ANALYZING STEMMING AND SENTENCE SIMPLIFICATION METHODOLOGIES FOR TURKISH MULTI-DOCUMENT TEXT SUMMARIZATION

by

Muhammed Yavuz Nuzumlalı

B.S., Computer Engineering, Boğaziçi University, 2011

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University

2015

ANALYZING STEMMING AND SENTENCE SIMPLIFICATION METHODOLOGIES FOR TURKISH MULTI-DOCUMENT TEXT SUMMARIZATION

APPROVED BY:

Assist. Prof. Arzucan Özgür (Thesis Supervisor)	....
Assoc. Prof. Banu Diri	....
Prof. Tunga Güngör	....

DATE OF APPROVAL: 15.01.2015

ACKNOWLEDGEMENTS

Foremost, I would like to thank my advisor Arzucan Özgür for her continuous and never ending support during my study and research. She always kept me motivated with her unlimited patience when I lost my concentration, and guided me with her comprehensive domain knowledge when needed. This thesis would not be finished without her supervision and encouragement.

Additionally, I would like to thank my thesis committee, Prof. Banu Diri and Prof. Tunga Güngör, for their insightful comments and feedback.

I would like to thank my friend Ferhat Aydın for his valuable efforts on collecting the multi-document summarization data set from scratch and writing human reference summaries. I also thank Burak Sivrikaya and Serkan Bugur for the reference summaries written by them.

Besides, I would like to thank the following human annotators for evaluating our text simplification system: Sümeyye Sena İbil Nuzumlalı, Ayşenur Işık, Beyza Arslan, Merve Terzioğlu, Tuğba Ata, Rumeysa Çamdereli, Ahmet Faruk Yücel, Nur Şeyda Koç, Ayşe Berre Karaman and Enes Aksu.

Moreover, I am thankful to The Scientific and Technological Research Council of Turkey (TÜBİTAK), The Science Fellowships and Grant Programmes Department (BİDEB) for providing me financial support with 2210 National Scholarship Programme for MSc Students during my Master of Science study.

Most importantly, I would like to thank my wife Sümeyye, the sweetest person in my life, for everything she did for me during this tough period. She always pushed me to study more when I became lazy in times. She is the one to be congratulated for her deep support both in good times and bad times.

ABSTRACT

ANALYZING STEMMING AND SENTENCE SIMPLIFICATION METHODOLOGIES FOR TURKISH MULTI-DOCUMENT TEXT SUMMARIZATION

Automatic text summarization is the task of generating a compact and coherent version of a given text document or a set of text documents. Although there is a vast number of studies for automatic document summarization on English, there is only a limited number of studies for other languages, especially for Turkish. Text simplification aims to reduce the grammatical or lexical complexities of the sentences. Automatic text simplification systems can be an important part of any NLP task to improve system performance. In this thesis, we analyzed the effects of applying different levels of stemming approaches such as fixed-length word truncation and morphological analysis and the effects of applying text simplification techniques for multi-document summarization (MDS) on Turkish, which is an agglutinative and morphologically rich language. We constructed a manually annotated MDS data set, and to the best of our knowledge, reported the first results on Turkish MDS. Additionally, we developed a rule-based text simplification system for Turkish that utilizes the syntactic features of the sentences to identify simplification patterns. Our results show that a simple fixed-length word truncation approach performs slightly better than no stemming, whereas applying complex morphological analysis does not improve Turkish MDS in terms of ROUGE scores. Applying simplification rules that split complex sentences to individual simpler sentences as a preprocessing step slightly improves summarization performance, whereas applying a compression-based simplification approach relying solely on rule matching decreases the obtained ROUGE scores.

ÖZET

KÖK BULMA VE CÜMLE SADELEŞTİRME YÖNTEMLERİNİN TÜRKÇE ÇOKLU BELGE ÖZETLEME ÜZERİNE ETKİLERİ

Otomatik belge özetleme, verilen bir ya da birden çok belgenin içeriğinin kısa ve kapsayıcı bir şekilde özetlenmesi işlemidir. Otomatik belge özetleme alanında İngilizce dili üzerine yapılmış çok sayıda çalışma olmasına rağmen, diğer diller için, özellikle Türkçe için, yapılmış çok az çalışma bulunmaktadır. Metin sadeleştirme, cümlelerin dil bilgisi ve sözlük dağarcığı açısından içerdikleri karmaşıklıkların azaltılmasını hedefler. Bu yüzden otomatik metin sadeleştirme sistemleri Doğal Dil İşleme alanındaki problemlerde sistem başarımını iyileştirecek önemli bir aşama olarak değerlendirilmektedir. Bu tezde, farklı seviyelerde uygulanan kelime kökü bulma yöntemlerinin ve cümle sadeleştirme tekniklerinin Türkçe dili için otomatik çoklu belge özetleme başarımı üzerine etkileri incelenmiştir. Otomatik özetleme sisteminin değerlendirilmesi için insanlar tarafından özetlenmiş bir veri kümesi derlenmiş, bildiğimiz kadarıyla Türkçe için ilk çoklu belge özetleme sistemi çalışması gerçekleştirilmiştir. Ayrıca cümlelerin sözdizimsel özelliklerini kullanan kural tabanlı bir cümle sadeleştirme yöntemi geliştirilmiştir. Elde edilen sonuçlarda, kelime sonundan harf atma tekniği en iyi başarımı elde ederken, detaylı morfolojik analiz yöntemleri başarımı ROUGE ölçütüne göre artırmamıştır. Ayrıca, verilen bir cümleyi birden fazla daha sade cümleye ayıran cümle sadeleştirme tekniklerinin özetleme sistemi öncesinde uygulanması başarımı az miktarda yükseltirken, cümle kısaltmaya dayalı cümle sadeleştirme teknikleri ROUGE ölçütü değerlerini düşürmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AMT	Amazon Mechanical Turk
CWD	Consecutive Word Detection
HMM	Hidden Markov Model
ILP	Integer Linear Programming
LSA	Latent Semantic Analysis
ML	Machine Learning
NLP	Natural Language Processing
MDS	Multi-Document Summarization
NMF	Non-negative Matrix Factorization
NP	Noun Phrase
PCFG	Probabilistic Context-Free Grammar
RASP	Robust Accurate Statistical Parsing
RC	Relative Clause
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RST	Rhetorical Structure Theory
SDS	Single-Document Summarization
SVM	Support Vector Machine
tf	term frequency
tf-idf	term frequency–inverse document frequency
WWW	World Wide Web

1. INTRODUCTION

Since the foundation of the World Wide Web (WWW), people have the ability to reach extensive, and continuously growing amounts of information resources and services via the Internet. However, this situation brings its own challenges such as finding the relevant documents, and absorbing a large quantity of relevant information [1].

Information overload is a continuously growing problem thanks to the online information which is a click away from us. Speier *et al.* describes the effect of this problem as follows [2]:

"Information overload occurs when the amount of input to a system exceeds its processing capacity. Decision makers have fairly limited cognitive processing capacity. Consequently, when information overload occurs, it is likely that a reduction in decision quality will occur."

To overcome this decision quality reduction possibility, it is important for people to be exposed to only the important portions of the available information. That is why systems that can automatically summarize a given content are crucial, and this motivation could explain the increasing interest to automatic summarization systems within the Natural Language Processing (NLP) community.

Automatic text summarization can be defined as the task of generating a more compact but still coherent version of a given text document or a set of text documents. Actually, the Oxford dictionary defines the word "summary" as *"a brief statement or account of the main points of something"*. Referring to this definition, a summary text is expected to be shorter than the original content, and preserve the important information and the overall meaning of the main theme. Correspondingly, any text summarization system should consider these points when producing summaries from given text documents.

The history of the automated text summarization research has begun in the late 50s. The very first studies in this area considered the problem as an information extraction problem, and depended only on surface level features like *word and phrase frequency* [3], *position of the sentences in the text* [4] and *key phrases* [5]. However, the remarkable increase of interest on this domain was observed in the 90s, possibly with the application of Machine Learning (ML) techniques on the NLP domain and the growing public usage of the WWW.

There are several kinds of categorizations used in the literature for summarization systems, each of which differentiate systems from a different perspective. Systems can be classified as *extractive* or *abstractive* depending on the sentence formation of the summary. Extractive summaries generally focus on selecting important sentences that discuss the main concept of the original text, and use the selected sentences in the summary without any modification. On the other hand, abstractive summaries strongly consider how to present the important content in a new and grammatically correct format using complex lexical and syntactic analysis and language generation techniques (to resemble human summaries).

Text summarization is categorized also according to the number of documents to be summarized. While *single-document summarization (SDS)* generates a summary of one text document, *multi-document summarization (MDS)* systems try to generate a summary from a set of related source documents. Multi-document summarization is fairly harder than single-document summarization, because it should also handle the redundancy occurring among different documents and maintain the cohesion between the sentences generated from different source documents.

Another differentiation is in what aspect a summary should be generated. *Generic* summaries try to represent all major topics in the original documents as equally important. *Query-focused* summaries detect what portion of the text is related to a given query and generate summaries correspondingly. *Comparative* summaries emphasize the differences between documents or portions of a document when producing the summary.

Although text summarization is a hot topic since the 90s, most of the studies have only been applied to a small set of languages (mostly English, additionally Chinese, Arabic and Spanish). The most important problem regarding applying the proposed summarization techniques to other languages is that generation of a manually annotated data set for the summarization task is very challenging. There are some very recent studies that try to collect manually annotated multi-lingual corpora for future multi-document summarization studies on other languages [6,7]. The MultiLing Workshop currently collected standardized corpora for eight languages (English, Arabic, Greek, Chinese, Romanian, Czech, Herbew, and Spanish), and they are planning to increase the language coverage in the future.

Another challenge for transporting the current knowledge base to other languages is the compatibility issues resulting from the different morphosyntactic features of the languages. The previous studies in the NLP domain have shown that methods proposed for languages like English do not generally work well for morphologically rich languages like Finnish, Turkish, and Czech, therefore additional methods considering the morphological structures of these languages are needed [8]. For instance, Turkish is an agglutinative language where root words can take many derivational and inflectional affixes. This feature results in a very high number of different word surface forms, which eventually leads to the data sparseness problem. Hakkani-Tür *et al.* analyzed the number of unique terms for Turkish and English and showed that the term count for Turkish is three times more than English for a corpus of 1M words [9].

There are only a few studies that focus on text summarization on Turkish, all of which are about single-document summarization. Altan [10] and Çığır *et al.* [11] proposed feature-based approaches, Özsoy *et al.* [12] and Güran *et al.* [13] used Latent Semantic Analysis (LSA), Güran *et al.* [14] applied non-negative matrix factorization and used consecutive words detection as a preprocessing step. Even though some of these studies applied morphological analysis methods, none of them analyzed their effects in detail.

Although the state-of-the-art systems that rely on the extractive summarization paradigm obtain promising results on evaluations, they have an implicit assumption that could cause the generated summaries to be less comprehensive in terms of the main theme of the documents. They consider document summarization as a problem that is handled only in inter-sentence level, rather than intra-sentence level. This is obviously not a genius assumption because there are sentences that discuss more than one topic in themselves, or sentences that contain explanatory clauses that give peripheral information about the main topic of the sentence that should be omitted from a summary document. We believe that detecting the unnecessary information contained in sentences should be a crucial step for a good summarization system. This requirement orients us to another NLP problem, which is automatic text simplification.

Automatic Text Simplification is the process of modifying a given text to reduce its grammatical or lexical complexity without losing its main content and overall meaning. This process may include modification of the lexicon, the syntax, or both. There are several studies that explore the effect of text simplification in the text summarization task [15–18]. These studies have shown that text simplification is indeed a valuable component for the summarization problem.

The main contribution of text simplification systems in the summarization domain is that they provide a way to get rid of the unnecessary parts of the sentences. This provides two main aspects: one is the gain from space, that is if we could reduce sentence length for summary sentences, we could include more sentences in the summary. The other aspect is that summary sentences would be more focused to the main content, therefore the overall summary would be more precise and readable.

To the best of our knowledge, this thesis proposes the first multi-document summarization system for Turkish. We used LexRank [19] as the main summarization algorithm of our system, applied and analyzed different levels of stemming methods such as complex morphological analysis and fixed-length word truncation as a preprocessing step. We also created the first manually annotated MDS data set for Turkish, which has been made publicly available for future studies.

Additionally in this thesis, we propose the first text simplification system for Turkish that could be used to improve the performance of various NLP tasks such as summarization and machine translation. We implemented a rule-based system which uses the syntactic features of the sentences to identify patterns that could be used to simplify sentences, by means of compression or splitting.

Lastly, we applied our text simplification system to the multi-document summarization problem by integrating it to our multi-document summarization system. We examined the effects of using simplification as a preprocesssing or postprocessing step for the MDS task. We also analyzed the effects of utilizing different combinations of the proposed simplification rules on the summary performance.

The rest of the thesis is organized as follows. Chapter 2 presents a comprehensive survey of the multi-document summarization and text simplification problems, as well as the applications of morphological analysis on Turkish for different Natural Language Processing (NLP) and Information Retrieval (IR) problems. Chapter 3 gives a brief introduction to the Turkish morphology. In Chapter 4 we give details about the stemming methods that we proposed and their effects on the summarization performance. The details about the created data set are also included in this chapter. Chapter 5 provides detailed information about our rule-based text simplification system, details for the simplification rules, and the evaluation of the system. Chapter 6 examines the effects of our text simplification approach on the MDS problem. We conclude our thesis in Chapter 7.

2. RELATED WORK

2.1. Text Summarization

2.1.1. Early History

The first studies that have pioneered automated text summarization research appeared in the late 50s, and focused on technical documents. The researches done by Luhn [3] and Baxendale [4] in IBM can be regarded as the pioneer works for the domain. Luhn claimed that the frequency of the words in an article can be used as a means of describing the importance of a particular sentence in the article. He ranked sentences using a function depending on the frequencies of the words in the sentences (also discarding very frequent words - namely the stop words), then selected the top k sentences to generate a summary.

Baxendile examined the positions of the sentences in the text as a decision-maker feature. Exploring nearly two hundred paragraphs to find out in which positions the topic sentence for a paragraph appears, the author observed that 85% of the topic sentences were in the first position and 7% were in the last position. Referring to the found result, he proposed that naively selecting one of these two positions would be an accurate way of summarizing documents. In fact, the position feature is still used in many complex summarization system as an important feature, also as a way of providing a strong baseline.

In 1969, Edmundson [5] utilized *cue words and phrases* (e.g. significantly, hardly etc.) and the words appearing in the titles or headings of the documents, in addition to features proposed in [3] and [4] while determining the importance of the sentences.

1990s witnessed an increased interest in summarization systems, probably with the contribution of applying machine learning techniques in the NLP domain and also the increase in the public use of the WWW. While the initial proposed systems generally assumed feature independence, later methods broke that assumption with methods like decision trees.

Probably the first study performed learning from data is done by Kupiec *et al.* [20]. They used a *naive Bayes classifier* to decide whether a particular sentence should be in the summarization or not. In addition to features proposed in Edmundson's work, features like sentence length and presence of uppercase words were used.

Lin modeled the sentence extraction problem without assuming that features are independent, and utilized decision trees to train a classifier using various features like IR signature (which detects the saliency of the words using a tf-idf like weighting), query signature (which is a score given to sentences depending on the query words they contain), presence of quotation, and presence of proper names [21].

Conroy and O'leary considered sentence extraction as a sequence classification problem and used a Hidden Markov Model (HMM) based approach [22]. By using a sequence classification model, they aimed to take into account the local dependencies between sentences. Sentence position, likeliness of words in a document, and sentence length were the features used by the authors.

Apart from the systems that depend on machine learning techniques, there were approaches which do not use machine learning and try to solve the problem via modeling the discourse structure of texts. Barzilay and Elhadad applied the lexical chain theory to the summarization problem [23]. They used WordNet [24] to identify the lexical chains, then selected summary sentences using some heuristics that depend on strong lexical chains.

Marcu developed a rhetorical parser to model the discourse relations between the sentences in a document, and selected summary sentences via a weighted function of rhetorical relations among sentences [25]. The study was based on the Rhetorical Structure Theory (RST) [26].

2.1.2. Multi-document Summarization

The first study about generating a single summary from multiple documents is done by McKeown and Radev in 1995, in which they tried to summarize a series of news articles on the same event together with a single summary text in an abstractive way [27]. Their method consists of two components: the content planner (the module that determines the important content to be included in the summary) and the linguistic component (the module that determines the words and the surface syntactic form of the summary). While their work was promising, it was not suitable for extending to other domains because it had domain specific heuristics generated by hand, also it was not generating summaries from raw text documents, rather it was using a database that is previously built by a template-driven message understanding system.

McKeown *et al.* [28] and Barzilay *et al.* [29] developed systems that generate summaries from a set of raw text documents via information fusion and word reformulation. Their system is composed of the following substeps: identification of themes (a set of similar paragraphs) in documents, information fusion (to detect similar phrases that are repeated enough to be included in the summary), and text reformulation via language generation. Theme identification was done by a clustering approach, where the similarity measure uses syntactic features like noun phrases, proper nouns, synsets from WordNet and positional and relational information between word pairs in addition to classical surface-level word features. The information fusion step includes detection and intersection of common phrases within themes utilizing the statistical parser developed by Collins [30] and converting parse results to a dependency grammar representation. At last, a grammatical text is generated using a language generation system.

In 2000, Radev *et al.* made a signficant contribution to the multi-document summarization domain by proposing a method for detecting and using *cluster centroids* (pseudo-documents which consist of words which have tf-idf scores above a predefined threshold in the documents of the cluster) to generate summaries [31]. The main contribution of the study was that the proposed system does not depend on any language dependent tools (i.e. language generation tool or any syntactic parser). The system was modeled with a bag-of-words paradigm, also it was easily scalable and domain independent.

In 2004, two similar studies that rely on graph theory were published at the same time [19,32]. Both works were heavily inspired from the PageRank [33] algorithm used by Google for web indexing. While TexRank was first applied to keyphrase extraction and then to summarization [34] solely, LexRank was specifically designed for multi-document summarization as a part of the MEAD summarization system which applies additional features like sentence position or sentence length. The main motivation for using the PageRank algorithm was the assumption that a sentence should be highly ranked for summarization if it is recommended by (i.e. connected to) many other highly ranked sentences.

A more recent graph-based study applied *minimum dominating set* theory to generate summaries. The system used an approximate minimum dominating set calculation algorithm to detect the minimum dominating set of a given sentence connectivity graph. Sentences that belongs to the minimum dominating set are included in the summary. They adapted the method to different summarization tasks such as generic, query-based, comparative, and update summarization, and achieved promising results.

An inevitable problem with systems that depend on the supervised learning paradigm is the need of an acceptable amount of labeled data on which classifiers can be trained [35]. The problem gets even bigger for the summarization domain because generating human summaries is a challenging and time consuming process. Moreover, annotator agreement can be low because summarization is somewhat a subjective process. Wong *et al.* aimed to solve this problem by applying a semi-supervised learning approach to the summarization task [36]. They co-trained two different classifiers iteratively by adding unlabeled training data with top confidence values to the labeled training set, and re-training the classifiers on the extended labeled data. They obtained comparable results with their supervised system, while gaining from manual labeling cost.

There is an interesting research done in 2013 by Christensen *et al.* that mainly aims to maintain coherence in the generated summary texts [37]. Unlike most of the previous systems that consider sentence selection and ordering as different steps, they merge these two processes to maintain coherence in the generated summaries. They model discourse relations among sentences via a graph based on indicators like discourse cues, co-reference and deverbal nouns. While their system performs worse than the previous studies based on the ROUGE evaluation, evaluations done by Amazon Mechanical Turk (AMT) annotators who are expected to compare summary quality between system summaries show that the proposed system has produced summaries that have significantly better quality compared to the previous systems. Their interpretation of the results is that their system sacrifices content coverage over coherence when needed, so they have lower ROUGE scores. Their AMT evaluation also gives insight about humans' tendency to favor integrity and cohesiveness over content completeness.

There were several studies that are based on mathematical reduction methods, especially on Latent Semantic Analysis (LSA) [38, 39]. LSA is a strong unsupervised mathematical technique that can model implicit semantic relatedness of the sentences based on co-occurrences of the words. It provides a way to identify important topics of the documents without any lexical features. Among studies that applied different variations of the LSA algorithm to the summarization domain, the main difference is about the way sentences are scored and chosen.

Gong and Liu used LSA to extract topics in the documents, and then to select the most representative sentence for each extracted topic [38]. The problem with their method is that they treated each topic as equally important, which is clearly not the case. To solve this, Steinberger and Ježek detected the importance of the topics and selected sentences that touch to several important topics as summary sentences [39].

2.1.3. Summarization for Languages Other than English

While most of the research done on automatic text summarization have only been applied to English, summarization data sets and systems for other languages like

Czech, Romanian, and Arabic have also been proposed in the recent years, thanks to the initiative taken by the MultiLing workshop organizers and community [40]. The workshop aims to contribute to three domains: multilingual multi-document summarization, multilingual summary evaluation, and multilingual summarization data collection and exploitation. An important output of the 2013 workshop is the creation of a multilingual manually annotated multi-document summarization corpus standarized for 8 languages, which can be used for future studies on non-English languages.

2.1.4. Summarization for Turkish

Previous studies on automatic summarization for Turkish only tackled the problem of single-document summarization (SDS). Altan [10] and Çığır *et al.* [11] proposed feature-based approaches that use a comprehensive set of surface-level features. Çığır *et al.* analyzed the effects of the individual features on the summarization performance, and showed that sentence position and sentence centrality feature are the most effective features.

Güran *et al.* compared a feature-based method that combines a rich set of features with manually tuned weights and an LSA based method, and showed that the LSA based method performs better, possibly due to it's ability to model the semantic relations between sentences [13].

Özsoy *et al.* evaluated a few LSA based summarization methods on Turkish data [12]. They applied two previously proposed LSA based methods, and proposed two different methods that are slight modifications of the existing methods. They show that one of their proposed methods — namely the *Cross* method which applies a preprocessing step to remove the possibility of selecting sentences that are not the core sentences representing the topic, but related to the topic in some way — performs best.

Güran *et al.* applied non-negative matrix factorization (NMF) and used different preprocessing methods such as stop word removal, stemming and consecutive words detection (CWD) — a method which utilizes Wikipedia URL links to find out commonly occurring consecutive words in documents like “Domuz Gribi” (Swine Flu) or “Anayasa Mahkemesi” (Constitutional Court) etc. [14]. Their results imply that while stemming does not increase performance, CWD slightly increases the summarization scores, though the increase is not statistically significant.

There is a study done by Pembe and Güngör which aims to summarize Turkish web documents in a query-based approach [41]. The system firstly analyzes the document structure (to detect the headings and the sections of the web documents in a structured way), then generates a summary by selecting sentences depending on features like location, term frequency and presence of heading words or query words. They compare their system with Google’s extracts that also depend on the query. Results show that thanks to the usage of document structure information, the proposed system performs significantly better than Google’s method.

The effect of morphological analysis for Turkish was analyzed in detail for Information Retrieval [42] and Text Categorization [43]. Can *et al.* showed that using a fixed-length word truncation approach performs similarly to complex lemmatization-based stemming for information retrieval [42]. Akkuş and Çakıcı obtained better results for text categorization with fixed-length word truncation rather than complex morphological analysis, but the difference was not significant [43]. For other morphologically rich languages, there is a case study on Greek by Galiotou et. al. [44]. They compared two available stemmers for Greek, and showed that one of the stemmers performs better. Depending on the result, they claim that if better morphological analyzers become available, summarization performance can further improve.

2.2. Text Simplification

There are several approaches applied to the text simplification problem, which are generally applied independently from each other and have distinct methodologies. While some methods consider the task as a lexical simplification problem in which complex words or phrases are identified and replaced with their simpler alternatives, others have made use of syntactic features of the sentences to detect grammatical complexities in the text and rewrite or drop these complex or possibly unnecessary parts. There are also a few approaches which apply the machine translation paradigm to the problem, especially the mono-lingual text-to-text generation approach.

2.2.1. Lexical Approaches

Lexical simplification mainly focuses on simplifying the parts of the text that have complex vocabulary rather than trying to simplify the grammar of the sentences. The task generally contains four steps, namely *complex word identification*, *generation of substitution alternatives*, *word sense disambiguation*, and *selection of best synonym alternative based on ranking.*

One of the pioneering studies on text simplification was published in 1999 by Carroll *et al.* [45]. They generate synonym alternatives for complex words using WordNet, then rank them using the Kucera-Francis frequencies of the words, the most common synonym is replaced with the original word.

One major drawback of this method and other early studies was that they may lose the semantic meaning of the complete sentence because of word sense ambiguity, that is a word can have more than one meaning. In order to distinguish the correct meaning, contextual information should be used. Word sense disambiguation methods that consider the context could be used to eliminate alternative synonyms that do not fit the meaning of the original words in the given context. One study utilized the built-in structure of WordNet, namely the *"synsets"* which are groups of words that are semantically similar, to make word sense disambiguation [46]. Another study makes disambiguation via context vectors [47]. They collect various information from the surrounding context of each word to build vector data. Disambiguation is then done by comparing vector similarities of the alternatives to the original words.

There is a recent study that focused on improving the frequency calculation metrics to enhance lexical simplification [48]. This study showed that the most effective method for frequency counting is the usage of Google Web 1T 5-gram corpus [49], which is an extremely huge corpus collected from the web sites indexed by Google.

2.2.2. Syntactic Approaches

Syntactic simplification aims to identify and resolve the grammatical complexities of the sentences. This may include splitting long and compound sentences to their clauses, resolving anaphora or just dropping the unnecessary sub-parts. The process is mainly composed of three phases, namely Analysis, Transformation, and Generation. In the analysis phase, sentences are parsed and parts that need simplification are determined. The transformation phase is where modifications are applied to the parse tree according to some rules (which can be generated manually or automatically). The generation phase generally includes methods to maintain relevance and cohesion on simplified versions.

The first system that performs syntactic simplification tried to automatically learn rewrite rules from an annotated corpus for domain specific sentence simplification, with the aim of improving performance of other NLP applications [50].

Siddharthan tried to formalize the relation between syntax and discourse of the sentence in order to preserve the conjunctive and anaphoric cohesive relations during simplification [51]. The author implemented a rule based simplification platform that applies rules to simplify sentences using conjunctions, relative clauses and appositions. After rules are applied, a regeneration procedure is performed to correct anaphoric and conjunctive cohesion by correctly ordering the splitted sentences.

Jonnalagadda and Gonzalez developed an open source sentence simplification system to improve information extraction performance in the biomedical domain [52]. They performed syntactic simplification by traversing through the Penn trees of the sentences to find simplification patterns that are identified by predefined hand-written rules. Penn trees were generated from the McClosky parser [53]. They showed that text simplification significantly improved the protein-protein interaction extraction task.

2.2.2.1. Dependency Parser Based Simplification. Fillipova and Strube proposed an unsupervised method for the text simplification task by utilizing the dependency trees of the sentences [54]. They evaluated two different parsers, namely the RASP [55] and the Stanford PCFG parser [56]. Their method is composed of three steps. In the *tree transformation* step, dependency trees are modified using predefined rules like *"inserting an explicit node, and connecting verbs to that node"* and *"decomposing conjunctions"*. The *tree compression* step formulates the task as an optimization problem and solves using an integer linear programming (ILP) approach that uses an objective function which contains word importance values (that are calculated similarly to the tf-idf calculation) and probability of dependencies as parameters. The last step is *tree linearization* in which they simply put the remaining words in the original order for English, but use a more complex approach that considers grammatical features for German.

Jonnalagadda *et al.* developed a text simplification system that performs simplification using both syntactic and non-syntactic features to improve performance of the syntactic parsers on the biomedical domain *jonnalagaddaEtAl09*. They used Link Grammar Parser [57] to parse sentences. For simplification, they firstly split the sentence from commas. Then, they iteratively check all the clauses starting from the first clause, whether the corresponding clause can be a sentence by giving the clause to the Link Parser and checking if the parse result contains a *"S link"* which indicates that the clause can be a sentence. If S link is found, sentence is split from here, and the next clause is evaluated. If not, the process continues by attaching the next clause to the previous one, and checking for S link again, recursively. Their method improved the parser performance significantly.

2.2.3. Machine Translation Approaches

There are a few recent studies that ported machine translation techniques to the text simplification task [58, 59]. Zhu *et al.* proposed a tree-based simplification model that performs transformations based on statistical machine translation techniques [58]. Their model covers splitting, dropping, reordering, and word/phrase substitution tasks integrally. To train their system, they generated an aligned corpus obtained by mining Wikipedia and Simple Wikipedia. On the other hand, Wubben *et al.* applied a phrase-based machine translation procedure trained on a monolingual parallel corpus [59]. Different to previous systems, they compared their system with previous studies using human judges who evaluated the quality of the simplified text.

2.2.4. Text Simplification for Summarization

The first theoretical application of text simplification approaches on the text summarization task appeared in 2000 by Knight and Marcu [15]. They considered sentence compression as an initial step for the larger problem of text summarization. Two methods were applied and compared that are based on two different statistical models, namely the noisy-channel model and decision trees. The system take the parse trees of the sentences generated by Collins' Parser as input. Grammaticality is ensured checking alternative compressions with a grammar learned from Penn Treebank.

Siddharthan *et al.* proposed the first system that practically applies text simplification on multi-document summarization [16]. They performed syntactic simplification by identifying and then removing appositions and relative clauses using shallow techniques that are based on local context and animacy information obtained from WordNet. The only used lexical tools were a POS-tagger and a simple noun-chunker. Their results show that applying simplification significantly improves the ROUGE scores of the summarizer system.

Vanderwende *et al.* applied sentence simplification on topic-focused multi-document summarization [17]. They generated multiple shortened alternatives for each sentence by dropping patterns such as *noun appositives*, *gerundive clauses*, *nonrestrictive relative clauses* and *lead adverbials and conjunctions* via performing a set of heuristic rules on the parse trees of the sentences. After these steps, all the alternative simplified sentences are given to the summarizer system together with the original sentences, with the assumption that their summarization system has the ability to detect and handle redundancy.

In a recent study, the effect of text simplification is analyzed for multi-document summarization on Portuguese [18]. The authors developed a cluster-based summarization system, and analyzed the effect of performing text simplification as a preprocessing step. They removed appositions, adjectives, adverbs, parentheticals and relative clauses by detecting them using a constituency parser. Applying text simplification gave promising results. They also evaluated the effect of using simplification before or after clustering, and showed that the results were indifferent.

3. TURKISH MORPHOLOGY

Before diving into the details of our proposed system, we provide a brief description of the morphological structure of the Turkish language.

Turkish is an agglutinative language with a productive morphology. Root words can take one or more derivational and inflectional affixes; therefore, a root can be seen in a large number of different word forms. Another issue is the morphological ambiguity, where a word can have more than one morphological parse.

Table 3.1. Different word forms and their morphological analysis for the stem "gör" (to see). The derivational boundaries are marked with (DB).

Word	Analysis
gören *(the one who sees)*	gör+en(DB)
görülen *(the one which is seen)*	gör+ül(DB)+en(DB)
görüş *(opinion)*	gör+üş(DB)
görüşün *(your opinion)*	gör+üş(DB)+ün
görüşler *(opinions)*	gör+üş(DB)+ler
görüşme *(negotiation)*	gör+üş(DB)+me(DB)
görüşmelerin *(of negotiations)*	gör+üş(DB)+me(DB)+ler+in

In Turkish, verbs can be converted into nouns and other forms, and nouns can be converted into verbs and other grammatical constructs, through affixation [60]. Table 3.1 shows an example list of different word forms for the stem "gör" (to see). All the words in the table have the same root, but the different affixation via derivational and inflectional suffixes leads to different surface forms which may have similar or different meanings . When the surface forms of these words are used in a summarization system, they will be regarded as totally different words. However, if a morphological analysis method is applied to the sentences before giving them to the summarization system, words with similar meanings can match during the sentence similarity calculations. That is the main intuition behind our proposed method.

4. TURKISH MULTI-DOCUMENT SUMMARIZATION

4.1. Methodology

This section contains detailed information about the application of different levels of morphological features during the summarization process.

4.1.1. Stemming Policies

In this section, we explain the different stemming methods that we investigated. All of these stemming policies have been applied as a preprocessing step, before giving the documents as input to the summarization system. There are four different methodologies:

4.1.1.1. Raw. In this method, we take the surface forms of words, without applying any stemming.

4.1.1.2. Root. This method takes the most simple unit of the words, namely the root form. For example, in Table 3.1, the words "gören", "görüşün", and "görüşmelerin" have the same root (gör), so they will match during sentence similarity calculations done by the summarization system.

4.1.1.3. Deriv. Using the Root method may oversimplify words because some words that are derived from the same root may have irrelevant meanings. In the above example, "görüşler" and "gören" have different meanings, but they have the same root (gör). If we use the Root method, we will lose the semantic difference between these two words. In order to solve this oversimplification issue, we propose to preserve the derivational affixes, and only remove the inflectional affixes from the words. In this method, "görüşler" and "gören" will not match because when we remove only the inflectional affixes, they become "görüş" and "gören". On the other hand, the words "görüşler" and "görüşün" will match because their Deriv forms are the same, which is "görüş". The intuition behind this policy is that derivational affixes are the ones that modify the meanings of the words that they are appended, therefore they should be preserved to distinguish the semantically different words having the same root in the similarity calculations.

4.1.1.4. Prefix. In Turkish, affixes almost always occur as suffixes, not prefixes.[1] Additionally, applying morphological analysis methods is a time consuming process, and may become an overhead for online applications. Therefore, a fixed-length simplification method is also tried, since it is both a fast method and can help to match similar words while calculating similarities by taking the first N characters of words which have lengths larger than N. We explored a wide range of N from 3 to 12 to find the best threshold value.

4.1.2. LexRank

As the summarization algorithm, we used LexRank [19], which is a salient graph-based method that achieves promising results for MDS. In LexRank, first a sentence connectivity graph is constructed based on the cosine similarities between sentences, and then the PageRank [33] algorithm is used to find the most important sentences.

4.2. Experimental Setup

4.2.1. Data Set

One of the greatest challenges for MDS studies for non-English languages is the lack of available manually annotated corpora necessary for training and evaluating the systems. That is also the case for Turkish, there does not exist a manually annotated data set. In this study, we have collected and manually annotated a Turkish MDS data

[1]Actually in Turkish, the only regular use of prefixation is to intensify the meaning of adjectives (and less commonly of adverbs), such as "dolu" (i.e., full) and "dopdolu", or "tamam" (i.e., complete) and "tastamam" [60]. This type of intensifying is generally not used, also not suitable for news articles.

set from scratch, and made it publicly available for future studies.[2]

In order to match the standards for MDS data sets, we tried to follow the specifications of the DUC 2004 data set while generating the data set. Our data set consists of 21 clusters, each consisting of around 10 documents. We selected 21 different topics from different domains (e.g., politics, economics, sports, social, daily, and technology), and selected 10 documents on average for each topic. The documents were obtained from the websites of various news sources. The average number of words per document is 337, and the average number of letters in a word is 6.84 for the collected data set.

For manual annotation, we have generated three reference summaries for each cluster. Cluster documents are sent to three annotators different from the authors. We required the human summaries not to exceed 120 words for the summary of each cluster. We determined the above number by interpolating the limit defined in DUC 2004 data set (which is 665 bytes) to corresponding approximate number of words in Turkish.

The annotation guidelines provided to the annotators are presented in Appendix A.1.

4.2.2. Tools

4.2.2.1. Turkish Morphological Analysis. In order to perform different levels of morphological analysis on documents, we used the two-level morphological analyzer of Oflazer [61] and the perceptron-based morphological disambiguator of Sak *et al.* [62]. The morphological analyzer is actually a rule-based system which takes a word and divides it to its root word and the affixes that it took, and outputs all possible analyses. Table 4.1 shows the morphological analyzer output for the word “ocağında”.

The word “ocağında” has two possible parses because it may have the meaning *on his/her oven* or *on your oven* depending on the context. To solve this ambiguity

[2]The data set can be retrieved from the following github repository: `https://github.com/manuyavuz/TurkishMDSDataSet_alpha`

Table 4.1. Output of the Morphological Analyzer for the word "ocağında"

ocağında
ocak+Noun+A3sg+P3sg+Loc *(on his/her oven)*
ocak+Noun+A3sg+P2sg+Loc *(on your oven)*

problem, we used the morphological disambiguator of Sak *et al.* which is a perceptron based disambiguator trained with a corpus of about 750,000 tokens from news articles. The accuracy of the disambiguator has been reported as 96.80% [62]. We generated the Root and Deriv forms of the words using the disambiguator output.

4.2.2.2. MEAD Summarization Toolkit. We used MEAD, which is an open-source toolkit created for extractive MDS, in our experiments [63]. MEAD is a comprehensive tool that handles all the necessary processes to generate a summary from a set of documents (sentence ranking, selection, re-ordering etc.).

We used the LexRank implementation that comes with MEAD as a feature, together with the Centroid and Position features (each feature is equally weighted) in the sentence scoring step. We used the default classifier of MEAD, which simply takes the linear combination of the scores given by the above features as the aggregate score for each sentence.

After the sentences are scored using LexRank and the other features, the reranker step starts, in which sentence selection is performed. In this step, summary sentences are determined in a manner that handles redundancy across sentences. We used the default reranker which takes sentences one by one according to their scores into the summary, but skips sentences that are too similar to sentences that are previously selected as summary sentences. Similarity is calculated using term frequency (tf) based cosine similarities between sentences, and sentences having cosine similarity higher than a predefined threshold are considered as too similar.

After selecting a sufficient number of sentences that fill the given word limit, MEAD includes the selected sentences to the summary in the order they appeared in the documents.

We forced the generated summaries not to exceed 120 words. However, we define the following exception in order to preserve the readability and the grammaticality of the generated summary. For a candidate sentence S having n words, if the absolute difference between the threshold (which is 120) and the summary length including sentence S (say N_w) is less than the absolute difference between the threshold and the summary length excluding sentence S (say N_{wo}), and if N_w is less than 132 (which is $120 * 1.1$), we allow the summary to exceed the threshold and add sentence S as the last summary sentence.

We also required the length of the candidate summary sentences to be between 6 and 50 words (which we found empirically) in order to increase the readability of the summaries. The reason behind applying this filtering is that very short sentences generally do not contain much information to become a summary sentence, whereas very long sentences decrease the readability and fill a significant percentage of the summary limit.

4.2.2.3. ROUGE. For the evaluation, we used ROUGE (Recall-Oriented Understudy for Gisting Evaluation), which is a standard metric for automated evaluation of summaries based on n-gram co-occurrence [64]. We report ROUGE-1 (based on unigrams), ROUGE-2 (based on bigrams), and ROUGE-SU4 (matching bigrams with skip distance up to 4 words) scores in our experiments.

In a recent study [65], ROUGE-2 with stemming and no stopword removal has been shown to agree with manual evaluations most, so we give importance to ROUGE-2 while interpreting the results. Also, we computed the ROUGE scores for two different approaches, namely with partial stemming (we converted each word in the system summaries and reference summaries to their corresponding *Deriv* forms before giving to ROUGE for evaluation), and with no stemming (we give the summaries without any modification, i.e., their *Raw* forms). We do not apply stopword removal.

We interpret all results with ROUGE scores with their corresponding F_1-measures using *Deriv* stemming in the following sections.

4.3. Evaluation and Results

We ran MEAD with the proposed stemming policies using different cosine similarity threshold values to analyze the effect of the similarity threshold on the summarization performance. As we stated earlier, after the sentences are ranked according to their scores using LexRank and the other features, the similarity threshold is used to decide whether to include a sentence to the summary or not. A sentence is not included to the summary, if its similarity to a previously picked sentence is higher than the given similarity threshold value.

In our preliminary experiments, we used the default similarity threshold 0.7, which was found empirically by the MEAD developers for English. However, it produced poor results on the Turkish data set.

Figure 4.1 shows the F_1-measures for the ROUGE-2 metric for policies with different thresholds. We explored threshold values ranging from 0.15 to 0.7 to see the effect on the summary performance. Although we explored these values for all Prefix thresholds from 3 to 12, we report here only Prefix8 (taking the first 8 letters of the words) which is the best one among them in order to make the chart more readable. We also examined the effect of similarity threshold on other ROUGE metrics, and obtained very similar trends, therefore we do not report them here.

In general, Raw and Prefix8 achieve better performances with lower threshold values, whereas Root and Deriv operate better with relatively higher threshold values. However, there is no such a direct correlation between threshold values and ROUGE scores. Raw and Prefix8 policies achieve their best when threshold is 0.35, whereas the

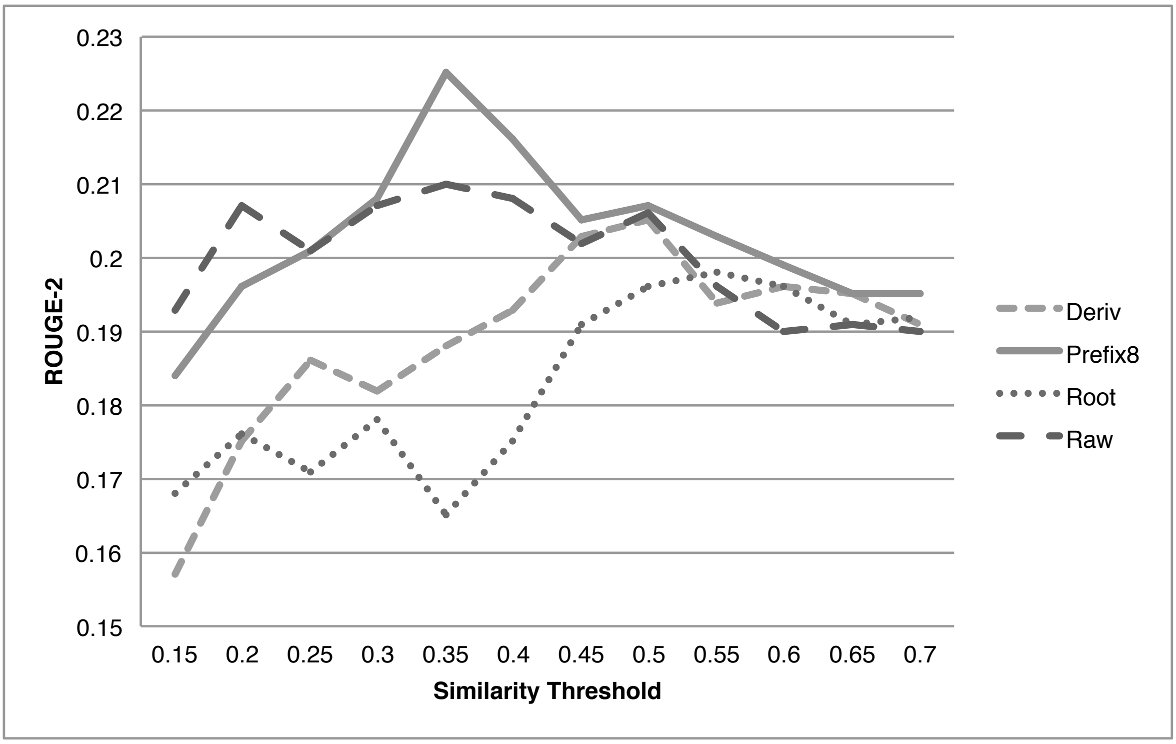

Figure 4.1. ROUGE-2 scores for different similarity threshold values.

bests for Deriv and Root are 0.5 and 0.55, respectively. Starting from 0.55, ROUGE scores for all policies seem to converge to very similar values. As we discussed before, in Turkish, words with similar meanings can occur in text with different surface forms due to their inflections. Such words can not be matched during similarity computation if morphological analysis is not performed. Therefore, using higher similarity threshold values cause very similar sentences to occur together in the summaries, and eventually result in poor scores.

Table 4.2 shows the best scores obtained by each policy. While the Prefix policies generally outperform the Raw policy, the Prefix8 policy achieves the best scores for each metric. On the other hand, the policies that apply complex morphological analysis (i.e. Root and Deriv) are not able to outperform the simple Prefix and Raw policies. The Deriv policy performs slightly lower than the Raw policy, whereas the Root policy obtains the lowest scores for each metric on our data set.

Table 4.2. Best scores for different policies.

Policy	ROUGE-1	ROUGE-2	ROUGE-SU4
Prefix8	**0.457**	**0.225**	**0.231**
Prefix7	0.450	0.223	0.227
Prefix10	0.450	0.219	0.226
Prefix6	0.448	0.218	0.224
Prefix11	0.450	0.215	0.221
Prefix12	0.450	0.214	0.219
Prefix9	0.448	0.213	0.222
Raw	0.445	0.210	0.217
Prefix4	0.446	0.209	0.214
Prefix5	0.446	0.206	0.214
Deriv	0.434	0.205	0.214
Prefix3	0.433	0.200	0.210
Root	0.431	0.198	0.207

4.3.1. Discussion

The results show that using a simple fixed-length prefix policy outperforms all other methods, and applying complex morphological analysis does not improve Turkish MDS. The poor performance of the Root policy is somewhat expected due to the fact that, if we preserve only the roots of the words, we lose the semantic differences among the surface forms provided by the derivational affixes. On the other hand, the reason behind the observation that Deriv performs slightly lower than Raw is not obvious.

In order to further analyze this observation, we used an entropy based measure, which is calculated as shown below, to quantify the homogeneity of the clusters in the data set in terms of the variety of the surface forms corresponding to the Deriv forms of each word in the cluster. We first compute the entropy for each Deriv form in a cluster. The entropy of a Deriv form is lower, if it occurs with fewer different surface forms in the cluster. The entropy of a cluster is computed by summing the entropies of the Deriv forms in the cluster and dividing the sum by the number of words in the cluster (i.e. N).

$$
\begin{aligned}
D_{Deriv_i} &= \{t \mid t\, inflected\, from\, Deriv\, i\} \\
H(Deriv_i) &= \sum_{t \in D_{Deriv_i}} -p(t) \log p(t) \\
H(C) &= \sum_i \frac{H(Deriv_i)}{N}
\end{aligned}
$$

To compare with the data set clusters, we generated random document clusters by randomly selecting 10 different clusters and then randomly selecting one document from each selected cluster. The average entropy value for the data set clusters and the random clusters were 4.99 and 7.58, respectively. Due to this significant difference, we can hypothesize that the documents about the same topic show a more homogeneous structure. In other words, a Deriv form is usually seen in the same surface form in a cluster of documents which are about the same topic. Therefore, the Deriv policy does not contribute to the task of summarizing documents about the same topic.

During evaluation, we ran ROUGE with the Deriv versions of the human summaries and the system summaries in order to match semantically similar words having different surface forms. We also experimented with ROUGE using the Raw versions, but the results followed very similar patterns, so those results were not reported.

5. TEXT SIMPLIFICATION FOR TURKISH

In this chapter, we describe our rule-based syntactic text simplification system developed for the Turkish language. Section 5.1 explains our methodology and the identification of the rules in detail. Section 5.2 describes the collected data set and the syntactic tools that we use. The last section interprets the results of the human evaluations performed to quantify the quality of the system output, and presents a discussion about the issues that could affect the system performance.

5.1. Methodology

We followed the syntactic simplification paradigm while developing our simplification system. In syntactic text simplification, grammatical and syntactic features of the sentences are utilized to identify parts of the sentences which are worth simplifying. In order to determine the syntactic relations between the words in the sentences, we have utilized the output of the morphological analyzer, together with the dependency relations between words that are obtained using a dependency parser specifically developed for Turkish [66].

The previous studies on the text simplification domain examined the effects of using a comprehensive set of grammatical features like *conjunctions, appositives, relative clauses, adjectives and adverbs* while simplifying text. Among these, Rhetorical Structure Theory [26] considers *appositives and non-restrictive relative clauses* as the *parentheticals*, which are structures that provide background information on entities and relate entities to the discourse of the text. Therefore, removal of these parts rather than adjectives or adverbs would be more effective while simplifying sentences, because they do not affect the main meaning of the sentences. This is the theory behind our choice of simplification rules.

5.1.1. Simplification Rules

Our simplification system applies three rules in the simplification step. One of the rules aims to shorten the sentence by dropping the detected relative clauses. The other two rules aim to split compound sentences into two or more sentences using comma or semicolon conjunctions and the *-ken* affix. These rules are explain in detail in the following subsections.

5.1.1.1. RULE-1: Drop relative clauses. This rule tries to identify the relative clauses in the sentences and drop them to get rid of the possibly unnecessary parts of the sentences.

In English, relative clauses are formed by means of relative pronouns such as *who, which and that.* On the other hand, in Turkish, relative clauses are formed by morphemes such as *-En, -DIk, -mIş and -(Y)EcEk* that are attached to the verb stem of the clause sentence. Unlike English, the above morphemes used in Turkish contain the tense information of the clause. Consider the following sample sentence and its English translation. Its dependency parse tree is given in Figure 5.1.

[$_{\text{NP}}$ [$_{\text{RC}}$ Japonya'da yaşay**an**] kardeşim] yarın geliyor.

[$_{\text{NP}}$ My brother, [$_{\text{RC}}$ *who* lives in Japan]], is coming tomorrow.

Here, the relative clause is formed using the *-an* affix, which makes the tense of the clause present tense.

Relative clauses provide additional information about the entity that they modify. Therefore, removing them would not effect the main meaning of the sentence too much. This is the reason why we applied this rule.

The morphological analyzer identifies the above morphemes with the *PresPart*

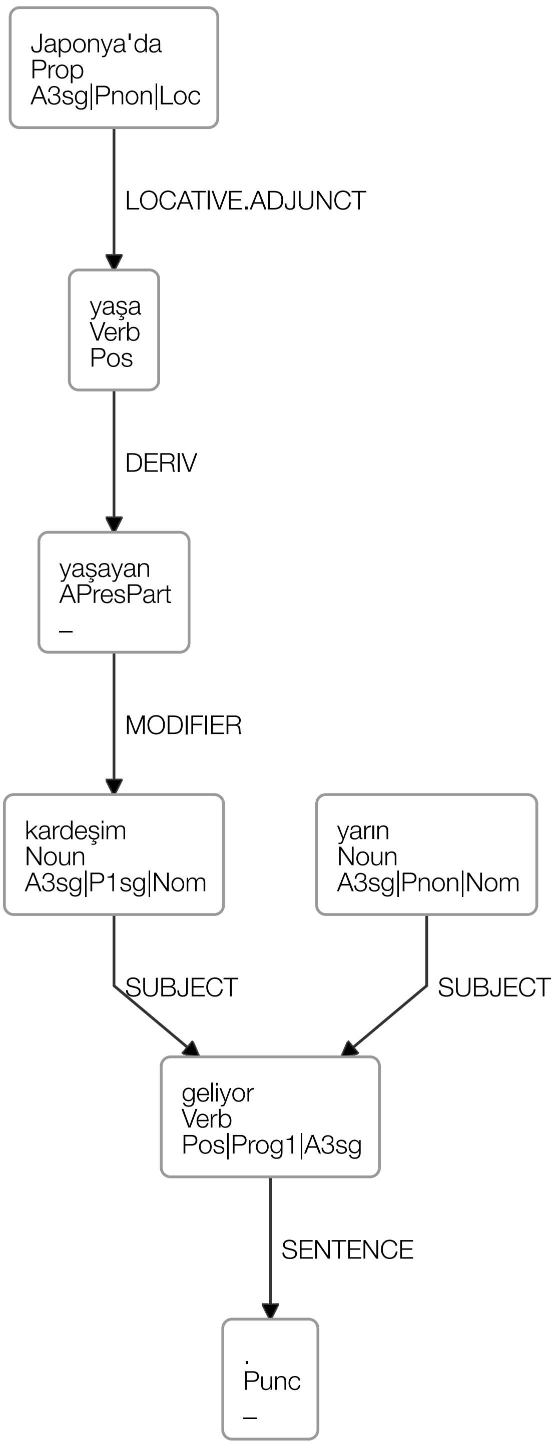

Figure 5.1. Dependency parse of the sentence *"Japonya'da yaşayan kardeşim yarın geliyor."*. This sentence matches the pattern for RULE-1.

or *PastPart* tags according to their tenses in the sentences. For example, here is the analyzer output for the word *"yaşayan"*:

yaşa+Verb+Pos^DB+Adj+PresPart

While applying simplification, we use this information to detect the words signaling relative clause boundaries, then we traverse the dependency parse tree to remove all the words that are connected to the clause indicator word together with itself. We only consider the adjective words (identified via the *Adj* tag) having the PresPart tag, because we do not want to remove clauses which themselves are noun phrases with the assumption that they are most probably a subject or an object of the main sentence.

Currently we only match PresPart tags and not PastPart tags, because while developing rules via analyzing the sentences in our development data set, we observed that words with PresPart tag are much more frequent than the ones with the PastPart tag. In addition, words having the PastPart tag are generally in noun form.

Actually for this rule, we would want to get rid of only the non-restrictive relative clauses, because restrictive relative clauses generally contain information that is definitive for the entity that they are related to. In English, it is relatively easy to detect if a relative clause is restrictive or non-restrictive, since a non-restrictive relative clause is typically preceded by a pause in speech or a comma in writing, whereas a restrictive clause normally is not [67]. However, in Turkish, there are no such formal distinctions between restrictive and non-restrictive relative clauses [68]. One possibility could be to identify the type of the relative clause using contextual information and discourse, however this is a fairly complex task. Therefore, we currently do not make a distinction between different types of relative clauses.

5.1.1.2. RULE-2: Split compound sentences into their sub-sentences. In Turkish, there are compound sentences which connect two related sub-sentences with commas or semicolons. These sentences can be divided into their sub-sentences as a simplification step.

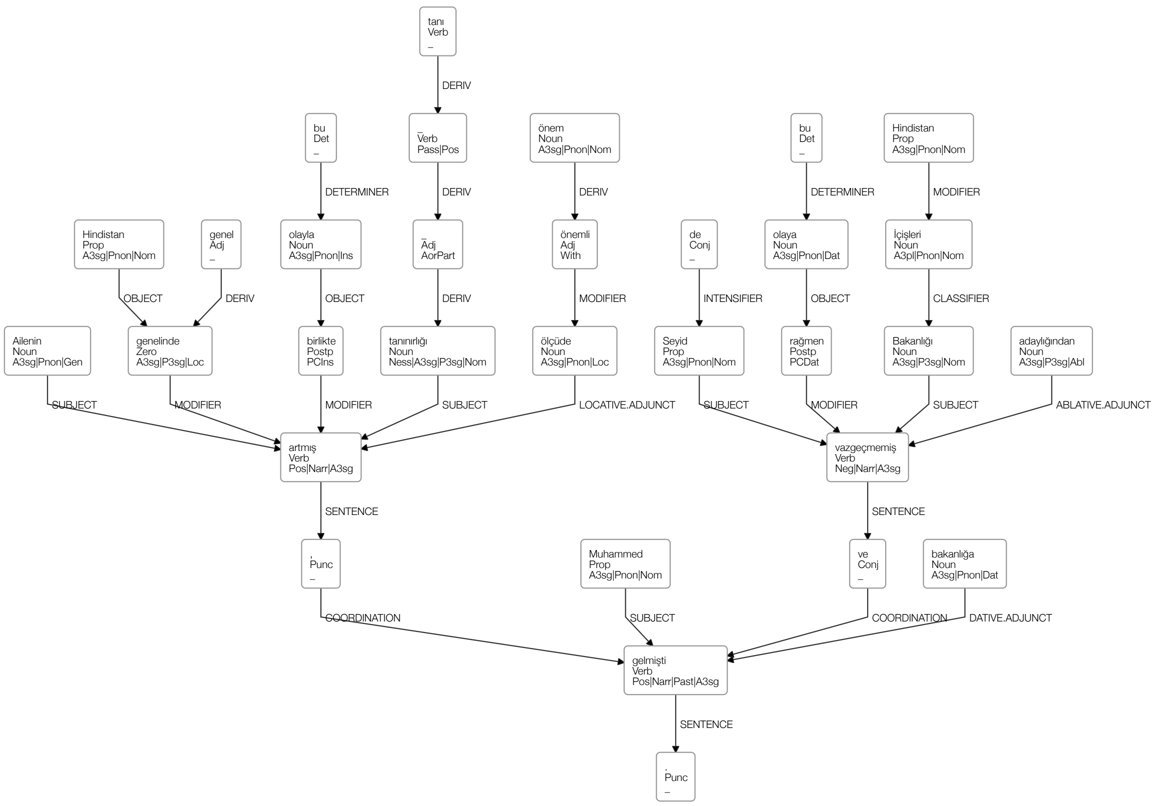

Figure 5.2. Dependency parse of the sentence *"Ailenin Hindistan genelinde bu olayla birlikte tanınırlığı önemli ölçüde artmış, Muhammed Seyid de bu olaya rağmen Hindistan İçişleri Bakanlığı adaylığından vazgeçmemiş ve bakanlığa gelmişti."*. This sentence matches the pattern for RULE-2.

Consider the following sentence and its dependency parse tree, which is given in Figure 5.2.

Ailenin Hindistan genelinde bu olayla birlikte tanınırlığı önemli ölçüde artmış, Muhammed Seyid de bu olaya rağmen Hindistan İçişleri Bakanlığı adaylığından vazgeçmemiş ve bakanlığa gelmişti.

Here, this long sentence can be divided from the comma punctuation mark to following sub-sentences.

Ailenin Hindistan genelinde bu olayla birlikte tanınırlığı önemli ölçüde artmış.

Muhammed Seyid de bu olaya rağmen Hindistan İçişleri Bakanlığı adaylığından vazgeçmemiş ve bakanlığa gelmişti.

For this rule, we traverse the dependency parse tree of the sentence and search for a node that is connected to a comma or semicolon punctuation with a SENTENCE dependency label. The SENTENCE dependency indicates that the node is in verb form and it is the verb of a sub-sentence. If such a pattern is found, we divide the sentence from the corresponding punctuation into two individual sentences.

5.1.1.3. RULE-3: Split sentences from words having the "-ken" affix. In the English grammar, the "while" word can be used when talking about different activities happening at the same time. The corresponding conjunction for Turkish is the affix *"-ken"*. Consider the following sentence and its its dependency parse tree given in Figure 5.3.

*Müşerref Akıcı çarpmanın etkisiyle kaldırıma savrulur**ken**, Birol Şimşek otobüsün altında kaldı.*

***While** Müşerref Akıcı skid towards the sidewalk due to the severity of the crash, Birol Şimşek remained under the bus.*

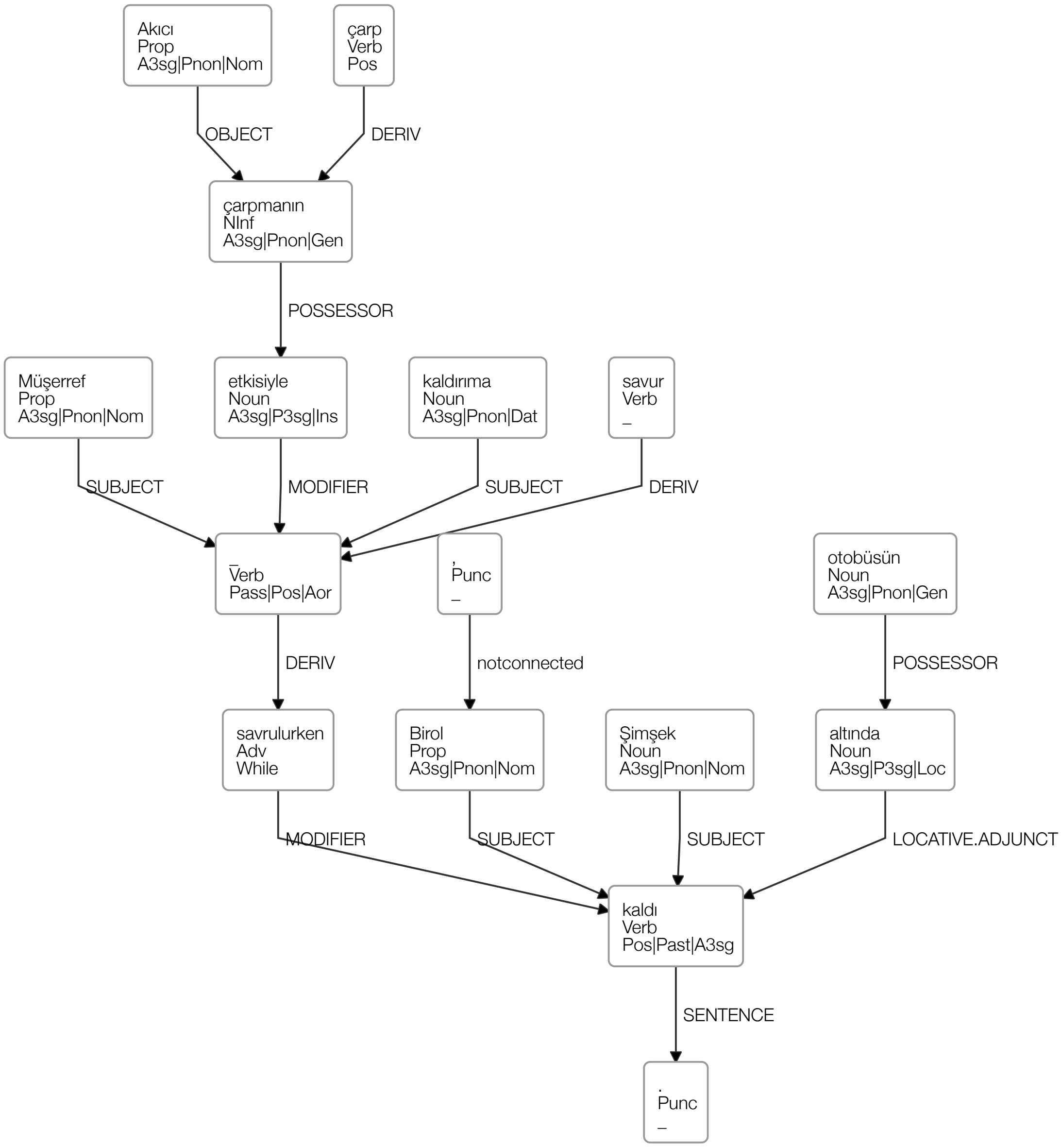

Figure 5.3. Dependency parse of the sentence *"Müşerref Akıcı çarpmanın etkisiyle kaldırıma savrulurken, Birol Şimşek otobüsün altında kaldı."*. This sentence matches the pattern for RULE-3.

Here, we can divide this sentence from the word “savrulurken”, because the sentence contains two events that can also be expressed with two different sentences individually.

For this rule, we traverse the dependency tree of the sentence and look for a node that has the *While* tag in its morphological parse. If such a word is found, the sentence is divided from this node into two sentences.

5.2. Evaluation

5.2.1. Dependency Parser

For dependency parsing, we used the dependency parsing model developed by Eryiğit *et al.* [66]. The developed model was generated using the MaltParser [69], a data-driven dependency parsing system, trained with support vector machines (SVMs) using the METU Sabancı Turkish Treebank. The Turkish dependency parser is reported to obtain 75.82% unlabeled and 65.68% labeled attachment scores and 67% F-score for Turkish on the CoNLL 2006 data set of the shared task on multilingual dependency parsing.

5.2.2. Data Set

For the identification of the rules, we have collected approximately 100 sentences from online news resources that are complex enough to be simplified. The sentences are parsed with the pipeline which consists of the morphological analyzer, morphological disambiguator, and dependency parser. The patterns for the corresponding rules were identified via manually examining the parse trees of the sentences in detail.

For the test data set, we have collected another set of 205 sentences, again from different online news sources. For this collection, we do not discard any sentences, therefore there are also simple sentences that do not need to be simplified. Note that both the development and the test sentences have been extracted from new articles that are not included in our summarization data set.

5.2.3. Human Evaluation

We ran our simplification system on the test data set in order to perform simplification. Our system proposed a simplification for approximately half of the test data set (actually 103 out of 205 sentences were simplified).

We evaluated the quality of the proposed simplifications manually with human judges consisting of ten people.[3] We only gave the sentences that were simplified by our system to the judges. For the evaluation criteria, we defined the following metrics which were inspired from the study by Woodsend and Lapata [70].

- *Simplicity*: Quantifies the simplicity of the proposed simplification compared to the original sentence.
- *Grammar*: Quantifies the grammatical correctness of the proposed simplification.
- *Meaning*: Quantifies how well the proposed simplification represents the main theme of the original sentence.

The judges have been told that their answers will be used to evaluate the quality of an automatic text simplification system. However, no any information about the implementation details of the system and the applied rules were provided to the judges. We required the human evaluators to give a score between 1 and 5 to each metric for each sentence. Table 5.1 shows the interpretations of the five possible scores that can

Table 5.1. The possible scores that can be given by the human judges and their interpretations.

1	2	3	4	5
Very Poor	Poor	Moderate	Good	Very Good

[3] The evaluation guidelines are provided in Appendix A.2

Table 5.2. The average scores given to each metric by the human judges. The numbers in parenthesis are the standard deviations.

Simplicity	Grammar	Meaning
3.334 ($\pm$0.301)	3.936 ($\pm$0.326)	3.609 ($\pm$0.387)

be assigned by the human judges.

Table 5.2 shows the average scores assigned by the judges for the three quality metrics, together with their standard deviations. The scores sit between "Moderate" and "Good". The grammar metric is very close to "Good". Relying on this empirical results, we can hypothesize that our simplification system proposes acceptable or good simplifications in general.

5.2.4. Discussion

There are some known deficiencies in our current implementation. Here we discuss these issues.

An important problem is the low accuracy of the dependency parser. The dependency parser that we use for Turkish has some characteristic errors that seem to be repeated during parsing. During our analysis of the dependency parse trees of the sentences in the development set, we observed that the parser generally could not detect correctly the dependencies among words that form named entities or simple adjective clauses that have numeric adjectives.

Figure 5.4 shows a sample dependency parser result to demonstrate a common parser error. In this parse tree, the node having the word "3" is considered as a subject of the verb "açıkladı", while it should have been connected to the node "gün", because it actually is a modifier of the word "gün". When we try to simplify this sentence using RULE-1 to drop the relative clause of the sentence, we get the following output, which

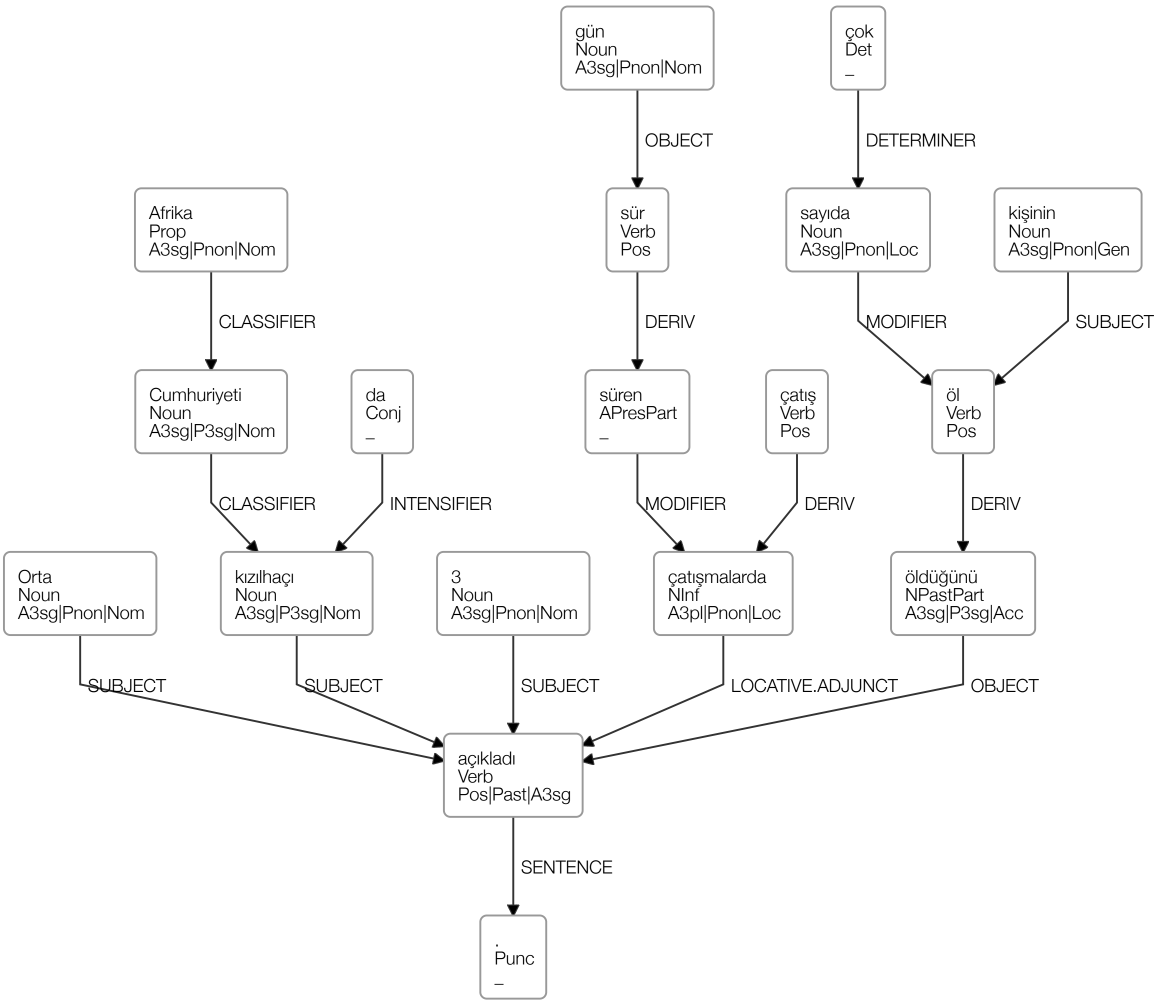

Figure 5.4. Dependency parse of the sentence *"Orta Afrika Cumhuriyeti kızılhaçı da 3 gün sür**en** çatışmalarda çok sayıda kişinin öldüğünü açıkladı." (The Red Cross of the Central African Republic also stated that numerous people died in the battles **which** continued for three days.)*

is an ungrammatical sentence:

Orta Afrika Cumhuriyeti kızılhaçı da 3 çatışmalarda çok sayıda kişinin öldüğünü açıkladı.

However, if the parser would gave us the correct parse, we would get this output, which is a grammatical and a well simplified sentence:

Orta Afrika Cumhuriyeti kızılhaçı da çatışmalarda çok sayıda kişinin öldüğünü açıkladı.

This sentence was one of the sentences that obtain poor scores from the human judges, just because of not removing the word "3". This tendency of the judges is acceptable, because the contained error directly destructs the grammar and the meaningfulness of the sentence. Actually this example is a good demonstration of why the text simplification task could be a fairly hard task. If the proposed system is not very accurate, the resultant text might not make sense because sentences could easily lose their readability and understandability because of these types of small errors.

Another problem is the system's inability to discriminate between restrictive and non-restrictive relative clauses. While applying RULE-1, our system also removes the restrictive relative clauses, since there is no a grammatical indicator in Turkish to detect the type of the clause. Restrictive relative clauses generally contain definitive information about the entities that they modify, and removing them may cause losing the semantic completeness of the sentences.

Lastly, we currently do not apply a tense resolution step for RULE-3 which splits sentences from *while* (represented with the *-ken* affix in Turkish). The example sentence that we gave while describing the rule is simplified as the following two sentences by our simplification system:

However, a grammatically correct simplification would resolve the tense of the

Müşerref Akıcı çarpmanın etkisiyle kaldırıma savrulurken.
Birol Şimşek otobüsün altında kaldı.

first sentence depending on the tense of the second sentence, and modify the word having the *-ken* affix like this:

*Müşerref Akıcı çarpmanın etkisiyle kaldırıma savrul**du**.*
*Birol Şimşek otobüsün altında kal**dı**.*

We currently do not focus on tense resolution strategies because our morphological analysis policies isolate this problem, while evaluating the system performance for the summaries.

6. EFFECTS OF TEXT SIMPLIFICATION ON TURKISH MDS

We also analyzed the effects of integrating our text simplification system to multi-document summarization for Turkish. This chapter gives the details about how we integrated our simplification system to the previously developed MDS system for Turkish. Additionally, the comparison of different strategies like applying simplification before or after summarization, and utilizing different combinations of simplification rules is presented.

6.1. Methodology

At first, we integrated our text simplification system as a preprocessing step to the summarization system. In this method, we give all documents in a cluster to the simplification system, and generate a simplified document for each document in the cluster. Then, these simplified documents are given to the summarization system, and a summary is generated for each cluster. We examined different combinations of the simplification rules for this approach as explained below:

- *SimpALL*: All three simplification rules are applied in the simplification step.
- *SimpSPLIT*: In this method, only rules applying a sentence disaggregation (i.e., splitting) policy, namely RULE-2 and RULE-3, are used.
- *SimpDROP*: This method uses only RULE-1 which is a drop policy that compresses sentences containing relative clauses.

Secondly, we investigated how applying simplification after the sentence selection step affects the summary performance. Our intuition is that we may include more sentences into the summary, if we can shorten the sentences that are selected by the summarization system, which can eventually result in more content coverage. For this approach, we increased the limit for the maximum number of words that can be included into the summary from 120 to 300, and generated longer summaries that contain more sentences. After the summaries are generated, we applied the simplification methods on the sentences in the summary, then took sentences one by one until the limit of 120 words is reached. The resulting summaries were given to ROUGE for evaluation.

6.2. Evaluation and Results

Table 6.1 and Table 6.2 list the ROUGE scores of different simplification policies for generating summaries using the Prefix8 and Raw policies, respectively. While SimpSPLIT performs similar to or better than summarizing without simplification, SimpALL and SimpDROP cause a noticeable decrease in the summary performance. The effect of SimpSPLIT is more obvious for the Raw policy, where SimpSPLIT increases the ROUGE scores for each metric. However, for the Prefix8 policy, it only has a little impact on the ROUGE-2 metric, while it does not change the scores for the other ROUGE metrics.

Table 6.3 presents the results for applying simplification as a preprocessing or postprocessing step. The results show that applying the SimpSPLIT method in the postprocessing step does not affect the performance as opposed to the case of applying it as a preprocessing step. Generally speaking, using simplification before summarization results in better scores than using it after summarization.

Table 6.1. ROUGE scores for different simplification policies when the summarization system is run with the Prefix8 stemming policy.

Policy	ROUGE-1	ROUGE-2	ROUGE-SU4
Prefix8 - SimpSPLIT	0.378	**0.193**	0.185
Prefix8	0.378	0.19	0.184
Prefix8 - SimpALL	0.35	0.163	0.16
Prefix8 - SimpDROP	0.351	0.163	0.16

Table 6.2. ROUGE scores for different simplification policies when the summarization system is run with the Raw stemming policy.

Policy	ROUGE-1	ROUGE-2	ROUGE-SU4
Raw - SimpSPLIT	**0.376**	**0.182**	**0.178**
Raw	0.364	0.177	0.172
Raw - SimpALL	0.347	0.163	0.16
Raw - SimpDROP	0.347	0.159	0.157

Table 6.3. ROUGE scores for applying the simplification policies before and after summarization. Results are for the Raw stemming policy.

Policy	ROUGE-1	ROUGE-2	ROUGE-SU4
Raw - SimpSPLIT - Before	0.376	0.182	0.178
Raw	0.364	0.177	0.172
Raw - SimpSPLIT - After	0.366	0.176	0.171
Raw - SimpDROP - Before	0.347	0.159	0.157
Raw - SimpDROP - After	0.340	0.149	0.148

6.3. Discussion

The detailed analysis of the different simplification strategies provided in the previous section revealed important results. Firstly, the results showed that using only the splitting rules improves performance, whereas using the drop rule causes a sharp decrease in the ROUGE scores. The decrease caused by the drop rule was actually an expected observation, because the current implementation might result in an over-simplification of the sentences. We always drop a relative clause if we match the corresponding pattern, and we do not distinguish between non-restrictive and restrictive relative clauses for Turkish. As we stated before, this is not a genius idea because we also always remove restrictive clauses which generally provide additional information that is important for the main content of the sentences. The other reason is that the current implementation does not consider any contextual information while removing the found relative clauses. However, it would be a good idea to remove a part of the sentence if it has been discussed before in the context of the generated summary.

Meanwhile, the performance increase obtained by applying the split rules is a promising result, since it demonstrates that simplifying complex sentences to individual simpler sentences could indeed increase summary quality. We should also note that applying the split rules before summarization is more effective. The reason behind this observation could be that these rules do not compress the information included in the sentences, rather they distribute them to more focused sentences. By this way, our summarization system could evaluate the simplified sentences individually, which eventually enables it to distinguish the important content included in one of the simplified sentences and to select it as a summary sentence.

Another observation is that using the split rules after the sentence selection process does not affect the system performance. This was also somewhat expected, since we remove one of the obtained simpler sentences in order to shorten the summary, but we do not apply any intelligent strategies to identify the best sentence to keep in the summary and to determine which of the sentences obtained by splitting should be dropped from the summary. On the other hand, when we give the splitted sentences to LexRank, the system can distinguish the important parts of the complex sentence, and select the related splitted sentences for the summary.

7. CONCLUSIONS

In this thesis, we developed the first multi-document summarization system for the Turkish language. A manually annotated data set has been constructed from scratch containing online news articles, and made publicly available for future studies. We utilized the LexRank summarization algorithm, and analyzed the effects of different stemming policies for Turkish MDS in detail. Our results show that simple fixed-length truncation methods with high thresholds (especially taking the first 8 letters) improves summarization scores. In contrast to our expectation, using morphological analysis does not enhance Turkish MDS, possibly due to the homogeneousness of the documents in a cluster to be summarized.

Another contribution of the thesis is that we implemented the first text simplification system for Turkish, a rule-based system that uses various syntactic features to simplify sentences. We evaluated the system using human judges to quantify the quality of the proposed simplifications. The empirical results show that the system generally proposes acceptable simplifications.

Lastly, we evaluated our text simplification system on the multi-document summarization domain via integrating into our existing MDS system. We compared the effects of applying text simplification before or after summarization. Different combinations of the simplification rules were also examined. The results show that applying simplification rules that split sentences could improve summary performance, especially if these rules are applied as a preprocessing step. In contrast, using the compression rule decreases the ROUGE scores. Our intuition for this result is that compression methodologies should be applied carefully in the summarization domain. Actually, they should consider contextual information, while deciding whether a candidate compression should be performed or not. Additionally, the inability of distinguishing the non-restrictive relative clauses from the restrictive ones, causes important clauses to be also dropped. Proposing methods to distinguish these different types of relative clauses would be another valuable contribution.

We believe that an accurate text simplification system could be a valuable tool also for the Machine Translation problem. Translating complex sentences to another language is harder than translating simpler sentences. Therefore, as future work, we plan to utilize our text simplification system on the Turkish-English machine translation problem with the aim of improving system performance via simplifying sentences that are hard to translate.

APPENDIX A: Guidelines for the Human Annotators and Evaluators

A.1. Annotation Guidelines for Multi-Document Summarization

Thank you for contributing to create this Turkish multi-document summary database.

A set of documents about a particular event or topic are given to you. Documents are gathered from news articles from different sources.

Please follow the below guidelines:

(i) You will create a short multi-document summary of the entire document set. A short summary should not be longer than 120 words. We will chop off words beyond the 120th, so please do not include more than 120 words.

(ii) Within the size limits, you should try to represent all the content of the document set to some degree.

(iii) Do not include your subjective opinions, rather summarize the documents in an objective manner.

(iv) Feel free to use your own words.

A.2. Evaluation Guidelines for the Text Simplification Human Judges

Bu çalışma, cümleleri otomatik olarak sadeleştirmeyi hedefleyen bir sistemin başarısının degerlendirilmesi amacıyla yapılmaktadır. İlgili Excel dosyasında, orijinal cümleler ve sistemin önerdiği sadeleştirmeler yer almaktadır.

Katılımcılardan beklenen bu cümleleri aşağıdaki 3 kriter açısından kıyaslamalarıdır:

- *Sadelik*: Sistemin önerdiği versiyon, orijinal versiyona göre ne kadar sade?
- *Dilbilgisi*: Sistemin önerdiği versiyon, dilbilgisi açısından ne kadar doğru?
- *Anlam*: Sistemin önerdiği versiyon, orijinal cümlenin anlatmak istediği ana temayı ne kadar içeriyor?

Değerlendirme, her cümlede ilgili kriterler için 1 den 5'e kadar bir sayı verilerek yapılacaktır. Rakamların anlamlarını şu şekilde düşünebilirsiniz:

1	Çok başarısız
2	Başarısız
3	Orta
4	Başarılı
5	Çok başarılı

Aşağıda iki adet örnek değerlendirme verilmiştir.

A.2.1. Örnek-1

Orijinal Cümle: E-postalara göre Ocak ayında planlanmaya başlayan Goliath için 500 bin dolarlık bir fon oluşturulması öngörüldü.

Sistem Çıktısı: E-postalara göre Goliath için 500 bin dolarlık bir fon oluşturulması öngörüldü.

Sadelik	Dilbilgisi	Anlam
5	4	4

A.2.2. Örnek-2

Orijinal Cümle: Zschaepe'nin başsanık olarak yargılandığı, geçen yıl mayıs ayında başlayan davanın, en erken 2016 yılında sonuçlanması bekleniyor.

Sistem Çıktısı: Zschaepe'nin davanın, en erken 2016 yılında sonuçlanması bekleniyor.

Sadelik	Dilbilgisi	Anlam
3	2	2

NOT: Yukarıdaki örnekler sadece değerlendirme şeklini göstermek için eklenmiştir. Değerlendirmeye dair bir yönlendirme amacı taşımamaktadır. Değerlendirmeyi kendinize göre doğru olan şekilde yapınız.

APPENDIX B: Sample Simplifications

B.1. Good Examples

ORIJINAL SENTENCE	SIMPLIFIED OUTPUT
Ortak açıklamanın geçerliliğini muhafaza ettiğini vurgulayan komisyon sözcüleri, gazetelere polis baskınlarının ve gazetecilerin tutuklanmasının **demokrasinin temel ilkelerinden olan** basın hürriyeti ile bağdaşmadığını kaydetti.	Komisyon sözcüleri, gazetelere polis baskınlarının ve gazetecilerin tutuklanmasının basın hürriyeti ile bağdaşmadığını kaydetti.
AB Komisyonu, **alınan** tavrın **üye ülkelerin ve Avrupa Parlamentosu'nun endişelerini yansıtan** ortak bir pozisyon olduğunu ve AB'nin kaygılarını net bir şekilde ifade ettiğini belirtti.	AB Komisyonu tavrın ortak bir pozisyon olduğunu ve AB'nin kaygılarını net bir şekilde ifade ettiğini belirtti.
Ülkenin en zengin isimlerinin bir araya toplandığı yemekte konuklara seslenen Putin, Rusya'daki ekonomik krizin ülkenin **öteden beri var olan** sorunlarını çözmek için bir fırsat olduğu mesajını verdi.	Putin, Rusya'daki ekonomik krizin ülkenin sorunlarını çözmek için bir fırsat olduğu mesajını verdi.
Ülkenin en turistik noktası olan başkent Paris'teki Eyfel Kulesi çevresinde komandolar devriye geziyor.	Başkent Paris'teki Eyfel Kulesi çevresinde komandolar devriye geziyor.
88 yaşındaki laik Es-Sibsi, **dün akşam sandıkların açılmasından kısa süre sonra kutlamalar yapmaya başlayan** taraftarlarına erken zaferini ilan ettiği hitabında, ülkeyi istikrara kavuşturma sözü verirken, Tunuslulara, kimseyi ayırmadan birlikte çalışmaları gerektiğini söyledi.	88 yaşındaki laik Es-Sibsi taraftarlarına erken zaferini ilan ettiği hitabında, ülkeyi istikrara kavuşturma sözü verirken. Tunuslulara, kimseyi ayırmadan birlikte çalışmaları gerektiğini söyledi.

B.2. Bad Examples

ORIJINAL SENTENCE	SIMPLIFIED OUTPUT
Ankara **Emniyet Müdürlüğü'ne girmek isteyen** vatandaşlar, uzun kuyruk oluşturdu.	Ankara vatandaşlar, uzun kuyruk oluşturdu.
Medvedev, Ukrayna para birimi **grivna ile birlikte rublenin bu yıl en kötü performans sergileyen** para birimi olduğunu söyledi.	Medvedev, Ukrayna para birimi para birimi olduğunu söyledi.
Putin, milyarderlerden petrol ve **doğalgaza bağlı olan** ekonominin çeşitlendirilmesi için işbirliği istedi.	Putin, milyarderlerden petrol ve ekonominin çeşitlendirilmesi için işbirliği istedi.

APPENDIX C: Sample Summaries

C.1. Good Example

Table C.1. This is a good summary output of our MDS system which receives an F_1-score of 0.428 in terms of the ROUGE-2 metric. The summary is generated using the Prefix8 stemming policy. No simplification policy is used.

Reference	Mardin'in Nusaybin İlçesi İpekyolu üzerindeki Yeniyol köyü yakınlarında dün gece 01:30 sıralarında Suriye sınırındaki mayınlı bölgeden kaçak olarak Türkiye'ye geçmeye çalışan 3 Suriyeli açılan ateş sonucu öldü. 3 kişi mayınlı alanı geçmek isterken görev değişimi yapmış Mardin'e giden Özel Harekat Timleri tarafından Nusaybin-Kızıltepe karayolunun 3. km'sinde fark edildi. İddiaya göre askerlerin dur ihtarına rağmen sınırı geçip İpekyolu'na çıkan ve Yeniyol köyüne doğru kaçmaya çalışan 3 kişiye askerler tarafından zırhlı araçtan ateş açıldı. 3 Suriyeli olay yerinde öldü. Konuyla ilgili Mardin Valiliği açıklama yaptı. Olayın ardından bölgeye Jandarma Olay Yeri İnceleme ekipleri ile Özel Harekat Polis Timleri sevk edildi. Ölen 3 kişinin cesetleri Malatya Adli Tıp Kurumu'na gönderildi. Olayla ilgili soruşturma sürüyor.
System	Mardin'in Nusaybin İlçesi'nde dün gece sınırdaki mayınlı bölgeden kaçak olarak Türkiye'ye geçmeye çalışan 3 Suriyeli Özel Harekat timlerinin dur ihtarına uymayınca açtığı ateş sonucu öldü. İddiaya göre Özel Harekat timinden yapılan dur ihtarına rağmen sınırı geçip İpekyolu'na çıkan ve Yeniyol Köyü'ne doğru kaçmaya çalışan 3 kişiye zırhlı araçtan ateş açıldı. 3 kişi Suriye'den sınırdaki mayınlı alanı kaçak olarak geçmek isterken askerler tarafından fark edildi. Mardin Valiliğinden olaya ilişkin yapılan açıklamada "Yasadışı yollardan sınırı geçmeye çalışan bir gruba görev değişimi maksadıyla Nusaybin'den il merkezine intikal etmekte olan Polis Özel Harekat timi tarafından Nusaybin-Kızıltepe yolunun 9. kilometresinde müdahale edilmiş olay esnasında Suriye uyruklu 3 şahıs hayatını kaybetmiştir." denildi. Olayla ilgili Nusaybin Cumhuriyet Başsavcılığınca adli soruşturma başlatıldığı belirtilen açıklamada öldürülen 3 Suriyelinin cesedinin Malatya Adli Tıp Kurumuna gönderildiği kaydedildi.

C.2. Bad Example

Table C.2. This is a bad summary output of our MDS system which receives an F_1-score of 0.050 in terms of the ROUGE-2 metric. The summary is generated using the Prefix8 stemming policy. No simplification policy is used.

Reference	Saatteki hızı 385 kilometreye ulaşan Haiyan tayfunundan Filipinler'de yaklaşık 4-5 milyon kişi etkilendi. 800 bin kişi evini terk etmek zorunda kaldı. 1.7 milyon çocuk etkilendi. Sadece Leyte bölgesinde en az 10 bin kişinin ölmüş olabileceği tahmin ediliyor. UNİCEF Filipinler'de gıda su barınak ve enerji ihtiyacı bulunduğunu dile getirdi. 60 tonluk acil ihtiyaç malzemesi Kopenhang limanından gemiyle yola çıkarıldı. Hollanda 2 milyon euro nakdi yardımda bulunacağını açıkladı. Açlık ve içme suyu eksikliği nedeniyle dükkanlar ve evler yağmalanmaya başlandı. Kızılhaç yetkilileri Filipinler'deki durumu kıyamet manzarası olarak değerlendiriyor. Yıkılan yollar köprüler ve havaalanları nedeniyle yardımlar havadan askeri kargo uçaklarıyla gerçekleştiriliyor. Arama kurtarma ekipleri salgın hastalıkları önlemek için sokaklarda yatan su birikintilerinde yüzen cesetleri toplamaya ve gömmeye çalışıyor Hızını kaybeden tayfun Vietnam'a doğru ilerliyor.
System	Kızılhaç'ın Filipinler'deki yetkilisi tayfun ardından ülkede bir kıyamet manzarası olduğunu söyledi. Binlerce kişinin yaşamını yitirdiğinin belirtildiği Filipinler'deki tayfun felaketine ilişkin açıklama yapan Birleşmiş Milletler Çocuklara Yardım Fonu UNICEF tayfundan etkilenen 4 milyon çocuğun acil yardıma muhtaç olduğunu belirtti. Filipinler'de günlerdir etkisini sürdüren Haiyan tayfunu büyük bir yıkıma yol açarken tayfundan dolayı en az 10 bin kişinin yaşamını yitirmiş olabileceği belirtiliyor. Filipinler'i Cuma günü vuran Haiyan tayfunu sebebiyle ölü sayısının 10 bine yükseldiği bildirildi. Yetkililer tayfundan en çok etkilenen Leyte eyaletinin başkenti Tacloban'da 10 binden fazla kişinin öldüğünü binlerce binanın yerle bir olduğunu heyelan ve devrilen ağaçlar nedeniyle yolların kapandığını havaalanında terminal binasının yıkıldığını açıkladı. Filipinler'e tarihinin en büyük doğal felaketlerinden birini yaşatarak binlerce kişinin ölümüne neden olan Haiyan tayfunu Vietnam'da tropik fırtına biçiminde karayı vurdu.

REFERENCES

1. Gupta, V. and G. S. Lehal, "A Survey of Text Summarization Extractive Techniques.", *Journal of Emerging Technologies in Web Intelligence*, Vol. 2, No. 3, 2010.

2. Speier, C., J. S. Valacich and I. Vessey, "The influence of task interruption on individual decision making: An information overload perspective", *Decision Sciences*, Vol. 30, No. 2, pp. 337–360, 1999.

3. Luhn, H. P., "The automatic creation of literature abstracts", *IBM Journal of research and development*, Vol. 2, No. 2, pp. 159–165, 1958.

4. Baxendale, P. B., "Machine-made index for technical literature: an experiment", *IBM Journal of Research and Development*, Vol. 2, No. 4, pp. 354–361, 1958.

5. Edmundson, H. P., "New methods in automatic extracting", *Journal of the ACM (JACM)*, Vol. 16, No. 2, pp. 264–285, 1969.

6. Giannakopoulos, G., M. El-Haj, B. Favre, M. Litvak, J. Steinberger and V. Varma, "TAC 2011 MultiLing pilot overview", , 2011.

7. Giannakopoulos, G., "Multi-document multilingual summarization and evaluation tracks in ACL 2013 MultiLing Workshop", *MultiLing 2013*, p. 20, 2013.

8. Eryiğit, G., J. Nivre and K. Oflazer, "Dependency parsing of Turkish", *Computational Linguistics*, Vol. 34, No. 3, pp. 357–389, 2008.

9. Hakkani-Tür, D. Z., K. Oflazer and G. Tür, "Statistical Morphological Disambiguation for Agglutinative Languages", *COLING*, pp. 285–291, Morgan Kaufmann, 2000.

10. Altan, Z., "A Turkish Automatic Text Summarization System", *Proceedings of the IASTED International Conference Artificial Intelligence and Applications*, pp. 74–83, 2004.

11. Çığır, C., M. Kutlu and İlyas Çiçekli, "Generic text summarization for Turkish", *ISCIS*, pp. 224–229, IEEE, 2009.

12. Özsoy, M. G., İlyas Çiçekli and F. N. Alpaslan, "Text Summarization of Turkish Texts using Latent Semantic Analysis", C.-R. Huang and D. Jurafsky (Editors), *COLING*, pp. 869–876, Tsinghua University Press, 2010.

13. Güran, A., E. Bekar and S. Akyokuş, "A comparison of feature and semantic-based summarization algorithms for Turkish", *International Symposium on Innovations in Intelligent Systems and Applications*, Citeseer, 2010.

14. Güran, A., N. Bayazıt and E. Bekar, "Automatic summarization of Turkish documents using non-negative matrix factorization", *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, pp. 480–484, IEEE, 2011.

15. Knight, K. and D. Marcu, "Statistics-Based Summarization - Step One: Sentence Compression", *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 703–710, AAAI Press, 2000, `http://dl.acm.org/citation.cfm?id=647288.721086`.

16. Siddharthan, A., A. Nenkova and K. McKeown, "Syntactic Simplification for Improving Content Selection in Multi-document Summarization", *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Association for Computational Linguistics, Stroudsburg, PA, USA, 2004, `http://dx.doi.org/10.3115/1220355.1220484`.

17. Vanderwende, L., H. Suzuki, C. Brockett and A. Nenkova, "Beyond SumBasic:

Task-focused Summarization with Sentence Simplification and Lexical Expansion", *Inf. Process. Manage.*, Vol. 43, No. 6, pp. 1606–1618, Nov. 2007, http://dx.doi.org/10.1016/j.ipm.2007.01.023.

18. Silveira, S. B. and A. Branco, "Combining a double clustering approach with sentence simplification to produce highly informative multi-document summaries.", C. Zhang, J. Joshi, E. Bertino and B. M. Thuraisingham (Editors), *IRI*, pp. 482–489, IEEE, 2012, http://dblp.uni-trier.de/db/conf/iri/iri2012.html#SilveiraB12.

19. Erkan, G. and D. R. Radev, "LexPageRank: Prestige in Multi-Document Text Summarization", *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain* [71], pp. 365–371, http://www.aclweb.org/anthology/W04-3247.

20. Kupiec, J., J. Pedersen and F. Chen, "A trainable document summarizer", *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 68–73, ACM, 1995.

21. Lin, C.-Y., "Training a selection function for extraction", *Proceedings of the eighth international conference on Information and knowledge management*, pp. 55–62, ACM, 1999.

22. Conroy, J. M. and D. P. O'leary, "Text summarization via hidden markov models", *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 406–407, ACM, 2001.

23. Barzilay, R. and M. Elhadad, "Using Lexical Chains for Text Summarization", *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pp. 10–17, 1997.

24. Miller, G. A., "WordNet: A Lexical Database for English", *Commun. ACM*,

Vol. 38, No. 11, pp. 39–41, 1995.

25. Marcu, D., *Sixth Workshop on Very Large Corpora*, chap. Improving summarization through rhetorical parsing tuning, 1998, http://aclweb.org/anthology/W98-1124.

26. Mann, W. C. and S. A. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization", *Text*, Vol. 8, No. 3, pp. 243–281, 1988.

27. McKeown, K. and D. R. Radev, "Generating Summaries of Multiple News Articles", *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pp. 74–82, ACM, New York, NY, USA, 1995.

28. McKeown, K. R., J. L. Klavans, V. Hatzivassiloglou, R. Barzilay and E. Eskin, "Towards Multidocument Summarization by Reformulation: Progress and Prospects", *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pp. 453–460, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1999.

29. Barzilay, R., K. R. McKeown and M. Elhadad, "Information Fusion in the Context of Multi-document Summarization", *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pp. 550–557, Association for Computational Linguistics, Stroudsburg, PA, USA, 1999, http://dx.doi.org/10.3115/1034678.1034760.

30. Collins, M. J., "A New Statistical Parser Based on Bigram Lexical Dependencies", *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pp. 184–191, Association for Computational Linguistics, Stroudsburg, PA, USA, 1996, http://dx.doi.org/10.3115/981863.981888.

31. Radev, D. R., H. Jing and M. Budzikowska, "Centroid-based summarization of

multiple documents: sentence extraction, utility-based evaluation, and user studies", *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pp. 21–30, Association for Computational Linguistics, 2000.

32. Mihalcea, R. and P. Tarau, "TextRank: Bringing Order into Text", *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain* [71], pp. 404–411, `http://www.aclweb.org/anthology/W04-3252`.

33. Page, L., S. Brin, R. Motwani and T. Winograd, "The PageRank citation ranking: Bringing order to the web.", Stanford InfoLab, 1999.

34. Mihalcea, R. and P. Tarau, "A language independent algorithm for single and multiple document summarization", *In Proceedings of IJCNLP'2005*, 2005.

35. Nenkova, A. and K. McKeown, "A survey of text summarization techniques", *Mining Text Data*, pp. 43–76, Springer, 2012.

36. Wong, K.-F., M. Wu and W. Li, "Extractive Summarization Using Supervised and Semi-supervised Learning", *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pp. 985–992, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008, `http://dl.acm.org/citation.cfm?id=1599081.1599205`.

37. Christensen, J., S. S. Mausam and O. Etzioni, "Towards Coherent Multi-Document Summarization", *Proceedings of NAACL-HLT*, pp. 1163–1173, 2013.

38. Gong, Y. and X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis", *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pp. 19–25, ACM, New York, NY, USA, 2001, `http://doi.acm.org/10.1145/383952.383955`.

39. Steinberger, J. and K. Ježek, "Text Summarization and Singular Value Decomposition", *Proceedings of the Third International Conference on Advances in Information Systems*, ADVIS'04, pp. 245–254, Springer-Verlag, Berlin, Heidelberg, 2004, http://dx.doi.org/10.1007/978-3-540-30198-1_25.

40. Giannakopoulos, G., "Multi-document multilingual summarization and evaluation tracks in ACL 2013 MultiLing Workshop", *MultiLing 2013*, p. 20, 2013.

41. Pembe, F. C. and T. Gungor, "Towards a new summarization approach for search engine results: An application for Turkish", *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, pp. 1–6, IEEE, 2008.

42. Can, F., S. Koçberber, E. Balçık, C. Kaynak, H. C. Öcalan and O. M. Vursavaş, "Information retrieval on Turkish texts", *Journal of the American Society for Information Science and Technology*, Vol. 59, No. 3, pp. 407–421, 2008.

43. Akkuş, B. K. and R. Çakıcı, "Categorization of Turkish News Documents with Morphological Analysis", *ACL (Student Research Workshop)*, pp. 1–8, The Association for Computer Linguistics, 2013.

44. Galiotou, E., N. Karanikolas and C. Tsoulloftas, "On the effect of stemming algorithms on extractive summarization: a case study", P. H. Ketikidis, K. G. Margaritis, I. P. Vlahavas, A. Chatzigeorgiou, G. Eleftherakis and I. Stamelos (Editors), *Panhellenic Conference on Informatics*, pp. 300–304, ACM, 2013.

45. Carroll, J., G. Minnen, D. Pearce, Y. Canning, S. Devlin and J. Tait, "Simplifying text for language-impaired readers", *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 269–270, 1999.

46. Thomas, S. R. and S. Anderson, "WordNet-based lexical simplification of a document", J. Jancsary (Editor), *Proceedings of KONVENS 2012*, pp. 80–88, ÖGAI, September 2012, http://www.oegai.at/konvens2012/proceedings/13_

thomas12o/, main track: oral presentations.

47. Biran, O., S. Brody and N. Elhadad, "Putting it Simply: a Context-Aware Approach to Lexical Simplification.", *ACL (Short Papers)*, pp. 496–501, The Association for Computer Linguistics, 2011, http://dblp.uni-trier.de/db/conf/acl/acl2011s.html#BiranBE11.

48. Kauchak, D., "Improving Text Simplification Language Modeling Using Unsimplified Text Data", *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1537–1546, Association for Computational Linguistics, Sofia, Bulgaria, August 2013, http://www.aclweb.org/anthology/P13-1151.

49. Brants, T. and A. Franz, "Web 1T 5-gram corpus version 1.1", *Linguistic Data Consortium*, 2006.

50. Chandrasekar, R. and B. Srinivas, "Automatic induction of rules for text simplification", *Knowledge-Based Systems*, Vol. 10, No. 3, pp. 183 – 190, 1997, http://www.sciencedirect.com/science/article/pii/S0950705197000294.

51. Siddharthan, A., "Syntactic simplification and text cohesion", *Research on Language and Computation*, Vol. 4, No. 1, pp. 77–109, 2006.

52. Jonnalagadda, S. and G. Gonzalez, "Sentence Simplification Aids Protein-Protein Interaction Extraction", *Proceedings of the 3rd International Symposium on Languages in Biology and Medicine (LBM'09)*, pp. 8–10, November 2009, http://www.public.asu.edu/~sjonnal3/home/papers/LBM2009.pdf, informal publication.

53. McClosky, D. and E. Charniak, "Self-training for Biomedical Parsing", *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pp. 101–104, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008,

http://dl.acm.org/citation.cfm?id=1557690.1557717.

54. Filippova, K. and M. Strube, “Dependency Tree Based Sentence Compression”, *Proceedings of the Fifth International Natural Language Generation Conference*, INLG ’08, pp. 25–32, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008, http://dl.acm.org/citation.cfm?id=1708322.1708329.

55. Briscoe, T., J. Carroll and R. Watson, “The Second Release of the RASP System”, *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL ’06, pp. 77–80, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, http://dx.doi.org/10.3115/1225403.1225423.

56. Klein, D. and C. D. Manning, “Accurate Unlexicalized Parsing”, *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL ’03, pp. 423–430, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, http://dx.doi.org/10.3115/1075096.1075150.

57. Sleator, D. D. and D. Temperley, “Parsing English with a Link Grammar”, *CoRR*, Vol. abs/cmp-lg/9508004, 1995, http://arxiv.org/abs/cmp-lg/9508004.

58. Zhu, Z., D. Bernhard and I. Gurevych, “A Monolingual Tree-based Translation Model for Sentence Simplification”, *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING ’10, pp. 1353–1361, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, http://dl.acm.org/citation.cfm?id=1873781.1873933.

59. Wubben, S., A. van den Bosch and E. Krahmer, “Sentence Simplification by Monolingual Machine Translation”, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL ’12, pp. 1015–1024, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, http://dl.acm.org/citation.cfm?id=2390524.2390660.

60. Lewis, G. L., *Turkish Grammar (2nd ed)*, Oxford, England: Oxford University

Press, 2000.

61. Oflazer, K., “Two-level description of Turkish morphology”, *Literary and linguistic computing*, Vol. 9, No. 2, pp. 137–148, 1994.

62. Sak, H., T. Güngör and M. Saraçlar, “Morphological Disambiguation of Turkish Text with Perceptron Algorithm”, A. F. Gelbukh (Editor), *CICLing*, Vol. 4394 of *Lecture Notes in Computer Science*, pp. 107–118, Springer, 2007.

63. Radev, D., T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu *et al.*, “MEAD-a platform for multidocument multilingual text summarization”, Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), 2004.

64. Lin, C.-Y. and E. H. Hovy, “Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics”, *HLT-NAACL*, 2003.

65. Owczarzak, K., J. M. Conroy, H. T. Dang and A. Nenkova, “An assessment of the accuracy of automatic evaluation in summarization”, *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pp. 1–9, Association for Computational Linguistics, 2012.

66. Eryiğit, G., J. Nivre and K. Oflazer, “Dependency parsing of turkish”, *Computational Linguistics*, Vol. 34, No. 3, pp. 357–389, 2008.

67. Huddleston, R., G. K. Pullum *et al.*, “The Cambridge Grammar of English”, *Language. Cambridge: Cambridge University Press*, pp. 1–23, 2002.

68. BALPINAR, Z., *Turkish phonology, morphology and syntax*, Anadolu Universitesi, 2011.

69. Nivre, J., J. Hall and J. Nilsson, “MaltParser: a data-driven parser-generator for dependency parsing”, *Proceedings of LREC-2006*, 2006.

70. Woodsend, K. and M. Lapata, "Learning to Simplify Sentences with Quasi-synchronous Grammar and Integer Programming", *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 409–420, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, `http://dl.acm.org/citation.cfm?id=2145432.2145480`.

71. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, ACL, 2004.