8320	0.1298
E27	256	0.5	0.001	0.8707	0.1026
E28	256	0.5	0.01	0.8102	0.1436

As seen in Table 5.2, the trial of E25 is found out to be the most optimized model of LSTM. In order to assure the performance, 10-fold cross validation has been applied to the most optimized model. The results are depicted in Table 5.3 below.

Table 5.3 The performance results of the most optimized model of LSTM using 10-fold cross validation

k-fold	**Validation Loss**	**Validation Accuracy**
1	0.1050	0.8609
2	0.1096	0.8545
3	0.1094	0.8548
4	0.1120	0.8501
5	0.1095	0.8528
6	0.1141	0.8488
7	0.1109	0.8538
8	0.1160	0.8427
9	0.1068	0.8534
10	0.1104	0.8555
Average	**0.1103**	**0.8527**

5.2.2 CNN (Conv1D)

After the empirical trials performed on CNN model by several tunings of the required hyper-parameters, the model is performed with the resulting values, shown in Table 5.4 below.

Table 5.4 The experimental results of CNN models, split by the number of layers

Experiment (E)	**Conv1D Filters**	**Kernel Size**	**Pooling Size**	**Validation Accuracy**	**Validation Loss**
E1	64	3	2	0.8248	0.1283
E2	64	3	3	0.8167	0.1327
E3	**128**	**3**	**2**	**0.8432**	**0.1207**
E4	128	3	3	0.8345	0.1213
E5	64, 64	3, 2	2, 2	0.8198	0.1294

Table 5.4 continues

E6	64, 64	3, 3	2, 2	0.8018	0.1407
E7	64, 64	3, 3	2, 3	0.7918	0.1451
E8	64, 64	3, 3	3, 2	0.8177	0.1339
E9	64, 128	3, 2	2, 2	0.8100	0.1346
E10	64, 128	3, 3	3, 2	0.8276	0.1245
E11	**128, 128**	**3, 2**	**2, 2**	**0.8374**	**0.1228**
E12	128, 128	3, 3	3, 2	0.8247	0.1251
E13	64, 64, 64	3, 3, 2	3, 2, 2	0.8054	0.1365
E14	64, 64, 64	2, 3, 2	3, 2, 2	0.7912	0.1438
E15	64, 64, 128	3, 3, 2	3, 2, 2	0.8142	0.1323
E16	64, 128, 128	3, 3, 2	3, 2, 2	0.8244	0.1237
E17	**128, 128, 128**	**3, 3, 2**	**3, 2, 2**	**0.8433**	**0.1217**
E18	64, 128, 256	3, 3, 2	3, 2, 2	0.8125	0.1322
E19	256, 256, 256	3, 3, 2	3, 2, 2	0.8325	0.1165

As seen in Table 5.4, the trial of E17 is found out to be the most optimized model of CNN. In order to assure the performance, 10-fold cross validation has been applied to the most optimized model. The results are depicted in Table 5.5 below.

Table 5.5 The performance results of the most optimized model of CNN using 10-fold cross validation

k-fold	**Validation Loss**	**Validation Accuracy**
1	0.1211	0.8351
2	0.1113	0.8384
3	0.1147	0.8259
4	0.1152	0.8397
5	0.1173	0.8385
6	0.1216	0.8290
7	0.1244	0.8215
8	0.1140	0.8346
9	0.1194	0.8343
10	0.1174	0.8249
Average	**0.1176**	**0.8321**

5.2.3 KNN

The classifier of KNN has been attempted to be optimized through the number k determination between range of 3 and 50. After the emprical trials, the best number of k has been found out to be 3, depicted in Figure 5.3. The ROC Curve of the model with the AUC score of 0.75 is shown in Figure 5.4 below.

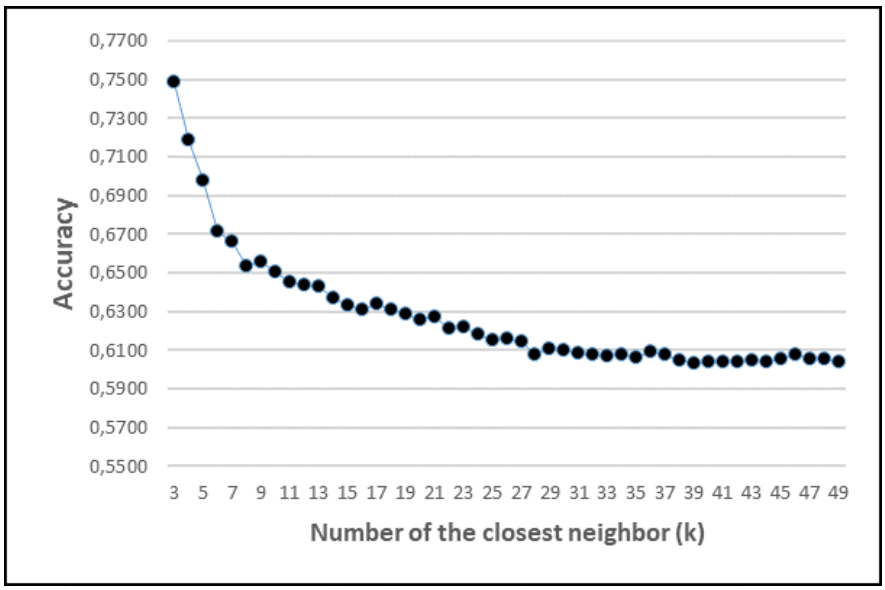

Figure 5.3 The experimental results of KNN in accordance with the number k

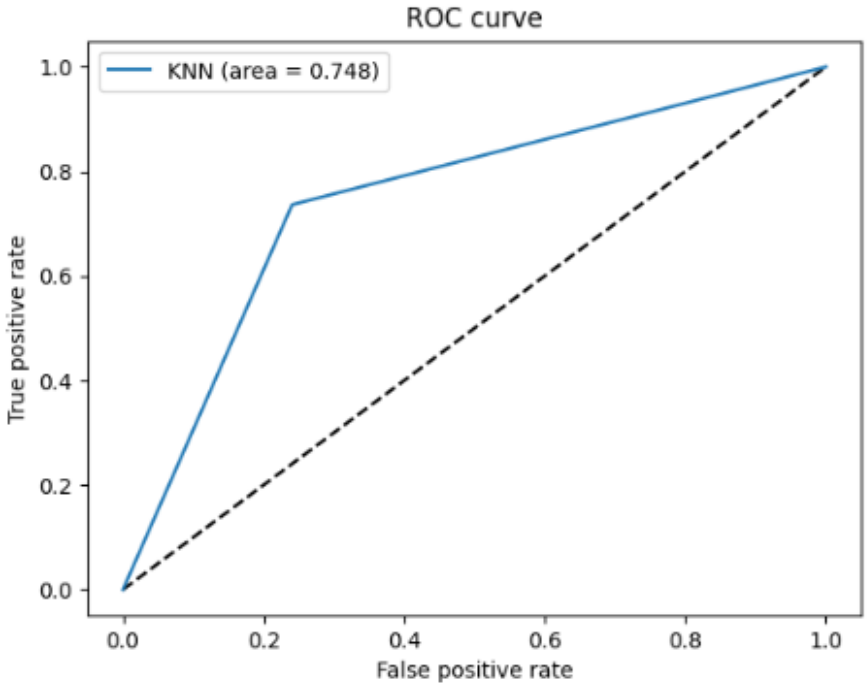

Figure 5.4 ROC Curve of KNN model

As seen in Figure 5.3, the best number of k is found out as 3. In order to assure the performance, 10-fold cross validation has been applied to the model with that number of k. The results are depicted in Table 5.6 below.

Table 5.6 The performance results of KNN model using 10-fold cross validation

Precision	**Recall**	**F-Score**	**Accuracy**
0.78	0.76	0.77	0.67

5.2.4 SVM and RF

Having practiced several experiments using SVM as linear classifier and RF as ensemble learning classifier, the best performances of these models are depicted in Table 5.7 below.

Table 5.7 Model performances of SVM and RF classifiers

Classifier	**Precision**	**Recall**	**F-Score**	**Accuracy**
RF	0.79	0.85	0.81	0.78
SVM	0.60	0.75	0.66	0.58

The ROC Curves of RF and SVM models with the AUC scores of 0.78 and 0.75 respectively are shown in Figure 5.5 and Figure 5.6.

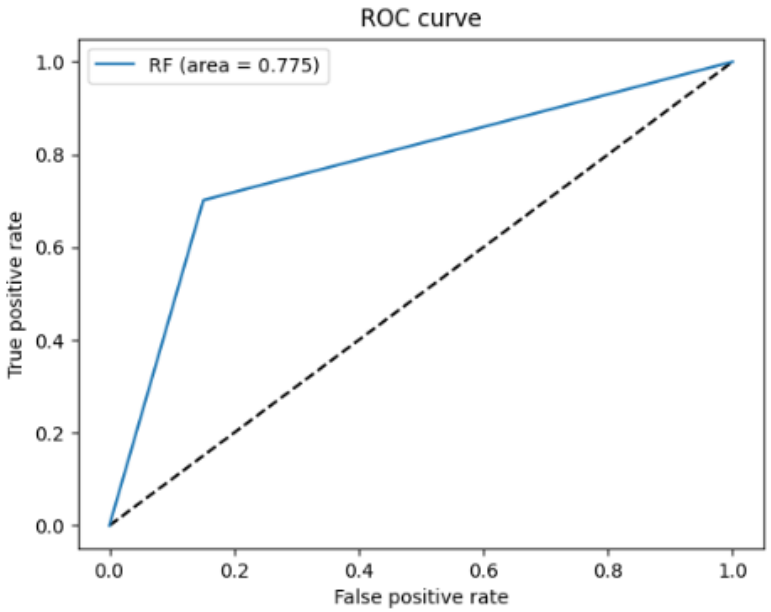

Figure 5.5 ROC Curve of RF model

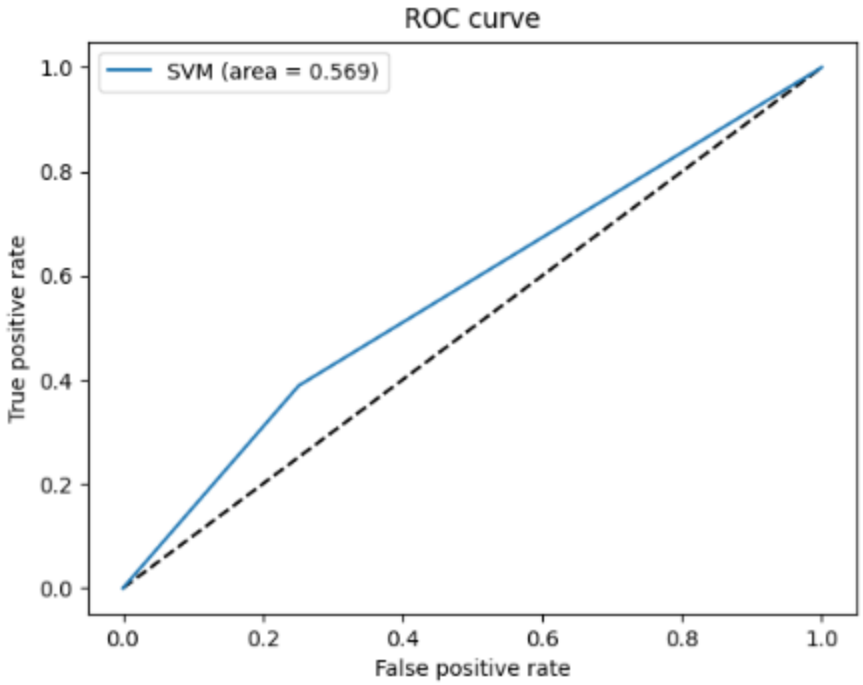

Figure 5.6 ROC Curve of SVM model

In order to assure the performance, 10-fold cross validation has been applied to the RF and SVM models. The results are depicted in Table 5.8 below.

Table 5.8 The performance results of RF and SVM models using 10-fold cross validation

Classifier	**Precision**	**Recall**	**F-Score**	**Accuracy**
RF	0.76	0.84	0.80	0.77
SVM	0.60	0.77	0.67	0.57

5.3 Discussion

This section analyzes the performance results of the proposed models for the task of detecting defective expressions in Turkish sentences. Each model has sui generis architecture, therefore learning processes for the task of this study; thus resulting performances of each model ended up with various success rates even though they have been trained with the same dataset. Furthermore, the fact that the absence of

such study with this subject in the literature is one of the most important constraints, this situation led us focus on benchmarking the performances of each model.

As mentioned in the previous chapters, the absence of a publicly available dataset was one of the major constraints for this study. After it is found out that the number of collected data was extremely insufficient for training a learning model, we performed a data augmentation operation. Before augmentation, the dataset has been applied on the learning models for training, which all resulted in pretty low performances, as depicted in Table 5.9.

Table 5.9 Comparison of the learning models trained before data augmentation

Classifier	**Precision**	**Recall**	**F-Score**	**Accuracy**
LSTM	0.51	0.57	0.58	0.57
CNN	0.50	0.56	0.56	0.56
KNN	0.56	0.61	0.67	0.68
SVM	0.56	0.60	0.73	0.66
RF	0.47	0.53	0.66	0.58

Augmenting the data using synonym words made a significant boost in this study in terms of model performances. In order to make a comprehensive analyse on the behaviors of the models in terms of handling input data, we performed two separate approaches. First, dataset has been split into train and test data without shuffling, that is the first 75% of the dataset is taken as train set and the rest of them is taken as test set. The results are depicted in Table 5.10.

Table 5.10 Comparison of the learning models trained after data augmentation without shuffling

Classifier	**Precision**	**Recall**	**F-Score**	**Accuracy**
LSTM	0.86	0.88	0.87	0.87
CNN	0.72	0.82	0.77	0.76
RF	0.64	0.84	0.73	0.71
KNN	0.58	0.56	0.57	0.60
SVM	0.48	0.67	0.56	0.53

As for the second approach, dataset has been split by applying shuffling, that is both the 75% part of the data for train set and the rest part as test set are taken from the dataset randomly. The results are shown in Table 5.11.

Table 5.11 Comparison of the learning models trained after data augmentation by shuffling

Classifier	**Precision**	**Recall**	**F-Score**	**Accuracy**
LSTM	0.88	0.89	0.88	0.87
CNN	0.80	0.88	0.84	0.84
RF	0.79	0.85	0.81	0.78
KNN	0.76	0.81	0.78	0.74
SVM	0.60	0.75	0.66	0.58

The analyses of data handling showed that shuffling operation during train-test split provides relatively better performances for all the models due to the fact that non-shuffled splits might ended up being biased.

Having analyzed several LSTM classification tasks in the literature, it can be clearly extracted that increase in the number of hidden layer tends to provide higher accuracy rates. In the experimental trials of LSTM, the number of hidden layer have been adjusted from 24 to 256 and the results evidently validates this fact, as seen in Figure 5.7 below.

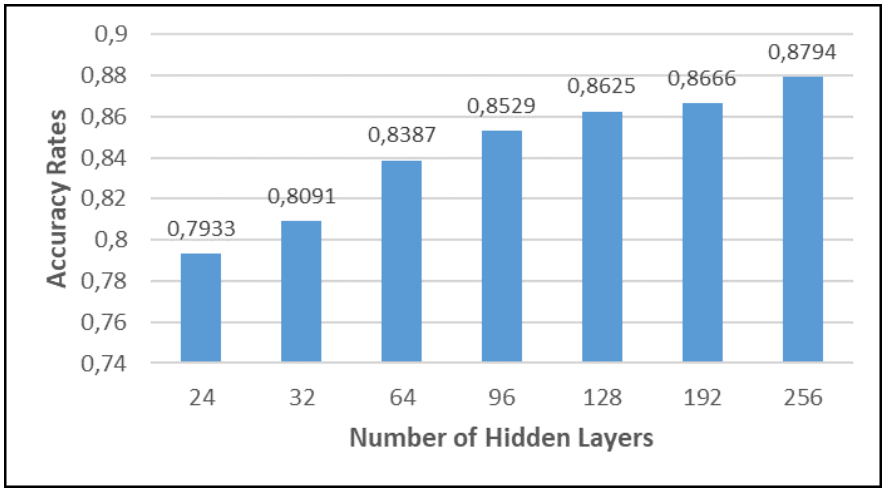

Figure 5.7 Best accuracy rates of LSTM model according to the number of hidden layers

Furthermore, tuning lower learning rates has positive effect on model success as well. In addition, applying a low dropout provides better results in accordance with higher ones so that less information is dropped in every iteration through learning process. However, reducing it more than enough may result in overfitting which must be considerably avoided in machine learning tasks. In the model structure, Early Stopping with 'patience' value of 3 has also been implemented in order to avoid overfitting by stopping the training operation by the time the loss increases 3 times consecutively. Figure 5.8 depicts the model accuracy and loss values per epoch without Early Stopping.

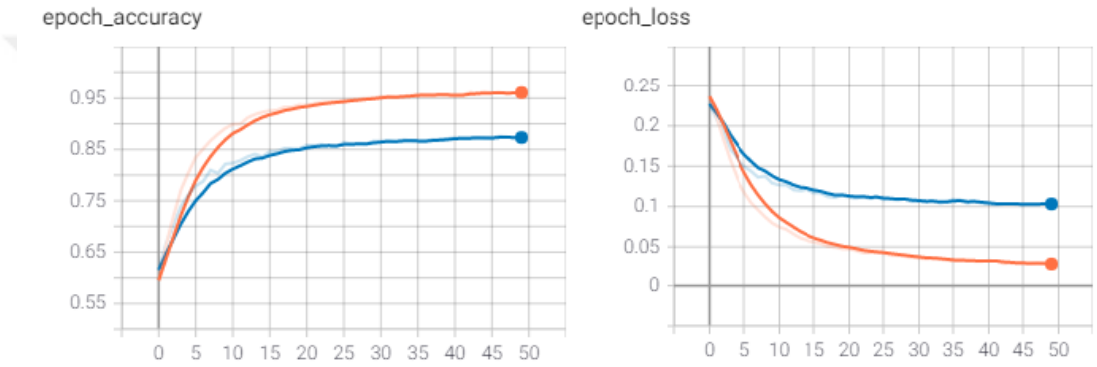

Figure 5.8 Epoch-accuracy (left) and epoch-loss (right) diagrams of LSTM's best performance without Early Stopping (Orange lines represent training and blue ones represent test operations)

To conclude, the best performance with LSTM has been implemented by adjusting the number of hidden layers as 256, the learning rate as 0.001 and the value of dropout as 0.3 as shown in Figure 5.9. This figure also verifies that there is no overfitting in this model since the loss has no increasing tendency through each epoch.

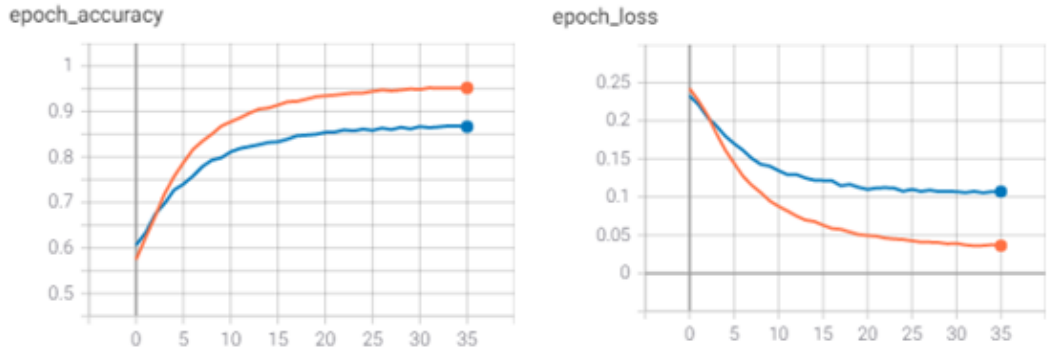

Figure 5.9 Epoch-accuracy (left) and epoch-loss (right) diagrams of LSTM's best performance using Early Stopping (Orange lines represent training and blue ones represent test operations)

In the analysis of CNN approach, it has been found out that the filter value of the layer has a significant boost on the performance regardless of number of the convolutional layer in the model. Using small size of kernel in addition to the pooling size performed pretty well, however they did not affect the results as much as filters did since filters are the main processes which extracts patterns from the text sequence. However, when comparing to LSTM, the importance of long-term dependencies between word sequences comes into prominence in favor of LSTM which benefits them for feature extraction. Table 5.12 depicts the best results of each CNN model in accordance with the number of layers.

Table 5.12 Best accuracy rates of CNN models regarding the number of layers

# of Hidden Layers	**Conv1D Filters**	**Kernel Size**	**Pooling Size**	**Validation Accuracy**	**Validation Loss**
1	128	3	2	0.8432	0.1207
2	128, 128	3, 2	2, 2	0.8374	0.1228
3	128, 128, 128	3, 3, 2	3, 2, 2	0.8433	0.1217

As seen in Table 5.12, despite the slight difference, three-layer convolutional neural network provided the best accuracy among other models. Figure 5.10 describes the best model in terms of relationships between Epoch-Accuracy and Epoch-Loss. This figure also verifies that there is no overfitting throughout the training.

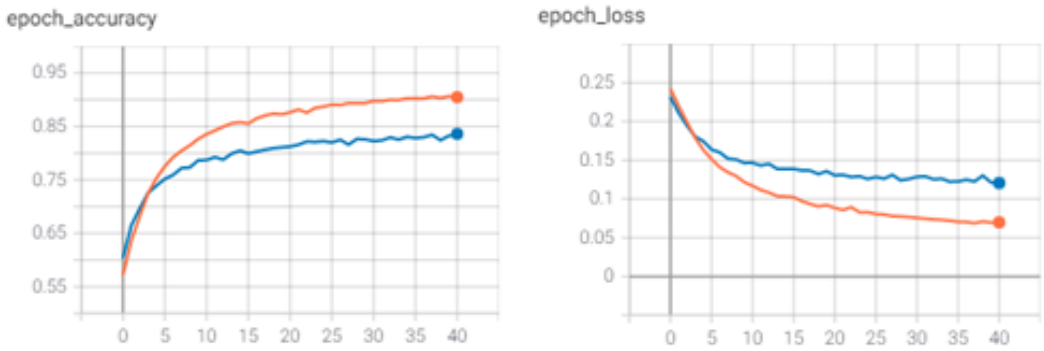

Figure 5.10 Epoch-accuracy (left) and epoch-loss (right) diagrams of CNN's best performance (Orange lines represent training and blue ones represent test operations)

In the analysis of machine learning classifiers, RF and KNN classifiers come into prominence in accordance with SVM classifier in terms of model performance. Even though there is slight difference between RF and KNN, the reason why RF performed better is because it creates numerous variations of decision trees during the classification and each final scores are averaged to get higher accuracy. However, since the performance of KNN excessively depends on determining the best number of k which specifies the calculation of closest neighbor distances, it would not provide such an optimum performance as RF did. On the other hand, since being a linear classifier, SVM provides more successful performances in regression and outlier analyze problems in general when comparing to text classification.

In the light of all experimental trials, one of the first outcomes of this thesis is that deep learning approaches provide excessively better performances for the task of detecting defective expressions in Turkish sentences in accordance with the traditional machine learning classifiers. Among the all models, LSTM performed with best accuracy performance with the accuracy rate of 87.94%. The highest accuracy rates of CNN, KNN, SVM and RF are 84.33%, 74.89%, 58.5% and 78.12% respectively, shown in Figure 5.11. The reason why deep learning approaches distinguish for this task in comparison to machine learning classifiers is their learning capabilities using previous experiences and applying them on future predictions.

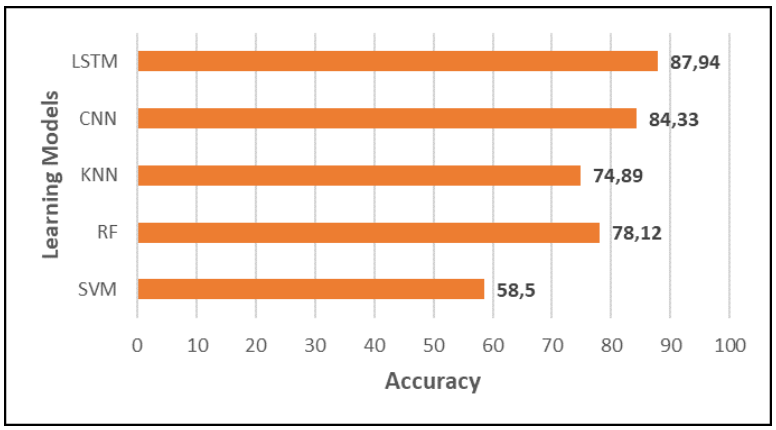

Figure 5.11 The model performances of each model in terms of accuracy rates

CHAPTER SIX
CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Natural language is the main tool for understanding and addressing our opinions and feelings to other people. Since natural language is ambiguous, defective expressions become one of the most important issues that should be considered in order to avoid misleading and misunderstanding while speaking or writing. In this thesis, deep learning approaches and machine learning classifiers are proposed to detect defective expressions in Turkish sentences. The variety of semantic and morphological types of them and the fact that there was almost no previous research related to defective expressions make this study more grueling. Even though several studies have been performed by the researchers of education scientists, they are heavily related to measuring the capability of students or mass media about how successful they are in terms of determining or minding defective expressions in sentences. However, none of these analyzes have been performed using machine learning techniques.

One of the main constraints was the absence of domain specific dataset for the purpose of this study to train the learning models, therefore data collection operation was held from online sources of education organizations such as courses, schools or exam centers and an amount of input data has been collected sentence by sentence by labeling them manually whether they have defective expression or not. However, the amount of collected sentences was still inadequate to train a learning model, for this reason data augmentation operation was performed in order to increase the number of input data up to an acceptable amount. After that, a corpus of word embeddings was created to be used for feature extraction using the dataset with word2vec technique.

The experimental trials were conducted on the dataset using both deep learning approaches of LSTM and CNN in addition to machine learning classifiers of KNN, SVM and RF. In the light of the results, it is clearly interpreted that deep learning

approaches provided by far more accurate performances in comparison to the classifiers. Due to the learning capabilities of long term dependencies, LSTM come into prominence when comparing to CNN. This can also be pointed out that because Turkish has excessively complex structure in semantic context and morphological grammar, traditional classifiers may not be the best option for some NLP operations belonging to some specific languages such as Turkish.

To conclude, applying deep learning models to detect defective expressions in Turkish sentences provided acceptable results. By being an original study in this field, this study is a great contribution to Turkish NLP and an excessive source for other researchers studying this area.

6.2 Recommendations

This study has some limitations that must be addressed for future improvements. One of the most crucial issue is the qualification of dataset to train the models. Since the dataset was collected manually from online sources, the quality of each sentence and their compatibility with each other must be taken into consideration. Therefore, one of the future directions could be improving the quality of input data using some language-specific libraries and other NLP preprocessing techniques for providing a better domain-specific dataset on the models. Moreover, as for another future direction, augmenting the input data using different approaches may provide a comprehensive dataset for future studies. It is also estimated that adjusting the values of hyper-parameters in the models within a wider range may result in potential better accuracies. Finally, implementing hybrid or sequential learning models may also provide more accurate performances.

REFERENCES

Aktas, Ö., Birant, Ç. C., Aksu, B., & Çebi, Y. (2013). Automated synonym dictionary generation tool for Turkish (ASDICT). *Bilig*, *65*, 47.

Anavi, Y., Kogan, I., Gelbart, E., Geva, O., & Greenspan, H. (2015). A comparative study for chest radiograph image retrieval using binary texture and deep learning classification. In *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, *IEEE*, 2940–2943.

Aria, M., Cuccurullo, C., & Gnasso, A. (2021). A comparison among interpretative proposals for random forests. *Machine Learning with Applications*, *6*, 100094.

Bahar, M. (2006). *Teorik Gramer Bilgisi ile Yazılı Anlatım Bozukluğu Arasıdaki İlişki (İlköğretim II. Kademe Uşak Örneği)*. Master's thesis, Afyon Kocatepe University, Afyon.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Banerjee, I., Crawley, A., Bhethanabotla, M., Daldrup-Link, H. E., & Rubin, D. L. (2018). Transfer learning on fused multiparametric MR images for classifying histopathological subtypes of rhabdomyosarcoma. *Computerized Medical Imaging and Graphics*, *65*, 167–175.

Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., Chapman, B., Amrhein, T., Mong, D., Rubin, D. L., et al. (2019). Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artificial intelligence in medicine*, *97*, 79–88.

Bano, M. (2015). Addressing the challenges of requirements ambiguity: A review of empirical literature. In *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, *IEEE*, 21–24.

Berzak, Y., Barbu, A., Harari, D., Katz, B., & Ullman, S. (2016). Do you see what I mean? Visual resolution of linguistic ambiguities. *arXiv preprint arXiv:1603.08079*.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Büyükikiz, K. K. (2007). *İlköğretim 8.sınıf öğrencilerinin yazılı anlatım becerilerinin söz dizimi ve anlatım bozukluğu açısından değerlendirilmesi*. Master's thesis, Gazi University, Ankara.

Çetinkaya, G., & Ülper, H. (2015). Anlatim bozuklugu tasiyan tümcelerin kabul edilebilirligi ve kavranilabilirligi: Ögrenci okurlar üzerinden karsilastirmali bir inceleme. *Hasan Ali Yücel Egitim Fakültesi Dergisi*, *12*(1), 341.

Cho, J., Lee, K., Shin, E., Choy, G., & Do, S. (2015). How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv preprint arXiv:1511.06348*.

Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient knn classification algorithm for big data. *Neurocomputing*, *195*, 143–148.

Dogra, V., et al. (2021). Banking news-events representation and classification with a novel hybrid model using distilbert and rule-based features. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(10), 3039–3054.

Elkahky, A., Webster, K., Andor, D., & Pitler, E. (2018). A challenge set and methods for noun-verb ambiguity. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2562–2572.

Fang, G., Zeng, F., Li, X., & Yao, L. (2021). Word2vec based deep learning network for dna n4-methylcytosine sites identification. *Procedia Computer Science*, *187*, 270–277.

Fayed, H. A., & Atiya, A. F. (2021). Decision boundary clustering for efficient local SVM. *Applied Soft Computing*, *110*, 107628.

Ferrari, A., & Esuli, A. (2019). An NLP approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*, *26*(3), 559–598.

Garla, V., Taylor, C., & Brandt, C. (2013). Semi-supervised clinical text classification with laplacian SVMs: an application to cancer case management. *Journal of biomedical informatics*, *46*(5), 869–875.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, *12*(10), 2451–2471.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, *PMLR*, 1764–1772.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, *31*(5), 855–868.

Hoceini, Y., Cheragui, M. A., & Abbas, M. (2011). Towards a new approach for disambiguation in nlp by multiple criterian decision-aid. *Prague Bull. Math. Linguistics*, *95*, 19–32.

Hua, K.-L., Hsu, C.-H., Hidayati, S. C., Cheng, W.-H., & Chen, Y.-J. (2015). Computer-aided classification of lung nodules on computed tomography images via deep learning technique. *OncoTargets and therapy*, *8*.

Hughes, M., Li, I., Kotoulas, S., & Suzumura, T. (2017). Medical text classification using convolutional neural networks. In *Informatics for Health: Connected Citizen-Led Wellness and Population Health*, 246–250.

Hussain, M., Bird, J. J., & Faria, D. R. (2018). A study on cnn transfer learning for image classification. In *UK Workshop on computational Intelligence*, *Springer*, 191–202.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 1–11.

Kuchta, J., & Padhiyar, P. (2018). Extracting concepts from the software requirements specification using natural language processing. In *2018 11th International Conference on Human System Interaction (HSI)*, *IEEE*, 443–448.

Kumhar, S. H., Kirmani, M. M., Sheetlani, J., & Hassan, M. (2021). Word embedding generation for Urdu language using word2vec model. *Materials Today: Proceedings*.

Lauriola, I., Lavelli, A., & Aiolli, F. (2021). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Liang, Y., & Zhao, P. (2019). A machine learning analysis based on big data for eagle ford shale formation. In *SPE Annual Technical Conference and Exhibition*, *OnePetro*.

Liu, Y., Pu, H., & Sun, D.-W. (2021). Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. *Trends in Food Science & Technology*.

Mahadzir, N. H., Omar, M. F., & Nawi, M. N. M. (2018). Semantic similarity measures for Malay-English ambiguous words. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *10*(1-11), 109–112.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Mo, K., Park, J., Jang, M., & Kang, P. (2018). Text classification based on convolutional neural network with word and character level. *Journal of the Korean Institute of Industrial Engineers*, *44*(3).

Muhammad, P. F., Kusumaningrum, R., & Wibowo, A. (2021). Sentiment analysis using word2vec and long short-term memory (LSTM) for Indonesian hotel reviews. *Procedia Computer Science*, *179*, 728–735.

Nourani, V., & Behfar, N. (2021). Multi-station runoff-sediment modeling using seasonal LSTM models. *Journal of Hydrology*, *601*, 126672.

Özçift, A., Akarsu, K., Yumuk, F., & Söylemez, C. (2021). Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish. *Automatika*, *62*(2), 226–238.

Özdem, A. (2012). *Çanakkale'deki yerel gazetelerin anlatım bozuklukları açısından incelenmesi*. Master's thesis, Çanakkale Onsekiz Mart University, Çanakkale.

Qin, Z., Wang, A. T., Zhang, C., & Zhang, S. (2013). Cost-sensitive classification with k-nearest neighbors. In *International Conference on Knowledge Science, Engineering and Management*, *Springer*, 112–131.

Radhika, K., Bindu, K., & Parameswaran, L. (2018). A text classification model using convolution neural network and recurrent neural network. *International Journal of Pure and Applied Mathematics*, *119*, 1549–1554.

Sak, H., Güngör, T., & Saraçlar, M. (2011). Resources for Turkish morphological processing. *Language resources and evaluation*, *45*(2), 249–261.

Saydam, E. (2016). *Giresun ili yerel (yazılı) basınında yapılan yazım-noktalama yanlışlıkları ve anlatım bozuklukları*. Master's thesis, Giresun University, Giresun.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Schmidhuber, J., Hochreiter, S., et al. (1997). Long short-term memory. *Neural Comput*, *9*(8), 1735–1780.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, *35*(5), 1285–1298.

Shirin, A. F., & Raseek, C. (2018). Replacing idioms based on their figurative usage. In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*, *IEEE*, 1–6.

Sirbu, A. (2015). The significance of language as a tool of communication. *Scientific Bulletin" Mircea cel Batran" Naval Academy*, *18*(2), 405.

Staron, T., Alaçam, Ö., & Menzel, W. (2018). Incorporating contextual information for language-independent, dynamic disambiguation tasks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Sun, A., Lim, E.-P., & Liu, Y. (2009). On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, *48*(1), 191–201.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, *9*(3), 293–300.

Teng, J., Zhang, D., Lee, D.-J., & Chou, Y. (2019). Recognition of chinese food using convolutional neural network. *Multimedia Tools and Applications*, *78*(9), 11155–11172.

Usur Hızlı, G. (2004). *Anlatım Bozukluklarının Düzeltilmesinde Geri Bildirimin Etkisi*. Master's thesis, Afyon Kocatepe University, Afyon.

Vapnik, V. (1998). The support vector method of function estimation. In *Nonlinear modeling*, 55–85.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988–999.

Vylomova, E., Cohn, T., He, X., & Haffari, G. (2016). Word representation models for morphologically rich languages in neural machine translation. *arXiv preprint arXiv:1606.04217*.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, *14*(1), 1–37.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yiğit, M. (2009). İlköğretim sekizinci sınıf öğrencilerinin yazılı sınavlarda yaptıkları anlatım bozuklukları üzerine bir inceleme. Master's thesis, Afyon Kocatepe University, Afyon.

Yıldırım, E., Çetin, F. S., Eryiğit, G., & Temel, T. (2015). The impact of nlp on Turkish sentiment analysis. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, *7*(1), 43–51.

Yoon, K. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1746–1751.

Zaytar, M. A., & El Amrani, C. (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *International Journal of Computer Applications*, *143*(11), 7–11.

Zhang, D., Lindholm, G., & Ratnaweera, H. (2018). Use long short-term memory to enhance internet of things for combined sewer overflow monitoring. *Journal of Hydrology*, *556*, 409–418.

Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017a). Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(3), 1–19.

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017b). Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, *29*(5), 1774–1785.

Zhu, X., Suk, H.-I., & Shen, D. (2014). A novel matrix-similarity based loss function for joint regression and classification in AD diagnosis. *NeuroImage*, *100*, 91–105.