RULE-BASED TEXT SUMMARIZATION
IN TURKISH

by
Çağdaş Can BİRANT

April, 2015
İZMİR
We have read the thesis entitled "RULE-BASED TEXT SUMMARIZATION IN TURKISH" completed by ÇAĞDAŞ CAN BİRANT under supervision of PROF. DR. YALÇIN ÇEBİ and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Yalçın ÇEBİ

Supervisor

Prof. Dr. Alp KUT

Thesis Committee Member

Prof. Dr. Gürer GÜLŞEVİN

Thesis Committee Member

Assoc. Prof. Dr. Bekir Taner DİNÇER

Examining Committee Member

Asst. Prof. Dr. Şen ÇAKIR

Examining Committee Member

Prof. Dr. Ayşe OKUR
Director
Graduate School of Natural and Applied Sciences
ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Prof. Dr. Yalçın ÇEBİ for his patience and support during the writing process of this thesis; my thesis tracking committee members Prof. Dr. Alp KUT and Prof. Dr. Gürer GÜLSEVİN for their valuable comments and guidance in the tracking meetings.

I would also like to express my gratitude to my colleagues Asst. Prof. Dr. Özlem AKTAŞ and Asst. Prof. Dr. Özgün KOŞANER for giving countenance to me from the very beginning of my studies and having courage to follow this track with me as a sister and brother. I would like to thank Asst. Prof. Dr. S. Zafer DİCLE, Inst. İ. Serat KORSACILAR and Nezih ÇETİNSÖZ and all members of DEÜ Bureau of Informatics, and A. Ferit ÖZAY, Emrah GEZER and Evren AKALP. I would also like to thank to Turkish Linguistic Association (Türk Dil Kurumu, TDK) for their support during the development of the resources for this study. This research was supported under Contract Number: 2011.KB.FEN.030 Dokuz Eylül University Scientific Research Projects Coordination Unit.

Last but not the least, I have my deepest and sincerest thanks to my mother Gönül DENİZERİ, who has been a source of encouragement and inspiration to me through all my life, and Asst. Prof. Dr. K. Ulaş BİRANT for his endlesss help in my life.

Çağdaş Can BİRANT
RULE-BASED TEXT SUMMARIZATION IN TURKISH

ABSTRACT

The volume of data produced has exponentially increased with the digital revolution and it continues to race to the limits of the capacity of our computers and supercomputers. Automatic text summarization is one of efforts to tame the bestial product of our daily data production. In order to understand what a text is about, a summary is needed which is short enough not to compromise the understandability, and comprehensive to include the most important topics of that text. Numerous automatic text summarization software which aimed at achieving this goal use semantic relations, thesauri, and word frequency lists. Our aim is to develop a Rule Based Automatic Text Summarization Software (RB-TTS) for Turkish.

In this thesis, semantic relations dictionaries are developed to be used in an automatic text summarization software for Turkish, and also a series of natural language processing tools developed before but improved in this thesis is used. In addition, new software that use new methodology in automatic text summarization are devised. During the studies carried out for this thesis, both dictionaries as byproducts; the Synonymous Dictionary and the Antonymous Dictionary were developed, each of which was approved and electronically published by Turkish Language Association (Türk Dil Kurumu, TDK). With the help of the mentioned tools and dictionaries, RB-TTS was developed. The average success rate of the RB-TTS is analysed both quantitatively using ROUGE-N metrics and qualitatively. There is shown that, results of this analysis is close to the results of other works, which are explained in literature. The similarity between the summaries of authors and our software is also shown by people who participate to analysis work.

Keywords: Turkish, summarization, natural language processing, rule based, dictionary.
TÜRKÇE İÇİN KURAL TABANLI METİN ÖZETLEME

ÖZ


Anahtar kelimeler: Türkçe, özetleme, doğal dil işleme, kural tabanlı, sözlük.
# CONTENTS

<table>
<thead>
<tr>
<th>Chapter Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ph.D. THESIS EXAMINATION RESULT FORM</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>ÖZ</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xi</td>
</tr>
<tr>
<td>CHAPTER ONE - INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER TWO - LITERATURE REVIEW</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>4</td>
</tr>
<tr>
<td>2.2 The need for summaries</td>
<td>4</td>
</tr>
<tr>
<td>2.3 A summary or abstract</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Automatic Text Summarization Definitions</td>
<td>12</td>
</tr>
<tr>
<td>2.5 A Brief History of Automatic Text Summarization</td>
<td>17</td>
</tr>
<tr>
<td>2.6 Automatic Text Summarization Studies in Turkish</td>
<td>25</td>
</tr>
<tr>
<td>CHAPTER THREE - TEXT SUMMARIZATON METHODOLOGY</td>
<td>29</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>29</td>
</tr>
<tr>
<td>3.2 Pre-processing in Automatic Text Summarization</td>
<td>30</td>
</tr>
<tr>
<td>3.2.1 Splitting</td>
<td>31</td>
</tr>
<tr>
<td>3.2.2 Tokens</td>
<td>32</td>
</tr>
<tr>
<td>3.2.2.1 Tokenization</td>
<td>32</td>
</tr>
<tr>
<td>3.2.2.2 Stop Word Identification</td>
<td>33</td>
</tr>
<tr>
<td>3.2.2.3 Stemming and Lemmatization</td>
<td>33</td>
</tr>
<tr>
<td>3.2.2.4 Capitalization or Case-Folding</td>
<td>34</td>
</tr>
<tr>
<td>3.2.3 Annotation</td>
<td>35</td>
</tr>
</tbody>
</table>
3.2.3.1 Morphological Parsing .......................................................... 35
3.2.3.2 Part-of-Speech (POS) Tagging ........................................... 36
3.2.3.3 Named Entity Tagging ......................................................... 36

3.3 Processing in Automatic Text Summarization .................................. 37
3.3.1 Automatic Text Summarization Models .................................... 37
3.3.1.1 Boolean Model ................................................................. 37
3.3.1.2 Statistical Models ............................................................ 38
  3.3.1.2.1 The Vector Space Model ........................................... 38
  3.3.1.2.2 The Probabilistic Model ......................................... 39
3.3.1.3 Linguistic and Knowledge-Based Model ............................... 39
3.3.1.4 Graph-Based Model ......................................................... 40

3.3.2 Automatic Text Summarization Algorithms ............................... 41
3.3.2.1 Summarization by Extraction ....................................... 42
  3.3.2.1.1 Surface Level Approaches ....................................... 42
  3.3.2.1.2 Intermediate-Level Approaches ............................... 43
  3.3.2.1.3 Deep Parsing Approaches ................................... 43
3.3.2.2 Summarization by Abstraction ..................................... 44
3.3.2.3 Summarization by Sentence Compression ............................. 45
  3.3.2.3.1 Sentence Compression ........................................... 45
  3.3.2.3.2 Multisentence Compression ................................... 46
  3.3.2.3.3 Symbolic Approaches to Sentence Compression ............ 46
  3.3.2.3.4 Statistical Approaches to Sentence Compression .......... 47
  3.3.2.3.5 Statistical-Linguistic Approaches to Sentence Compression ... 47

3.4 Post-processing in Automatic Text Summarization .......................... 47
3.4.1 Paraphrasing ........................................................................ 48
3.4.2 Sentence Fusion ..................................................................... 48
3.4.3 Sentence Enhancement .............................................................................................. 48

3.5. Conclusion .................................................................................................................. 49

CHAPTER FOUR - PREPROCESSING PHASE OF TURKISH RULE-BASED TEXT SUMMARIZER ............................................................................................................ 51

4.1 Introduction .................................................................................................................. 51

4.2 Turkish from a Typological Perspective ................................................................. 51

   4.2.1 Ortography .............................................................................................................. 51

   4.2.2. Phonology ............................................................................................................. 52

   4.2.3 Morphology ............................................................................................................ 53

   4.2.4 Syntax ..................................................................................................................... 54

4.3 Rule-Based Text Summarization Algorithm for Turkish Language ............ 55

   4.3.1 Pre-Processing for Rule-Based Text Summarization in Turkish ............ 56

       4.3.1.1 Sentence Boundary Detection Algorithm .................................................. 57

       4.3.1.2 Tokenizer ........................................................................................................ 58

       4.3.1.3 Stop Word Elimination ................................................................................. 58

       4.3.1.4 Morphological Parser ................................................................................... 59

CHAPTER FIVE - PROCESSING FOR RULE BASED TEXT SUMMARIZATION IN TURKISH ................................................................. 64

5.1 Introduction .................................................................................................................. 64

5.2 Post-Processing for Rule-Based Text Summarization in Turkish ............... 67

5.3 The User Interface ....................................................................................................... 67

       5.3.1 Adding Text Files .............................................................................................. 68

       5.3.2 Entering the Title of the Text ............................................................................ 68

       5.3.3 Entering Keywords ......................................................................................... 69

       5.3.4 Raw and Processed Text Data ........................................................................ 69

       5.3.5 Sentence Ranking .............................................................................................. 70
5.3.6 Summary Output...................................................................................... 70

CHAPTER SIX - EVALUATION ........................................................................ 72

6.1 Introduction ................................................................................................. 72
6.2 ROUGE-N: Automatic Evaluation of Summaries ....................................... 72
6.3 Human Evaluation ....................................................................................... 73
6.4 Data Set ....................................................................................................... 75
6.5 ROUGE Score of the Automatic Text Summarization Algorithm .......... 75
6.6 Human evaluation of the Automatic Text Summarization Algorithm ....... 77

CHAPTER SEVEN - CONCLUSION ................................................................. 80

REFERENCES................................................................................................. 83

APPENDICES ................................................................................................. 97
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Cognitive processes in human summarization</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Automatic summarization process</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>A Timeline of Automatic Text Summarization</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>The general algorithm of automatic text summarization</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>A portion of stop word list by Can et al., 2008</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Graph-based representation of a document</td>
<td>40</td>
</tr>
<tr>
<td>3.4</td>
<td>Phases of automatic text summarization</td>
<td>50</td>
</tr>
<tr>
<td>4.1</td>
<td>Main scheme for RBSDM</td>
<td>57</td>
</tr>
<tr>
<td>4.2</td>
<td>Morphological parser algorithm</td>
<td>63</td>
</tr>
<tr>
<td>5.1</td>
<td>Automatic text summarization algorithm for Turkish</td>
<td>66</td>
</tr>
<tr>
<td>5.2</td>
<td>The GUI of the automatic text summarization software</td>
<td>67</td>
</tr>
<tr>
<td>5.3</td>
<td>A screenshot of a sample text file</td>
<td>68</td>
</tr>
<tr>
<td>5.4</td>
<td>A screenshot of the GUI after title input</td>
<td>68</td>
</tr>
<tr>
<td>5.5</td>
<td>A screenshot of the GUI after keyword input</td>
<td>69</td>
</tr>
<tr>
<td>5.6</td>
<td>A screenshot of the GUI showing raw and processed text data</td>
<td>69</td>
</tr>
<tr>
<td>5.7</td>
<td>A screenshot of the GUI showing sentence rankings</td>
<td>70</td>
</tr>
<tr>
<td>5.8</td>
<td>Output of the software, sentences not ordered in terms of their original position in the text</td>
<td>71</td>
</tr>
<tr>
<td>5.9</td>
<td>Output of the software, sentences ordered in terms of their original position in the text</td>
<td>71</td>
</tr>
<tr>
<td>5.10</td>
<td>Reference summary, the summary written by the author of the text</td>
<td>71</td>
</tr>
<tr>
<td>6.1</td>
<td>Survey for human evaluation of summaries</td>
<td>74</td>
</tr>
<tr>
<td>6.2</td>
<td>Statements in the automatic text summarization evaluation survey</td>
<td>74</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Major representation methods and their relation to levels of linguistic structure</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>Phonemes and graphemes of Turkish</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Some portion of Turkish stop word list</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Morphemes, tags and explanations</td>
<td>60</td>
</tr>
<tr>
<td>4.4</td>
<td>Morphological parser output</td>
<td>60</td>
</tr>
<tr>
<td>5.1</td>
<td>Some of the meta-discourse markers</td>
<td>64</td>
</tr>
<tr>
<td>6.1</td>
<td>Information about the data set used in evaluation</td>
<td>75</td>
</tr>
<tr>
<td>6.2</td>
<td>ROUGE-N metrics for text1</td>
<td>76</td>
</tr>
<tr>
<td>6.3</td>
<td>ROUGE-N metrics for text2</td>
<td>76</td>
</tr>
<tr>
<td>6.4</td>
<td>ROUGE-N metrics for text3</td>
<td>76</td>
</tr>
<tr>
<td>6.6</td>
<td>Average Scores for the statements in the survey form</td>
<td>79</td>
</tr>
</tbody>
</table>
CHAPTER ONE
INTRODUCTION

1980’s has witnessed first the proliferation of personal computers (PCs), and 1990’s, the propagation and the public access of internet. This rapid computerization and spread of online sources boosted the production of data and the number of the documents created each second.

The volume of data produced has exponentially increased and it continues to race to the limits of the capacity of computers and supercomputers. The increase is, now, at a point that humanity is drowning in a data flood; any information can be easily accessed but humanity is not well informed. This justifies the expression “too much information kills information”.

Researchers, most probably with a motivation from trade-related specialists, try to find a way to process this unstructured, yet extremely valuable data. Natural Language Processing, Information Extraction and Information Retrieval are the fruits of the attempts to structure the unstructured data. Automatic text summarization is one of efforts to tame the bestial product of the daily data production, which have generated the 90% of the data ever produced by humans, in the last two years.

Automatic text summarization studies have come a long way, yet there is more to cover. Some languages are more advantageous in this respect. For instance, English, due to being the lingua franca of today’s world, is fathoms ahead of Turkish. The natural language processing studies and automatic text summarization studies in particular, have arrived very late to Turkish, similar to the late arrival printing press. However, researchers have made achievements worthy of commendation. This thesis is another humble attempt to tackle the problem of mass-production of data.

In this respect this thesis aims at developing a rule-based automatic text summarization system for Turkish. It is a bar set very high. Turkish, as an agglutinating language poses numerous intricacies for linguists, and it poses even more for computer
scientists. The sparseness of the linguistic resources is another drawback. This drawback will be tried to be overcome by developing its resources from scratch, and by improving the resources that are present in the literature. Different approaches from different areas of research will tried to be combined to achieve the goal we set before us.

Keeping our goal in mind, we have organized our thesis by dividing it into five chapters, apart from this brief Introduction and the Conclusion part. Chapter Two asks why we need summaries, and defines what a summary is; what an abstract is; what the difference between them are; what automatic text summarization is. It also gives a brief history of automatic text summarization both for international literature, and for Turkish.

Chapter Three explains in detail the methodology used in automatic text summarization, elaborating its phases. The chapter begins with the pre-processing phase and gives an account of approaches, methods and techniques used. Later, the chapter continues with the processing phase, expounding the algorithms used in automatic text summarization. In the last sections, the chapter summarizes the post-processing phase of automatic text summarization.

Chapter Four gives detailed typological information on Turkish and continues with the details of the preprocessing system developed in this thesis, according to our goal defined here. The chapter elaborates on the resources either developed or improved here and gives a detailed presentation of the rule-based automatic text summarization algorithm.

Chapter Five provides the detailed information about the automatic text summarization algorithm for Turkish developed in this thesis and its graphical user interface.
Chapter Six conducts analyses to evaluate the performance of the algorithm. In this chapter both quantitative and qualitative evaluations are considered. Two distinct methods are used to evaluate the summaries generated by the algorithm.

In the Conclusion chapter, we review our results and make some remarks about the advantageous aspects of our study. We conclude the last chapter with recommendations for future studies.
CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

With the commercialization and the public access of internet in the 1990’s, there had been a surge in the number and size of the documents created each second. However, as Torres-Moreno (2014) indicates, the number of the documents which have been annotated with any mark-up language still remains small when compared to unstructured text documents. As data transforms into BigData (Zikopoulos, Eaton, deRoos, Deutsch & Lapis, 2012) in terms of volume, variety and velocity, the volume of unstructured documents increases exponentially affirming the expression “too much information kills information” (Torres-Moreno, 2014). Even worse text documents are often analyzed in a perfunctory and very superficial way (Torres-Moreno, 2014) and the means to categorize and organize these texts are far from being perfect. Also, each text genre requires different approaches to be processed. This renders the implementation of text analysis, text mining and information extraction a difficult task (Torres-Moreno, 2014). In the context of this issue this chapter will first give a brief explanation on why summaries are needed. Following the justification of the need for a summary, the study will define what a summary or abstract is, then define what automatic summarization is and later will give a detailed literature review for automatic text summarization first in general and then specific to Turkish.

2.2 The Need for Summaries

Humans have a limited capacity of processing data and thus are incapable of handling the vast amount of data created each day. In order to get the gist of any text one has to read the whole text. This may seem an easy task for short texts or even some longer texts however it would be impossible for a human to read and process all texts in the world, even impossible to read texts in one particular genre. Also documents in different languages multiply this difficulty. And if someone is reading to perform a
task, not just for his or her own leisure activity, reading becomes a time and resource consuming task. Therefore, one needs something that would help him or her to process large amounts of text in a more efficient way. At this point, summaries or abstracts step in. As the National Information Standards Organization (NISO) states, “[a] well-prepared abstract enables readers to identify the basic content of a document quickly, to determine its relevance to their interests, and thus to decide whether they need to read the document in its entirety. The abstract may provide an introductory overview of its topic or argument for readers to whom the document is of marginal interest, and make a reading of the full document unnecessary” (ANSI/NISO, 1997). Also summaries as short as 17% of full text length sped up decision-making by almost a factor of 2 with no statistically significant degradation in accuracy (Mani, et al., 2002).

Summaries may be written by the author of the documents, professional summarizers or a third party (Torres-Moreno, 2014). Some of the documents, especially scientific articles often come along with summaries written by the author, or some professional summarizes these documents for the scientific journals, etc. Sometimes companies hire staff to summarize large amount of documents to get some job done. However, although some of the texts, especially scientific texts, come along with summaries, this is not possible for all texts. Also summaries provided by professional summarizers or third parties are not sometimes welcome since the cost of production of a summary by a professional is very high and the reliability is controversial (Torres-Moreno, 2014), because the preparation of abstracts requires an intellectual effort and a general familiarity with the subject (Luhn, 1958). In addition to this, the summaries are almost always influenced by the summarizer’s background, attitude or disposition (Luhn, 1958).

These facts put forth several valid reasons in favor of the – automatic – summarization of documents (Torres-Moreno, 2014):

i) Summaries reduce reading time.

ii) When researching documents, summaries make the selection process easier.

iii) Automatic summarization improves the effectiveness of indexing.
iv) Automatic summarization algorithms are less biased than human summarizers.

v) Personalized summaries are useful in question-answering systems as they provide personalized information.

vi) Using automatic or semi-automatic summarization systems increase the number of texts processed.

2.3 A Summary or Abstract

Before giving numerous definitions for the concepts of summary and abstract it should be noted that the notions summary and abstract are frequently used interchangeably, although the ANSI/NISO Guidelines for Abstracts (ANSI, 1979) differentiates these two and dictates that they should not be confused. The point in this standard is that it defines summary (DEFINITION 2.4b below) as a brief restatement usually at the end of a document which intended to complete the orientation of the reader. Moens (2002) draws attention to the distinction between an abstract and a summary and argues that an abstract refers to a stand-alone document surrogate, while a summary is an inherent part of the document. However as Koltay (2010) points out, this type of summary is often called concluding summary a subcategory of the summary domain. Therefore it should be kept in mind, in this thesis, that summary and abstract denote an umbrella term which covers both author abstracts and abstracts whose authors differ from the authors of the original texts, such as professional summarizers or a third party, following Koltay (2010).

The dictionaries and the literature provide numerous definitions of summary or abstract. The Meriam-Webster Online Dictionary gives the following definitions for summary (Meriam-Webster, 2014):

DEFINITION 2.1a [a summary is a] a brief statement that gives the most important information about something.

DEFINITION 2.1b [a summary is a] an abstract, abridgment, or compendium especially of a preceding discourse.

Meriam-Webster Online Dictionary gives the following definition for abstract (Meriam-Webster, 2014):
DEFINITON 2.2 [an abstract is] a brief written statement of the main points or facts in a longer report, speech, etc.

The American National Standards Institute (ANSI) provides a comprehensive definition for a summary (ANSI, 1979):

DEFINITION 2.3 [an abstract] is an abbreviated, accurate representation of the contents of a document, preferably prepared by its author(s) for publication with it.

Later the American National Standards Institute (ANSI) and the National Information Standards Organization (NISO) provide short definitions for abstract and summary (ANSI/NISO, 1997):

DEFINITION 2.4a [an abstract is] a brief and objective representation of a document or an oral presentation.

DEFINITION 2.4b [a summary is] a brief restatement within a document (usually at the end) of its salient findings and conclusions intended to complete the orientation of a reader who has studied the preceding text.

The International Organization for Standardization (ISO) defines an abstract as below (ISO, 1976)

DEFINITION 2.5 [an abstract] signifies an abbreviated, accurate representation of the contents of a document, without added interpretation or criticism and without distinction as to who wrote the abstract.

Endres-Niggemeyer (1998) gives the definition below:

DEFINITION 2.6 [an abstract is a text produced as a result of] the skilled reduction of an information object to its most important points.

Kilborn (1998) gives the following definition for an abstract:

DEFINITION 2.7 an abstract is a condensed version of a longer piece of writing that highlights the major points covered, concisely describes the content and scope of the writing, and reviews the writing’s contents in abbreviated form.

Koltay (2010) develops further and arrives at the below definitions:

DEFINITION 2.8a an abstract is a text that contains the most important content of an already existing text in a concise, condensed and abbreviated form.
DEFINITION 2.8b an abstract is a text that reflects the most important information of an existing (primary) text in a form shorter than the original.

Moens (2002) defines an abstract and a summary as follows:
DEFINITION 2.9a the abstract usually has the form of a continuous, coherent text or of a profile that structures certain information of the original text.
DEFINITION 2.9b a summary is a condensed derivative of the source text. A summary is concerned about content information and its expression.

The definitions above can be combined and the following definitions for summary and abstract can be produced:
DEFINITION 2.10a a summary is a brief and condensed statement that gives (usually at the end) the most important information, salient findings and conclusions of a preceding text intended to complete the orientation of a reader who has studied it.
DEFINITION 2.10b an abstract is a brief, condensed, accurate, objective, continuous and coherent representation of the contents of a document, without added interpretation or criticism, highlighting the scope and the major points covered.

As mentioned in the definitions above, there is a slight distinction between an abstract and a summary; a summary often being an integral part of a text frequently placed at the end of the document. However, as it has been indicated before abstract and summary are umbrella terms and they may be subcategorized into different kinds of abstracts and summaries. Also, although an abstract appears at the beginning of a document, it is and should be written last since it gives a brief representation of the whole text. Therefore, any distinction between an abstract and a summary cannot be seen in terms of their position in the text, because they are both written at the end of the writing process to give a glimpse of the document. Also, this thesis is about automatic text summarization; although the product of this process may be called as abstract or summary, the process itself is called summarization, so the distinction in definitions would not pose a problem for the automatic text summarization.

However, attention should be paid that the definitions which has been given up to this point describe abstracts or summaries produced the author, a professional
summarizer or a third party, i.e. people. Thus, definitions of automatic summaries or automatic text summarization should be provided. This will be done in Section 2.4, and then a brief history of automatic text summarization will be presented.

Before proceeding to explain what automatic summarization is, the cognitive processes underlying summarization in humans should be discussed.

In order to explain the summarization process, one should explain what text kinds of cognitive processes underlie text understanding and production. Kintsch & van Dijk (1978) give a detailed account of text understanding and production, assuming the surface structure of a discourse can be interpreted as a set of propositions which are ordered by various semantic relations among them. This semantic structure of discourse is characterized at two levels:

i) microstructure: the structure of the individual propositions and the relations among them,

ii) macrostructure: the structure of the discourse as a whole, on a more global scale.

Kintsch & van Dijk (1978) propose macrostructure since the propositions of the text (text base in their terms) must not only be connected among themselves, they must be connected to the topic of discourse to fulfil the constraint that established a meaningful whole.

The mapping between these levels is realized via a set of semantic mapping rules, the macrorules. Lemaire, Mandin, Dessus & Denhiere (2005) treat macrorules as the core of the cognitive processes involved in the summarization activity. Kintsch & van Dijk (1978) defined the function of macrorules as reducing and organizing the more detailed information of the microstructure of the text, by describing the same facts but from a more global point of view. They postulated three macrorules:

- **deletion**: deletes any proposition which is not directly or indirectly related to a subsequent proposition,
- **generalization**: substitutes any sequence of propositions by a general proposition denoting an immediate superset
• **construction**: substitutes any sequence of propositions by a proposition denoting a global fact. This global fact is represented in the microstructure of the discourse as propositions denoting normal conditions, components or consequences of this global fact.

Lemaire, Mandin, Dessus & Denhire (2005) design three additional macrorules to describe the operations on the source text in more detail:

• **paraphrase**: writes down a semantically similar sentence,
• **copy**: copies the resulting sentence almost exactly
• **off-the-subject**: adds a sentence without being related to the subject.

Johnson (1983) argues that at least six processes are involved in the summarization process:

i. **comprehending the individual propositions of a text**: unless the individual elements of the text are understood, there is no basis for establishing relation between them;

ii. **establishing connections between propositions**: Causal and temporal relations between propositions of the text provide cohesion and distinguish connected discourse from a string of unrelated sentences;

iii. **identifying the constituent structure of the text**: the knowledge about the constituent structure of a text influences the macrostructure. This macrostructure is constrained by a schema of the text. “*The schema of the text determines which micropropositions or generalizations of micropropositions are relevant and, thus, which parts of the text will form its gist*” (Kintsch & van Dijk, 1978);

iv. **remembering the information in the text**: summaries are based on [macro]propositions that are automatically generated during comprehension [the first process]. The reader transforms micropropositions in the text into macropropositions via macrorules and remembers only these macropropositions.

v. **selecting the information to be represented in the summary**: the constituents of the text and the relations between them are represented in the memory in a
(tree) structure. Most attempts to characterize the importance of units within a text have been based on the notion that more important units either have more connection with other units or occur at higher levels in a hierarchical representation of the organization of the text.

vi. *formulating a concise and coherent verbal representation of that information*: the constraint of being concise or the fact that most or all of the information in the [text] requires the summarizer to perform transformations within the subset of information that has been identified as important [in the previous process].

Johnson (1983) provides the transformations of deletion and replacement. We argue that this process may involve the macrorules six defined above.

Hidi & Anderson (1986) argue that the discussion above have some points in common; that is, both of them lay down a selection process in which the summarizer consciously evaluates the information, deletes unnecessary segments of the text and selects other segments for inclusion in the summary. Than the selected segments are condensed by substituting them with higher level, more general concepts. Both of these models accept that a concise representation of the text requires complex cognitive operations on the original propositions (Hidi & Anderson, 1986).

Hidi & Anderson claim that these cognitive processes or operations are influenced by three factors (1986):

i. *characteristics of the original (target) material*: the quality of a summary depends on (in addition to one’s ability to write) the characteristics of the original text to be summarized, namely the length, the genre and the complexity.

ii. *presence or absence of the target material*: the summarizer’s access to the text to be summarized influences the performance. When the summarizer have continuous access to the original material, the memory load for remembering details in the text is reduced and these freed memory could be used for secondary cognitive operations such as making fine discriminations, clarifying inconsistencies and chunking larger text units. However, when the original text is removed, the summarizer has to depend his own memory to complete the summary and he would not enough cognitive resource for performing secondary cognitive processes.
iii. **type of summary**: the summaries can be categorized into two groups as writer-based summaries, which are produced for the summarizer himself, and reader-based summaries, which are produced for the benefit of an audience. Writer-based summaries are likely to be constructed as smaller text units such as paragraphs and the writer is not expected to worry too much about the grammatical rules and strict space limitations. On the other hand, in reader-based summaries, the summarizer would have to worry about grammatical construction and cohesion, as well as frequently conform to space limitations and formal constraints.

Torres-Moreno asserts that summarization requires a significant cognitive effort for selection, reformulation and creation processes to create a coherent text containing the most informative segments of a document (2014).

All of the studies summarized above postulate similar processes for summarization that may be illustrated as in Figure 2.1:

![Figure 2.1 Cognitive processes in human summarization](image)

**2.4 Automatic Text Summarization Definitions**
In Section 1.3 the definitions of summary and abstract have been given; however, these definitions described summaries produced by humans. In this section the definitions of automatic text summarization will be provided.

The Oxford English Dictionary defines automatic summarization as below:
DEFINITION 2.11 the creation of a shortened version of a text by a computer program. The product of this procedure still contains the most important points of the original text.

Mani & Maybury (1999) defines automatic text summarization as follows:
DEFINITION 2.12 automatic text summarization takes a partially structured source text from multiple texts written about the same topic, extracts information content from it, and presents the most important content to the user in a manner sensitive to the user’s needs.

Mani (2001) defines the goal of automatic text summarization as:
DEFINITION 2.13 the goal of automatic summarization is to take an information source, extract content from it, and present the most important content to the system’s user in a condensed form which is sensitive to the user’s task.

Borra, et al. (2010) give the following definition for automatic summarization:
DEFINITION 2.14 automatic text summarization produces the cluster of information which is most relevant to the needs of the user.

Gupta & Lehal (2010) define the function of automatic summarization is as follows:
DEFINITION 2.15 The goal of automatic text summarization is condensing the source text into a shorter version preserving its information content and overall meaning.

Saggion & Poibeau (2013) defines automatic text summarization as follows:
DEFINITION 2.16 automatic text summarization [is] the computer-based production of condensed versions of documents.
Babar and Thorat (2014) define automatic summarization as:

**DEFINITION 2.17** text Summarization is a process of creating a shorter version of original text that contains the important information.

Kumar (2014) gives a detail definition for automatic text summarization:

**DEFINITION 2.18** automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Automatic summarization reduces a text file into a passage or paragraph that conveys the main meaning of the text.

After the introduction of multi-document summarization systems the definitions of automatic text summarization have evolved. Radev, Winkel & Topper (2002) give the new definition as:

**DEFINITION 2.19** [a summary is] a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that.

Saggion & Lapalme (2002) gives the definition as below:

**DEFINITION 2.20** a summary is a condensed version of a source document having a recognizable genre and a very specific purpose: to give the reader an exact and concise idea of the contents of the source.

Hovy (2005) gives a definition for an automatic summary taking the length of the summary into consideration as below:

**DEFINITION 2.21** an automatic summary is a text generated by a software that is coherent and contains a significant amount of relevant information from the source text. Its compression rate $\tau$ is less than a third of the length of the original document.

Torres-Moreno (2014) gives the formula for length of a summary as the compression rate below:
\[ \tau = \frac{|\text{Summary}|}{|\text{Source}|} \] (2.1)

Here \(| \cdot |\) indicates the length of document in characters, words, or sentences and \(\tau\) can be expressed as percentage (Torres-Moreno, 2014).

The common points of these definitions are that an automatic text summary should be condensed and shorter version of the original text document or documents, the process being automatic; contain the most important information; be coherent and be sensitive to user’s needs. Following these common points we can define automatic text summarization as below:

**DEFINITION 2.22** automatic text summarization is an automated process that produces a coherent and condensed version of a document or a group of documents, retaining the information considered as the most important, according to the needs of the user.

As **DEFINITION 2.22** suggest automatic text summarization helps us to process the ever-increasing BigData efficiently, which we, as humans, are incapable of handling (Torres-Moreno, 2014).

Torres-Moreno (2014) gives a comprehensive list of summarization types, which we have summarized below:

i. According to their function
   - *indicative summary*: a summary resembling to a table of contents and providing information about the topics in the document;
   - *informative summary*: a short version of the document reflecting the content of the source text.

ii. According to the number of documents for summarization:
   - single document: a summary of one document;
   - multidocument: a summary of a group of documents often about a specific topic.

iii. According to genre of document:
   - *news summary*: a summary of news articles;
o **specialized summary**: a summary relating to a specialized domain;

o **literary summary**: a summary of narrative documents, literary texts;

o **encyclopedic summary**: a summary of encyclopedic documents;

o **social network summary**: a summary of blogs and very short documents, such as tweets.

iv. According to type:

o *extract*: a collection of sentence fragments from the document;

o *abstract*: a reformulated text by rewriting and paraphrasing;

o *sentence compression*: a text comprising of the same number of sentences as the original text, but with a reduced sentence length.

v. According to the type of summarizer:

o *author summary*: written by the author of the document;

o *expert summary*: written by somebody who specializes in the domain of the document;

o *professional summary*: written by a professional summarizer who has mastered the techniques of producing summaries.

vi. According to context:

o *generic summary*: a summary which ignores users’ information needs;

o *query-guided summary*: a summary guided by information needs or by users’ queries;

o *update summary*: a summary which only shows important new information and avoids repeating information.

vii. According to the target audience:

o *without a profile*: a a summary based uniquely on information from the source documents;

o *based on a user profile*: a summary targeted at users interested in a specialized domain
Following Torres-Moreno (2014) the automatic summarization process can be illustrated as in Figure 2.2:

Before going into detail and explaining the methods used in automatic summarization, the steps to be followed and the systems devised in the previous studies, in the next section, a brief history of the field and make mention of some important studies will be provided.

2.5 A Brief History of Automatic Text Summarization

Differently from its beginning, automatic text summarization is currently an interdisciplinary field of research benefiting from the expertise of numerous fields other than Natural Language Processing (NLP) such as computer science, artificial intelligence, statistics, cognitive sciences, Natural Language Generation, Machine
Learning, linguistics, discourse analysis. In this section we will only focus on automatic text summarization studies, sometimes emphasizing the contribution of other fields.

The concept of automatic text summarization dates back to Luhn (1958), where Luhn describes some exploratory research on automatic methods of obtaining abstracts. At the same time Baxendale (1958) will add information from the original resource. In 1963, Vasiliev presented a report to the UNESCO and gave the state of automatic abstracting at that time. This study makes mention of a statistical approach, a descriptor approach and a semantico-logical approach to automatic text summarization (Vasiliev, 1963).

In 1969, Edmundson develops a new approach to automatic summarization. Edmundson (1969), on the contrary to the previous studies, focuses not only on the presence of high-frequency content such as keywords, but also the pragmatic words (cue words); title and heading words; and structural indicators (the location of the words in a sentence). Following this study, Rush, Salvador & Zamora (1971) devised an algorithm using contextual inference, the location method, the cue method, the frequency data and coherence considerations to select or eliminate sentences from the document.

Later, Pollock & Zamora (1975), customized the Rush-Salvador-Zamora algorithm (Rush, Salvador & Zamora, 1971) for chemistry, more especially for pharmacodynamics. The study used a specialized database comprising of pharmacodynamics texts, where the previous studies used more general databases comprising of novels, textbooks, and magazine or newspaper articles. The study claimed that automatic summarization was more successful with some particular text genres than others.

In 1978, Yale Artificial Intelligence Project announced its new software FRUMP (Fast Reading Understanding and Memory Program) which was devised for skimming and summarizing newspaper articles (deJong, 1982). The FRUMP system used a
structure called *sketchy script*, some kind of pragmatic and semantic knowledge frame in addition to its linguistic knowledge. The pragmatic/semantic knowledge comprised a high-level reasoning system and the low-level (linguistic) text analyzer was sensitive to this high-level reasoning (deJong, 1982).

The 1980’s are relatively silent for automatic text summarization studies, but 1990’s and 2000’s witnessed an explosion in research on automatic text summarization due to the commercialization of the Internet. Spärck-Jones (1993) defined summarization as an information management process and devised a process structure for a summarizing task, based on human summarizing, automatic summarizing and discourse interpretation and representation. Spärck-Jones’ approach makes use of discourse analysis approaches such as Rhetorical Structure Theory (Mann & Thompson, 1987) and has references to other linguistics resources such as Halliday & Hasan (1976), Rumelhart (1977) and Kintsch & van Dijk (1978). The system uses linguistic sources, domain sources similar to the *sketchy scripts* used in FRUMP and communicative sources to perform summarization.

In 1995, Kupiec, Pedersen & Chen focused on document extracts as a particular kind of computed document summary and developed a trainable summarization program using a statistical framework. They used a feature set comprising of the *sentence length cut-off feature, fixed-phrase feature, paragraph feature, thematic word feature* and *uppercase word feature* to build the summarizer (Kupiec, Pedersen & Chen, 1995).

The same year, McKeown & Radev (1995) presented a multi-document summarizer system (SUMMONS) for articles on the same event. SUMMONS was comprised of a content planner, which selects information from an underlying knowledge base, and a linguistic generator which chooses the words to refer to the concepts contained in the selected information and which combines these words into grammatical sentences.

Marcu (1997), uses Rhetorical Structure Theory (Mann & Thompson, 1988) and refers, heavily, to discourse markers and lexico-grammatical constructs. The study
develops a rhetorical parsing algorithm to hypothesize rhetorical relations between
textual units and to map natural language texts onto discourse trees automatically.

Barzilay & Elhadad (1997) devised a technique to summarize texts without
requiring its full semantic interpretation using topic progression in the text derived
from lexical chains. The study merges several robust knowledge bases such as the
WordNet, a part-of-speech tagger, a shallow parser for nominal groups and a
segmentation algorithm.

Carbonell & Goldstein (1998) combines query relevance with information novelty.
The Maximal Marginal Relevance (MMR), which emphasizes relevant novelty, aims
at reducing the redundancy while maintaining query relevance, especially for multi-
document summarization.

Witbrock & Mittal (1999) proposes a summarization method that uses statistical
models of the term selection and term ordering process to produce even briefer and
compacter coherent summaries in a style learned from a training corpus.

Hovy & Lin (1998; 1999) proposes a summarization system (SUMMARIST) which
summarizes a text in three phases:
i) topic identification which includes position module, cue phrase module,
topic signature module, discourse structure module, topic identification
integration module;
ii) topic interpretation which includes concept counting and the wavefront,
interpretation using topic signatures;
iii) summary generation which includes a microplanner and a sentence
generator.

Knight & Marcu (2000) sets of from the fact that previous studies had focused on
only extraction of information; however, simply combining textual segments would
not yield coherent outputs. This study uses a decision-based model to reduce sentence
size and compress sentences.
Radev, Jing & Budzikowska (2000), present a multi-document summarizer (MEAD) which generates summaries using cluster centroids, which consist of words which are central not only to one article in a cluster but to all the articles, produced by a topic detection and tracking system.

Saggion & Lapalme (2002) present a text summarization system that takes a raw technical text as input and produces an indicative informative summary. SumUM first the topics of the document, and then elaborates on some of these topics according to the reader’s interest.

During 2000’s a series of Document Understanding Conference (DUC) yielded very fruitful research on automatic text summarization. Marcu (2001) proposes a system of discourse-based summarization algorithms (DUC-2001) for both single documents and collections of documents. The single document summarization employs the following steps (Marcu, 2001):

i) derive the discourse structure of the text given as input
ii) determine the important sentences in the input document
iii) determine all co-reference links in the input document
iv) increase summary coherence and compactness
v) generate summary

DUC-2001 uses the following stems for summarizing document collections (Marcu, 2001):

i) pre-process the collection
ii) select and order the sentences that summarize the collection
iii) resolve third person pronouns
iv) rank headlines
v) generate summaries

In 2004 Erkan & Radev introduce LexRank which uses a stochastic graph-based method for computing the relative importance of the textual units. Erkan & Radev
(2004) presented a new approach to define sentence salience based on graph-based centrality scoring of sentences. The authors claim that constructing the similarity graph of sentences provides a better view of important sentences compared to the centroid approach (Radev, Jing & Budzikowska, 2000).

Barzilay & McKeown (2005) discuss the “sentence fusion” method within a multi-document summarization system (MultiGen) and argue that sentence fusion involves bottom-up local multisequence alignment to identify phrases conveying similar information and statistical generation to combine common phrases into a sentence; thus producing abstract that contain sentences not found in any of the original documents.

Fernández, SanJuan, & Torres-Moreno (2007) present a Neural Network approach, inspired by statistical physics of magnetic systems, to model documents as neural network whose Textual Energy is studied. The model yielded good results in automatic summarization and Topic Segmentation.

Svore, Vanderwende & Burges (2007), presents a novel approach to automatic single-document summarization based on neural networks, called NetSum. The study is the first to use both neural networks for summarization and third-party datasets for features, using Wikipedia and news query logs.

Saggion (2008) presents a set of adaptable summarization components together with well-established evaluation tools. The toolkit (SUMMA) includes resources for the computation of summarization features which are combined in order to provide functionalities for single-document, multi-document, querybased, and multi/cross-lingual summarization.

Filippova (2010) devises a method called multi-sentence compression a syntax-lean method which requires little more than a tokenizer and a tagger. The method is a graph-based method which produced compressed and grammatical sentences which does not require neither a parser nor handcrafted linguistic rules.
Nenkova & McKeown (2011) provide a comprehensive overview the 50 years of research in summarization. They also discuss the challenges which are still open in the field of summarization such as language generation and deeper semantic understanding of language. They start by categorizing summaries and explaining how summarization systems work. Later they elaborate on the steps and methods used in summarization process.

Torres-Moreno (2012) presents another algorithm (ARTEX) for Automatic Text Summarization which calculates a simple inner product, comprising of a document vector (text topic) and a lexical vector (vocabulary in the sentence), between each sentence. Later the algorithm generate summaries by assembling the highest ranked sentences. The algorithm retains the salient information of each sentence of the document and it does not require any linguistic knowledge.

Litvak & Vanetik (2014) present a new model for extractive summarization and try to obtain a summary that preserves the information coverage as much as possible. They use a new tensor-based representation that describes the given document set in terms of its topics. Later these topics are ranked via Tensor Decomposition, and a summary from the sentences of the highest ranked topics.

Torres-Moreno (2014) give a more comprehensive account of automatic text summarization beginning with the foundations of the topic and discussing the most important concepts, methods, systems and evaluation systems.

After this brief review of the literature on automatic text summarization, the timeline for automatic text summarization as can be drawn as in Figure 2. following Torres-Moreno (2014). In Section 2.6 an account of the studies on automatic summarization in Turkey will be given.
Figure 2.3 A timeline of automatic text summarization
2.6 Automatic Text Summarization Studies in Turkish

This section aims at giving an account of the studies on automatic text summarization in Turkish. Although automatic text summarization studies, and natural language processing studies in general, can be traced back to Köksal (1981) in the early 1980’s, this section will only make mention of the ones which are related to the subject of this study, leaving aside many NLP studies, which are important in their own rights.

It can be claimed that the first systematic study of NLP and automatic text summarization started with Oflazer & Kuruöz (1994), in which they developed a POS tagger for Turkish based on a full-scale-two-level specification of Turkish morphology. The tagging tool integrated morphological analysis, multi-word and idiomatic construct recognition, morphological disambiguation, and root and lexical form statistic compilation; where the second and the third functionalities were implemented by a rule based subsystem (Oflazer & Kuruöz, 1994). The tagger uses a multi-word construct processor to detect and tag fixed expression such as duplications and other forms of semantic coalescences such as proper nouns which may generate spurious or incorrect results. The authors claimed that the tagger runs with 98-99% accuracy with minimal user intervention.

Later in 2003, Tür, Hakkani-Tür & Oflazer (2003) presented the results of their study on information extraction from unrestricted Turkish text using statistical language processing methods. The system developed in the study used statistical models. The authors of the study built a model which used surface forms of the words and their underlying forms, i.e. their morphological structure. This was done by using a preprocessing module which tokenized the data, analyzed these tokens and gave the most probable morphological analyses. The system used a topic segmentation module to define the topic clusters in the data using the word-based model and the stem-based model in combination. Later the system used a module to extract named entities, i.e. names of persons, locations, organizations, monetary values, percentage, dates and times. The authors claimed that their system reached an F-measure of 91.56%.
One of the most important studies for automatic text summarization in Turkish is Bilgin, Çetinoğlu & Oflazer’s (2004) initiative for developing a WordNet for Turkish. The authors started with a “first set of concepts” comprising of 1310 base concepts of the EuroWordNet project. After translating the first set of 1310 base concepts, the authors attempted to automatically extract synonyms, antonyms and hyponyms for these base concepts. Then they expanded the WordNet to 5000 base concepts during a second phase and then to 8000 base concepts in a third phase. The system run at 68% accuracy.

Karakaya & Güvenir (2004) conducted a study to integrate text classification and text summarization to compile and extract information from large bodies of texts. The system, named ARG, was a two-phase algorithm in which the paragraphs were classified according to given topics and then each topic was summarized into an automatically generated report. ARG, differently from other systems in the literature, used paragraphs as the unit of analysis, instead of the whole text. Also they used a user supervision for text classification to enable the users with a better option for expressing their information needs. As Karakaya and Güvenir (2003) explained In ARG;

i. the user determines the subject topics
ii. the user splits one or more article into their paragraphs and distributes them into topics
iii. Each topic is indexed using paragraphs in step ii
iv. Other documents are split into paragraphs
v. Paragraphs in step iv are classified according to given subject topics in step i
vi. Each topic is summarized
vii. Summaries are compiled and outputted as a report

However this algorithm needs too much intervention from the user and classification and summarization of large bodies of text could be a burden for the user.

Another text classification study was conducted by Amasyalı & Diri (2006) to determine the authors of documents and the gender of the author, and to classify the
genre of documents. They used an n-gram model and Naïve Bayes, Support Vector Machine, C4.5 and Random Forest methods to classify documents.

In another study, Ercan (2006) presented an extractive summarization system focusing on important sentence and key phrase identification. The system used lexical cohesion and cohesion, and identified the topics and the segments of the texts via lexical chains. The study focused on the specificity of the words in the WordNet and using this specificity score identified the key phrases. The study also used co-occurrence analysis to capture the links between actors, places and other objects, i.e. the participants of a state-of-affairs, which could not be captured via lexical chains approach. The author claimed that the system obtained promising results in overall.

Ercan & Çiçekli (2007) described a keyword extraction method using lexical chain features which improved the precision significantly. The authors tried to devise a way to form lexical chains out of phrases which were not represented in the WordNet already.

Kutlu, Çığır & Çicekli (2010) proposed a generic text summarization method via sentence ranking. The system calculated the sentence scores with regard to their surface level features and created the summaries by extracting the highest ranked sentences from the original documents. The system used features such as term frequency, key phrase centrality, title similarity and sentence position. The study was a first in many aspects that it showed the effectiveness of the centrality feature and introduced the use of key phrases in text summarization in Turkish.

Özsoy, Çiçekli & Alpaslan (2010) proposed two new LSA based summarization algorithms. They presented a generic extractive Turkish text summarization system based on LSA. The authors claimed that the Cross method devised in the study was better than any of other LSA methods.

Uzun-Per (2011) proposed a concept extraction system for Turkish. The system first pre-processed the characters of Turkish alphabet and then it only used nouns, which
were stripped from their inflectional morphemes. Then these nouns were clustered via k-means algorithm and concepts were assigned to this cluster of words using a user interface. Then the corpus of documents was tested against the list of concepts determined in the previous phase of the algorithm. The author claimed that the system devised in the study achieved 41% accuracy, which was low but higher than other studies in the literature.

In a different study Demir, El-Kahlout, Ünal & Kaya (2012) presented their efforts to build the first Turkish paraphrase corpus, which was an important step for creating summaries. They drew parallel sentences from multiple translations of literary texts, two different subtitles of a movie, multiple reference translation of a parallel corpus and human-written paraphrases of news sentences. Their system contained 1270 paraphrastic sentences.
CHAPTER THREE
TEXT SUMMARIZATION METHODOLOGY

3.1 Introduction

After providing a detailed account of what summary, extract and summarization are and providing an extensive history of automatic text summarization in general and specific to Turkish, now the methods, models, techniques and theories used in automatic text summarization can be explained. These may be grouped in various ways; according to their origin, i.e. linguistics, statistics, information extraction (IE), information retrieval (IR); according to their order of functioning in the automatic text summarization process, i.e. pre-processing, processing and post-processing. The second grouping will be used and the origins of the methods, models, techniques and theories will be provided as the occasion arises.

The second grouping, mentioned above, is based on the order of the methods, models, techniques and theories used in the automatic text summarization process. The grouping can be rendered as pre-processing, processing and post-processing in general; though different stratifications may be offered instead the grouping as a whole or within the group or subgroups. The general algorithm for automatic text summarization based on the grouping proposed here can be drawn as in Figure 3.1:

Figure 3.1 The general algorithm of automatic text summarization
As it is shown in Figure 3.1, automatic text summarization requires a three-phase approach. The pre-processing approach comprises of splitting the text into segments, tokens and annotating the text for process, i.e. the summarization process. Since most of the texts around us are unstructured plain texts or texts which are formatted in numerous ways, these need to be standardized and prepared for processing. The pre-processing phase prepares the texts to be summarized for the summarizer system.

After the texts are prepared for the summarizer, these need to be represented in a machine-readable way to be evaluated for summary. The segments or units prepared in the previous phase are represented in different ways, e.g. vector spaces, matrices, weights, graphs or energy. This is necessary for determining the important parts of the texts, which would be included in the summary, the final output of the system. After construction of a representation of the text to be summarized, this representation is processed by the summarization algorithm in terms of various features. Later, important and salient parts or units of the representation are selected to be included in the summary and assembled together to produce an output.

In the last phase the output of the algorithm may or may not be processed again. The output of the algorithm may be presented to the user as is, i.e. the most important or salient sentences in the representation are selected and adjoined as their original form in the original texts. However, these selected sentences may be processed again to form paraphrases and to present the summary differently.

3.2 Pre-processing in Automatic Text Summarization

The plain text document which is the input for summarization has an information load that cannot be easily processed and thus needs to be reduced. Pre-processing enables transferring plain texts, loaded with excess information, to the next phase, with their minimal parts, loaded with the minimal linguistic information needed for processing. The preprocessing phase of automatic text summarization comprises of three sub-phases which also have sub-phases which are discussed in the following sections.
### 3.2.1 Splitting

The pre-process phase begins with splitting the text or texts to be summarized into segments or parts. These segments may, according to the length and genre of texts, be episodes, chapters, paragraphs, or sentences, words, etc. Here, it will only be focused on sentence level splitting, and thus only sentence boundary processing and punctuation will be mentioned. Languages have different punctuation marks, e.g. Spanish has upside-down question and exclamations marks, and different punctuation customs which are not uniform and vary over time and location (Woods, 2006). Even language users from the same location and same time period use punctuation differently. In order to cope with the complications of this issue the texts should be standardized before summarization.

Sentence splitting is the process of inserting a suitable symbol that can be distinguished from the original text (for instance </S>) at the end of each recognizable sentence (Torres-Moreno, 2014). However, finding a sentence’s boundaries is not an easy task as it seems to be. It’s a general punctuation custom to end the sentences with a full stop (.), however, the use of quotation marks (“ ”), colons (:), bulleted lists (either symbolic or numeric), ordinal numbers (1.) complicates the issue. An example is presented in (3.1) (Aktaş & Çebi, 2013):

> Uluslar, bu ekonomik buhran sonucunda 2. Dünya Savaşı’ni yaşamıştır. 
> Nations faced with the World War II as a result of this economic crisis. 

In (3.1) the use of the ordinal number “2.” complicates the sentence boundary detection process and returns false boundaries. The punctuation mark “.” here does not signal the end of the sentence, but the ordinal number.
3.2.2 Tokens

Tokens can be defined as single instances of symbol, a linguistic unit in our context. A text summarizer should be capable of identifying different words (tokens) and thus giving the correct number of occurrences of a linguistic unit in the text. Tokens are important since they provide the unique word types, which are general classes of tokens, in a text (Torres-Moreno, 2014).

Tokens: Love, loves, loving, loved, hates, hated
Types: Love, hate

3.2.2.1 Tokenization

Tokenization is the task of chopping it up into pieces, called tokens, throwing away certain characters, such as punctuation (Manning, Raghavan & Schütze, 2008):

Input: Friends, Romans, Countrymen, lend me your ears
Output: Friends Romans Countrymen lend me your ears

However, Tokenization has some issues which are language specific (Manning, Raghavan & Schütze, 2008):

Input: O’neill
Possible Outputs: neill, oneill, o’neill, o’ neill, o neill

It’s difficult to decide which of the possible outputs of tokenization in (3.4) is the desired one. It is, again, not an easy task because idioms and frozen expressions raise complex issues (Habert et al., 1998); the exact limits of the notion of “word” are not easy to establish and even some “words” have no autonomous existence (Haber et al., 1998: 3). For instance “de” clitic in Turkish has no autonomous existence; it has to attach to a previous word. In addition to these complications, the inconsistencies in
orthography such as use or non-use of the circumflex (') in Turkish, and the negligence of the orthography by even well-educated writers (e.g. not writing interrogative clitic “mi” separately”) may cause problems for tokenization.

3.2.2.2 Stop Word Identification

Stop words are extremely common words that appear to have little or no semantic content, such as articles, conjunctions, figures, punctuation and special symbols (Torres-Moreno, 2014). These stop words should be removed from the text to be processed to ease and speed up the processing. The general strategy for determining a list of stop words is to sort the tokens in terms of their frequency, and then to make a list, a stop list, from the most frequent tokens, checking their semantic content (Manning, Raghavan & Schütze, 2008). An example of a stop list is presented in Figure 3.2:

<table>
<thead>
<tr>
<th>ama</th>
<th>bazı</th>
<th>çok</th>
<th>dolayısıyla</th>
<th>her</th>
</tr>
</thead>
<tbody>
<tr>
<td>ancak</td>
<td>belki</td>
<td>çünkü</td>
<td>eğer</td>
<td>iş</td>
</tr>
<tr>
<td>arada</td>
<td>bir</td>
<td>da</td>
<td>gibi</td>
<td>ki</td>
</tr>
<tr>
<td>ayrıca</td>
<td>böyle</td>
<td>daha</td>
<td>halen</td>
<td>kim</td>
</tr>
</tbody>
</table>

Figure 3.2 A portion of stop word list by Can et al., 2008

The tokens to be included in the stop word list should be selected very carefully because some frequent tokens may have semantic content or may have discursive functions which would signal the importance of a sentence.

3.2.2.3 Stemming and Lemmatization

Documents would have different forms of the same word, such as words forms related via inflection, e.g. *car, cars, car’s, and cars’* (Manning, Raghavan & Schütze, 2008). Also there are word forms that are related via derivation, e.g. *gözlık, gözlıkçü, gözlıkçülük*. In automatic summarization, sometimes, it is needed to reduce the word...
forms to a common base form, i.e. lemma or lexeme, by removing derivational and inflectional forms (Manning, Raghavan & Schütze, 2008).

**Word forms:** car, cars, car’s, cars’

**Lemma:** car

**Word forms:** am, is, were

**Lemma:** be

In (3.5) the derivational forms on the lemma *car* are removed. This process is called *stemming*. Stemming is usually chopping off the ends of words, generally hoping to achieve correct stems. It often includes the removal of derivational affixes (Manning, Raghavan & Schütze, 2008).

In (3.6), on the other hand, inflectional forms *am, is* and *were* are reduced to a single lemma *be*. This process is called *lemmatization*. Lemmatization is the process of removing inflectional endings only with the use of a vocabulary and morphological analysis of words, to return the base or dictionary form of a word, which is known as the *lemma* (Manning, Raghavan & Schütze, 2008). Issues in lemmatization in Turkish will be addressed in Chapter Four which explains the algorithm developed in this thesis, after providing typological information on Turkish.

### 3.2.2.4 Capitalization or Case-Folding

Capitalization or case-folding is a common strategy for reducing all letters to lower case to allow instances of words in different cases to be matched by the system (Manning, Raghavan & Schütze, 2008). This process may useful for search engines; however it may raise some issues such as matching proper nouns with their common noun or adjective counterparts, e.g. Mesut (proper noun) and mesut (adjective). This issue may be dealt with by converting to lowercase the words at the beginning of a sentence and all uppercase words; and leaving midsentence capitalized words as they are.
3.2.3 Annotation

Annotation can be defined as the practice of adding interpretative, linguistic information to an electronic corpus of spoken and/or written language data (Leech, 1997) to enhance the raw data with relevant linguistic information (Pogodolla, 2008). Also, annotation, as an output of the annotation process, covers any descriptive or analytic notations applied to raw language data (Bird & Liberman, 1999). Some central examples of annotations are morphological parsing, part-of-speech tagging and syntactic bracketing; phonetic segmentation and labeling; annotation of prosody and gesture, and discourse structure; marking of co-reference, ‘named entity’ tagging, and sense tagging; and phrase-level or word-level translations (Bird & Liberman, 1999).

However, since this thesis is interested in automatic text summarization, only the aspects presented below will be covered in the thesis:

i. Morphological parsing
ii. Part-of-speech (POS) tagging
iii. Named entities
iv. Discourse structure

3.2.3.1 Morphological Parsing

Morphological parsing is the process of breaking a token such as çocuklar (children) into the constituents, here to its morphemes, as {çocuk} (child, the root morpheme) and {-lAr} (the plural morpheme). A morphological parser generally comprises of three components (Sak, Güngör & Saraçlar, 2008):

i. a lexicon, listing the root morphemes of any language with part-of-speech information,
ii. a morphotactics component, which enumerates the inventory of morphemes and specifies in what order they can occur (Antworth, 1993)
iii. a morphophonemics component, which accounts for alternate forms or "spellings" of morphemes according to the phonological context in which they occur (Antworth, 1993).
3.2.3.2 Part-of-Speech (POS) Tagging

Lexical categories or parts-of-speech (POS) have been recognized in linguistics for a long time (e.g. by Panini, Aristoteles, Thrax, and Varro) (Voutilainen, 2003). In traditional grammar parts-of-speech are distinguished by semantic criteria; however, the most valid criteria for parts-of-speech are grammatical (Voutilainen, 2003). Morpho-syntactic tagging, also known as grammatical tagging or POS-tagging, is a process that consists of automatically associating words in a text with corresponding grammatical information: verbs, adjectives, nouns, gender, number, etc. (Torres-Moreno, 2014). An example of POS-Tagging is presented in (3.7):

**Input:** A man can be happy without any woman as long as he does not love her.

**Output:** A/DT man/NN can/MD be/VB happy/JJ with/IN any/DT woman/NN as/RB long/RB as/IN he/PP does/VBZ not/RB love/VB her/PP /SENT

Here the tag /DT represents a determiner, /NN a noun, /MD a modal, /VB a verb, /JJ an adjective, /IN a preposition, /RB an adverb, /PP a personal pronoun, /VBZ a verb inflected for 3rd person singular present tense and /SENT sentence.

Many POS-taggers have a remarkably similar architecture (Voutilainen, 2003):

i. Tokenization: see 3.2.2.1

ii. Ambiguity look-up: involves use of a lexicon and a guesser for tokens not represented in the lexicon

iii. Ambiguity resolution or Disambiguation: resolving the ambiguity for tokens which can be parsed in more than one way.

3.2.3.3 Named Entity Tagging
Named entities are tokens referring to people, organizations, places (cities, countries, etc.) and physical and temporal quantities (Torres-Moreno, 2014). Here, the task is to tag text to identify proper names of persons, organizations, and locations, as well as certain numerical and temporal expressions, (dates, days, months, times, percentages, money) (Hirschman & Mani, 2003). However, the definition of named entity still remains opaque (Torres-Moreno, 2014) and requires exhaustive lists of named entities to properly tag the tokens with named entity tag.

3.3 Processing in Automatic Text Summarization

After the text to be summarized is pre-processed, i.e. split into its sentences, then tokens, the morphological analysis is conducted on these tokens and the tokens are tagged in terms of parts-of-speech and named entity tags, all the data in the previous phase is transferred into processing phase. This data is cleaned from excess information, and tagged with necessary linguistic information which is required to represent and retrieve required information from the text to be summarized. This representation of texts in a machine understandable way constitutes the core of automatic text summarization.

3.3.1 Automatic Text Summarization Models

Numerous models have been developed to retrieve information such as the Boolean model, the Statistical model, which includes the vector space and the probabilistic retrieval model, and the Linguistic and Knowledge-based models. The Boolean model is also called as the "exact match" model (Spoerri 1995).

3.3.1.1 Boolean Model

In a Boolean model a query can be characterized with regard to four dimensions (Spoerri 1995):

i. Use of operators, such as AND, OR and NOT
ii. Proximity constraints between terms
iii. Term-Field relations

iv. Stemming operations on words

In the first dimension, the model coordinates the concept determined by the users by means of the Boolean operators (AND, OR, NOT). In the second dimension, the terms, in order to form phrase-like units, are subjected to proximity constraints, such as occurring next to each other, or in the same segment of the text. Third dimension enables identifying terms in particular fields of a structured text document (author, title, keywords, abstract, index). In the fourth dimension, stemming or lemmatization is performed to reduce a term to its simplest morphological form, i.e. the stem, thus use this as a prefix for queries (Spoerri, 1995).

The Boolean approach is easy to implement and computationally efficient; however it is difficult for an unexperienced user to build Boolean queries. In additions the Boolean approach does not differentiate between the query terms and their weight in the text (Spoerri, 1995).

The statistical methods provide a (relevance) ranking of the documents and their queries are easier to formulate. However, they have limited expressive power, they lack the structure to express important linguistic features such as phrases; they can be computationally expensive; queries have to contain large number of terms to better represent the documents (Spoerri, 1995).

3.3.1.2 Statistical Models

Statistical model uses the statistical information to determine the relevance of documents with regard to a query, in the form of term frequencies. There are two major examples of the statistical IR: vector space model and probabilistic model (Spoerri, 1995).

3.3.1.2.1 The Vector Space Model. The vector space model generates a representation of documents and query terms in the form of vectors in a multidimensional space. The dimension of this space is determined by the number of
the terms used to build an index of the documents (Spoerri, 1995). This index is generated by lexical scanning for identifying important terms, which are reduced to stems after morphological parsing. This model seeks to find the relation between the N documents in a collection and a set of k characteristics in this collection. It is also called the bag-of-word models since the order of the words are not important in this model (Torres-Moreno, 2014).

The vectorization of documents creates a document-term matrix where each column represents the weight of a term in a sentence (or a document). However, the matrices created have a great volume and noise (Torres-Moreno, 2014). Thus, the documents should be pre-processed to reduce volume and noise.

3.3.1.2.2 The Probabilistic Model. The probabilistic model ranks the documents based on their probability of relevance to the query terms. The most common probabilistic method is the statistical distribution of the terms in both relevant and non-relevant documents (Spoerri 1995).

3.3.1.3 Linguistic and Knowledge-Based Model

Linguistic and knowledge-based approaches perform a morphological, syntactic and semantic analysis to process documents more effectively. In morphological analysis, POS-tagging is conducted. In the syntactic analysis complete phrases are parsed. Then, the semantic relations between the words are analyzed to resolve ambiguities or generate relevant synonyms, quasi-synonyms and sometimes antonyms. It is a very sophisticated task to develop linguistic and knowledge based model since it requires a large linguistic knowledge base; hence these system often referred as to expert systems (Spoerri, 1995).
3.3.1.4 Graph-Based Model

The graph-based models try to determine the most prestigious sentences in a document. In order to achieve this goal, they represent the document by a graph of text units to find the most central or prestigious sentences in the graph (Torres-Moreno, 2014). The models first construct a graph by representing the sentences in the document as vertices and the similarity between these sentences as weighted edges or lines as in Figure 3.3 (Torres-Moreno, 2014). Then the sentences with the highest ranking, with the heaviest link are determined (Torres-Moreno, 2014). In automatic summarization generally, the whole sentences are extracted to protect the grammaticality of the segments selected for summary. Graph-based models enabled the selection of the most central sentences, which carry the most important information in the document (Torres-Moreno, 2014).

Figure 3.3 Graph-based representation of a document

Following Spoerri the representation models can be summarized, relating them to the levels of linguistic structure as in Table 3.1.
As it can be read in Table 3.1, Boolean and Statistical models deal with the lexical level using stop word lists, while linguistic and knowledge-based models and graph-based models have full-featured lexicon. For the morphological level, Boolean models use truncation symbols; statistical models and graph-based models need stemming in the text pre-processing phase; and linguistic and knowledge-based models need a full morphological parsing. For syntactic analysis, Boolean models use proximity of the tokens; statistical models and graph-based models use statistical phrases; the linguistic and knowledge based models conduct a full syntactic parsing to find grammatical phrases. Lastly, the semantic level is analyzed via a thesaurus in Boolean models; statistical models use the cluster of co-occurring forms; and the linguistic and knowledge-based models and graph-based models use a network of words or phrases in semantic relationships.

### 3.3.2 Automatic Text Summarization Algorithms

Torres-Moreno (2014) asserts that there are three families of approaches for automatic text summarization:

i. summarization by extraction;

ii. summarization by abstraction;

iii. summarization by sentence compression.

According to Radev, Hovy & McKeown (2002), extraction is the process of identifying important segments of a text; abstraction is the process of reformulating...
and fusing these important segments in novel terms; and compression is the process of trimming unimportant material.

3.3.2.1 Summarization by Extraction

Extraction is the process of selecting segments of text which are evaluated as carrying important information in the document and of assembling these units in an adequate way (Torres-Moreno, 2014).

Radev, Hovy & McKeown classify automatic summarization by extraction approaches into three categories, i.e. surface-level, intermediate-level and deep parsing techniques. Many & Maybury (1999) name these categories as surface-level, entity-level and discourse-level. Surface level approaches process shallow features, such as presence of statistically salient terms, location of terms in text, cue words. Entity-level approaches are based on modeling semantic, syntactic, and logical relations between text entities. Discourse-level approaches are based on modeling the global structure of the text (Iatsko, Shilov & Vishniakov, 2005).

3.3.2.1.1 Surface Level Approaches. Surface-level approaches use certain shallow linguistic elements to identify the most relevant segments of a document (Torres-Moreno, 2014). In order to identify these linguistics elements (i.e. words), the algorithm requires a lexicon (i.e. dictionary) to serve as a database. Then the algorithm scans the input text to find segments that contain the elements in the lexicon (Iatsko, Shilov & Vishniakov, 2005); this is called occurrence and the occurrences of words are used to weighted the segments (Torres-Moreno, 2014). The segments with more frequently occurring elements are ranked higher.

It is also possible to rank segments using the words in the title of the document. Title words are often acknowledged important and thus they put extra weight to the segment.
Some surface-level approaches use the position of elements in the segments. Although the position for important may vary between languages, the first line of the paragraphs are considered important.

Some other approaches also use index-expressions or cue-phrases, which signal important or unimportant information. Some linguistic clues that indicate the importance of a segment include “it is important to stress that, in conclusion, incidentally, consequently, etc.”; the cue phrases which signal the unimportance of a segment include “for example, anyway, by the way, etc.” These cue phrases are used to weight the segment. For instance, a cue phrase signaling importance increases the weight, while a cue phrase signaling unimportance decreases it.

3.3.2.1.2 Intermediate-level Approaches. Intermediate-level approaches build an internal representation for text, modeling text entities and their relationships (Mani & Maybury, 1999). They use linguistic information that is more sophisticated than surface-level algorithms. One type of this information is lexical chain recognition. Lexical chains are sequences of words connected by lexical semantic relations. (Torres-Moreno, 2014). For instance, \{house, loft, home, cabin\} is a chain, where house and home are synonyms, loft is part of a house and cabin is a specialization of house (Doran, Stokes, Carthy & Dunnion, 2003). After the text-preprocessing phase the algorithm constructs lexical chains and identifies the “strong” chains to rank or extract the segments (Torres-Moreno, 2014). Lexical chaining algorithms also require a knowledge-base to identify the semantic relationships between the words; and they mostly use WordNet.

3.3.2.1.3 Deep Parsing Approaches. Deep parsing approaches use in-depth linguistic techniques to exploit the discursive structure of texts. Some of these are based on Rhetorical Structure Theory (RST) (Mann & Thompson, 1988); some other are based on Meaning-Text Theory (MTT) (Mel’uk, 1988) (Torres-Moreno, 2014).

Rhetorical Structure Theory provides a general way to describe and represent the relations among clauses in a text, whether or not they are grammatically signaled
RST is based on the notion of rhetorical relation, which holds between a nucleus (N) and a satellite (S). The nucleus expresses what is more essential than the satellite; and that the nucleus of a rhetorical relation is comprehensible independent of the satellite, but not vice versa (Marcu, 2000).

An algorithm based on RST divides the text into discursive units using a minimal set of relations, a process which is known as discourse segmentation. The algorithm then divides the text into segments using a minimal set of relations and creates the discourse structure of the text. Later the algorithm weights the segments of this structure; and the segments with the highest weight are selected for the summary (Torres-Moreno, 2014).

3.3.2.2 Summarization by Abstraction

Summarization by abstraction approaches depend on text understanding and aims at generating a grammatically correct, concise and coherent summary, whose quality is comparable to summaries produced by people (Torres-Moreno, 2014). However, these approaches are extremely complex and far from being perfect since they seek to simulate human cognitive processes and behavior.

The semantic interpretation approach used in FRUMP (DeJong, 1982) tries to “understand” the documents by coding knowledge structures, based on semantics leaving syntax aside (Torres-Moreno, 2014). The algorithm uses a knowledge base comprising of scripts, proposed by Schank & Abelson, a structure that describes an appropriate sequence of events in a particular context (1977). Then the algorithm suggests scripts for keywords and weights the scripts and extracts some of them (Torres-Moreno, 2014). Although promising, this kind of algorithms requires manual entry of scripts and a huge lexicon.

Another extremely complex approach to abstraction-based automatic text summarization is topic interpretation used in SUMMARIST (Hovy & Lin, 1999). SUMMARIST process the pre-processed text by identifying topics using the
techniques explained in section 3.3.2.1.1 Surface Level Approaches. Then the results of these techniques are integrated. In the next step SUMMARIST interprets these topics via concept fusion. The algorithm fuses two or more identified topics into one or more unifying concepts (Hovy & Lin, 1999) by making use of WordNet.

Genest & Lapalme (2012) propose a methodology that uses information extraction (IE) and natural language generation (NLG). Their architecture depends on abstraction schemes consisting of IE rules, content selection heuristics and generation patterns, all created manually (Genest & Lapalme, 2012). The abstraction schemes make use of VerbNet (Kipper et al., 2006) which provides the semantic roles of the arguments of a verb. The algorithm then selects best schemes among many candidates and sends them to the generation module.

A study by Miranda-Jiménez, Gelbukh & Sidorov (2013) uses conceptual graphs to create a conceptual representation of the text. Conceptual Graphs (CGs) are structures for knowledge representation based on first-order logic. They are natural, simple, and fine-grained semantic representations (Miranda-Jiménez, Gelbukh and Sidorov, 2013). The system considers all words, except for stop words, as concepts; consider semantic roles as conceptual relations. The conceptual graphs are constructed manually and later they are ranked. The selected conceptual graphs represent the summary.

3.3.2.3 Summarization by Sentence Compression

Sentence compression approach is a relatively new subject in automatic text summarization. They enable reducing redundancy and creating summaries simulating human abstraction process (Torres-Moreno, 2014).

3.3.2.3.1 Sentence Compression. Sentence compression is the task of producing a summary of a single sentence. The compressed sentence should be shorter, contain the important content from the original, and be grammatical (Pitler, 2010). Since the
processes linguistic element is the single sentence, sentence compression may also be considered as a post-processing approach.

3.3.2.3.2 Multisentence Compression. Multisentence compression is the task of summarizing a cluster of related sentences, mostly from multiple documents (Filippova, 2010). Multisentence compression aims at reducing redundancy while maintaining the essential information contained in the cluster of semantically similar sentences (Torres-Moreno, 2014). This approach may also be used in the post-processing phase.

The tasks of sentence compression and multisentence compression may be conducted using three approaches: symbolic approach, statistical approach and a hybrid of the two, the linguistic-statistical approach.

3.3.2.3.3 Symbolic Approaches to Sentence Compression. In sentence compression the algorithm has to decide which parts to be removed from the sentence. In symbolic approaches, symbolic, in other words, linguistic clues are used to determine these parts. Jing (2000) uses a parallel corpus of full sentences and their compressed counterparts; a lexicon comprising of over 5000 verbs with their subcategorization frames; the WordNet lexical database which provides lexical relations between such as synonymy, antonymy, meronymy, entailment and causation; and a syntactic parser to produce a parse tree of sentences which are annotated with categories and thematic roles of the constituents. Using all these symbolic information about linguistic elements the system may decide which parts of a sentence are unimportant.

Gagnon & Da Sylva’s algorithm performs sentence reduction using syntactic pruning of the sentences. It proceeds by analyzing the sentence, then filtering the targeted relations, while applying anti-filters which prevent certain unwanted pruning by the filter (2006). At the end the algorithm produces sentences shorter than six words without compromising grammaticality.
3.3.2.3.4 Statistical Approaches to Sentence Compression. Starting from the claim that determining the most important textual segments in a text is only the half of what summarization system need to do, Knight & Marcu (2002) propose a sentence compression approach. Their approach, the noisy-channel approach is a probabilistic one; and considers a sentence as a long string which was added some additional/optional text, i.e. noise. The approach tries to remove the additional information, the noise from the long string and to achieve the short string, i.e. the compressed sentence. The approach calculates the probability of transforming short strings into long strings.

3.3.2.3.5 Statistical-Linguistic Approaches to Sentence Compression. This approach to sentence compression combines sentence level discourse segmentation and the textual energy approach. The system decomposes discourse into Elementary Discourse Units and forms the discourse trees from these (Molina et al., 2013). Discourse segmentation is based on RST. Then it uses textual energy, a similarity measure used in various NLP tasks (Molina et al., 2013). Textual energy approach weights the discourse segments using an energy matrix which connects segments having common words, as well as segments sharing the same neighborhood but not necessarily identical vocabulary (Molina et al., 2013). After ranking the segments a general linear model determines which segments should be deleted.

3.4 Post-processing in Automatic Text Summarization

After the processing phase, where the text document or documents are represented and the candidate segments are selected for summary, these selected segments are assembled in the post-processing phase. Most extractive summarization approaches select whole sentences as segments and simply put them together to form a summary. Abstractive approaches, on the other hand, try to simulate human behavior and to produce more coherent summaries by applying some post-processing steps such as paraphrasing and sentence fusion. Sentence compression approaches as it was mentioned in 3.3.2.3 Summarization by Sentence Compression may also be considered as post-processing steps.
3.4.1 Paraphrasing

Paraphrasing can be defined as the restatement of ideas in a text document using different words or phrases. Paraphrasing algorithms in the literature (Barzilay & Lee, 2003; Pang, Knight & Marcu, 2003) use parallel corpora to obtain paraphrases. Kauchak & Barzilay (2006) use a novel approach for paraphrasing; they use statistical methods to decide whether words in the output of the processing phase might be substituted with other words. They call this method as contextual substitution.

3.4.2 Sentence Fusion

Sentence fusion is a step beyond sentence extraction and it does not require a deep parsing approach, i.e. it can be performed with shallow preprocessing approaches. Differently from sentence compression, which removes unimportant parts of the sentences, sentence fusion unifies the important information contained in two or more sentences into one or more sentences.

3.4.3 Sentence Enhancement

Sentence enhancement is a novel technique which extends sentence fusion by combining the subtrees of many sentences into the output sentence, rather than just a few (Cheung & Penn, 2014). The algorithm comprises of five steps: (1) clustering to identify initial input sentences, (2) sentence graph creation, (3) sentence graph expansion, (4) tree generation, and (5) linearization. The initial outputs of the processing phase are accepted as initial core sentences, and they comprise the core of the new sentence to be generated. Differently from sentence fusion and sentence compression, sentence enhancement fuses disparate sentences, which are not similar to the core sentences. In order to do so, the system generated the dependency trees of the core sentences and fused them into an intermediate tree. Then the system searches other dependency trees for important information, and candidate trees, or parts of these
trees, to the intermediate tree. Using a directed acyclic graph the algorithm generates the final sentences.

3.5 Conclusion

In this chapter, automatic text summarization with its pre-processing, processing and post-processing phases were reviewed. As it was mentioned in the previous sections, automatic text summarization is a very complex process which requires numerous approaches, models, techniques and methods to implement, ranging from information extraction to information retrieval, from statistics to linguistics. Thus, automatic text summarization requires a good knowledge of software, statistics, and last but definitely not the least, language. In the next chapter we will begin to explain the automatic text summarization system we have developed in this thesis in detail, after presenting detailed typological information on Turkish, which poses problems for automatic text summarization per se. Before going further to the next chapter, we may present the whole process of automatic text summarization, though incomplete due to excessive number of methods, techniques, models and approaches, in Figure 3.4.
Figure 3.4 Phases of automatic text summarization

- **Pre-processing**
  - Splitting
    - Sentence boundary
    - Punctuation
  - Tokenization
  - Stopword identification
  - Lemmatization
  - Stemming
  - Capitalization/Case Folding

- **Processing**
  - Boolean Model
  - Statistical Models
    - Vector Space Model
    - Probabilistic Model
    - Linguistic and Knowledge-Based Model
    - Graph-Based Model

- **Representation**
  - TS by extraction
    - Surface-Level Approaches
      - Circumstance
      - Title words
      - Sentence position
      - Index expressions
    - Intermediate-level Approaches
      - Lexical chains
    - Deep Parsing Approaches
      - Rhetorical structure
  - TS by abstraction
    - Semantic interpretation
    - Natural Language Generation
    - Conceptual graphs
    - VerbNet

- **Tokens**
  - Annotation
    - Morphological Parsing
    - POS-Tagging
    - Named Entities

- **Selection**
  - Assembly
    - Paraphrasing
    - Compression
    - Fusion
    - Enhancement

- **Post-processing**
  - Summary
4.1 Introduction

In this chapter detailed typological information on Turkish will be provided, since the system developed in this thesis is an automatic text summarization system for Turkish. Next the preprocessing phase of the automatic text summarization tool for Turkish will be introduced and detailed explanations about the steps will be given with examples.

4.2 Turkish from a Typological Perspective

Turkish belongs to the Turkic branch of Altaic language family and is the largest language (in terms of number of speakers) in the Turkic family (Kornfilt, 1997) which have been spoken for many centuries across a vast territory from the Balkans to China (Göksel & Kerslake, 2005). Turkish is the official and dominant language of Turkey (Turkish Republic), where it is the native language of near 90 per cent of the population, and also a co-official language (together with Greek) in Northern Cyprus (Kornfilt, 1997). Turkish speakers outside Turkey fall into two groups (Göksel & Kerslake, 2005):

i. communities located in various lands that were formerly part of the Ottoman Empire, e.g. Bulgaria (760,000), Greece (115,000), Macedonia (80,000) and Romania (23,000);

ii. residents of Western European countries, Australia and North America, e.g. Germany (2 million), Australia (40,000), North America (50,000)

4.2.1 Ortography

Turkish alphabet comprises of 29 letters, i.e. letters, 8 for vowels and 20 for consonants. It has a special grapheme \(<\ddag>\), accepted as a consonant symbol, however
it is controversial (see Ergenç, 2002; Özsoy, 2004; Kılıç & Erdem, 2008 for discussion). Turkish also has additional graphemes, although their use is in decline now. These are constructed with the circumflex “^” diacritic, e.g. “â” as in hâlâ, “î” as in sanayîî, and “û” as in Halûk (Koşaner, Birant & Aktaş, 2013: 474).

Turkish Alphabet:

Uppercase:

Lowercase:
a, b, c, ç, d, e, f, g, ğ, h, i, İ, j, k, l, m, n, o, ö, p, r, s, ş, t, u, ü, v, y, z

4.2.2 Phonology

Turkish has 32 phonemes, i.e. the smallest distinct sound units in a language (Matthews, 1997) capable of distinguishing meaning. The phonemes of Turkish are presented in Table 4.1 with their grapheme counterparts and examples.

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Grapheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>/p/</td>
<td>&lt;p&gt;</td>
<td>pas</td>
</tr>
<tr>
<td>/t/</td>
<td>&lt;t&gt;</td>
<td>tas</td>
</tr>
<tr>
<td>/k/</td>
<td>&lt;k&gt;</td>
<td>kâr</td>
</tr>
<tr>
<td>/c/</td>
<td>&lt;ç&gt;</td>
<td>baş</td>
</tr>
<tr>
<td>/b/</td>
<td>&lt;b&gt;</td>
<td>dal</td>
</tr>
<tr>
<td>/d/</td>
<td>&lt;d&gt;</td>
<td>gaga</td>
</tr>
<tr>
<td>/g/</td>
<td>&lt;g&gt;</td>
<td>gâvur</td>
</tr>
<tr>
<td>/ʃ/</td>
<td>&lt;ʃ&gt;</td>
<td>fasıl</td>
</tr>
<tr>
<td>/ʒ/</td>
<td>&lt;ʒ&gt;</td>
<td>sap</td>
</tr>
<tr>
<td>/h/</td>
<td>&lt;h&gt;</td>
<td>hile</td>
</tr>
<tr>
<td>/v/</td>
<td>&lt;v&gt;</td>
<td>vakur</td>
</tr>
<tr>
<td>/z/</td>
<td>&lt;z&gt;</td>
<td>zurna</td>
</tr>
<tr>
<td>/ʒ/</td>
<td>&lt;ʒ&gt;</td>
<td>jalużi</td>
</tr>
<tr>
<td>/ç/</td>
<td>&lt;ç&gt;</td>
<td>çan</td>
</tr>
<tr>
<td>/c/</td>
<td>&lt;ç&gt;</td>
<td>can</td>
</tr>
<tr>
<td>/m/</td>
<td>&lt;m&gt;</td>
<td>muz</td>
</tr>
<tr>
<td>/n/</td>
<td>&lt;n&gt;</td>
<td>nasił</td>
</tr>
<tr>
<td>/l/</td>
<td>&lt;l&gt;</td>
<td>laf</td>
</tr>
<tr>
<td>/ğ/</td>
<td>&lt;ğ&gt;</td>
<td>al/gi</td>
</tr>
<tr>
<td>/r/</td>
<td>&lt;r&gt;</td>
<td>raf</td>
</tr>
</tbody>
</table>
4.2.3 Morphology

Turkish is often classified as an agglutinating, language (Bickel & Nichols, 2005). However, a language does not wholly fit to a class. The morphological classification of languages as isolating, agglutinating, fusional and polysynthetic are scales not absolute values. Thus a language may presents features from one or more classes, presenting a tendency for a particular class. Turkish shows a great tendency for agglutinating languages; however, it also has some features, albeit minute, from fusional and polysynthetic languages:

\[
\begin{align*}
\text{burun} - u & \rightarrow \text{burnu} \\
\text{nose} - \text{POSS} & \rightarrow \text{his nose} \\
\text{Baş} - \text{ar-} - \text{siz-laş-tur-} - \text{a-} - \text{ma-dık-} - \text{lar-imiz-dan mi-sımız} & \rightarrow \text{(4.3)} \\
\text{head-DS-DS-DS-CAUS-POS-NEG-SUBOR-PL- POSS- ABL Q- 2PL} \\
\text{“Are you one of those whom we could not make unsuccessful?”}
\end{align*}
\]

In (4.2) the suffix causes a loss of vowel in the root, the phonological process known as haplology, and thus changes the root word. This can be considered as a feature of fusional languages. In (4.3) thirteen suffixes (inflectional or derivational) are added to a simple root baş (head) and produce a sentence comprising of a single word. This is again considered as a feature of class other than agglutinating languages, i.e. polysynthetic languages.

As it can be seen in the examples above Turkish is mostly an agglutinating language, a concatenative one to more precise, with rich morphology.
In Birant (2008) 96 derivational suffixes were defined, referring to Adalı (1979) and Uzun (2000). Some of these have not been used productively but a great deal of derivational suffixes is in use in Modern Standard Turkish. In addition to these derivational suffixes Turkish has a rich conjugation and declension systems. There are 15 declension morphemes, mostly due to rich case system, added to nominal categories such as nouns, pronouns and adjectives; and 78 conjugation morphemes added to verbs. Some of these morphemes have the same surface forms and some are zero morphemes, i.e. do not have surface level realizations.

4.2.4 Syntax

Turkish is a subject pro-drop language with the base order SUBJECT-OBJECT-VERB (SOV) (Duman, Aygen, Özgirgin & Bastienne, 2007) which allows scrambling, almost free movement of the constituents of the sentence. It is a subject pro-drop language, i.e. the subjects in Turkish does not have to appear at the surface level. Since the verb carries all the necessary information for determining the subject of a sentence, the subject noun phrases (NP) do not have to appear, in contrast with English which require an overt subject.

Okul -a git-ti -m
School-DAT go-PAST-1SG
‘I went to the school’

In (4.4) the verb git- carries a suffix, {-(I)m}, which indicates the subject is the first person singular. Thus the subject NP does not occur in the sentence overtly.

The base order is SOV; however due its rich case morphology, Turkish allows word orders other than SOV.

Cam -ı kardeş -im kır -d towel -Ø
glass-ACC brother-POSS break-PAST-3SG
‘My brother broke the glass.’
In (4.5) the order of the constituents is changed to object-subject-verb. This is a phenomenon called scrambling. In Turkish NPs have case suffixes such as the accusative case marker in (4.5), {-I}, which signal the relation of that NP to the verb. Thus, the constituents of sentences can be moved almost freely in the sentence.

Turkish is a head-final language, i.e. the heads, the most important constituents of a phrase are positioned at the end, at the rightmost position of a phrase.

In (4.6) there is an NP comprising of the head kız (girl) and güzel (beautiful) an adjective modifying the head. The head is positioned at the final, rightmost position of the phrase. It is the head which carries the necessary information for semantic interpretation. Constituents other than the head, complements or adjuncts, complete the meaning of the head or give extra information about the head. In (4.7) there is an AdjP, headed by an adjective, güzel (beautiful); the other constituent, çok (very), provides additional information about the degree of the adjective. In (4.8) there is a PP, where the head is a postposition, için (for); this postposition is complemented by the noun babam (my father).

4.3 Rule-Based Text Summarization Algorithm for Turkish Language

Although there are numerous studies in Turkish on NLP and text summarization, the tools and resources generated by these studies are not easily accessible. The
algorithms or modules required for automatic text summarization and other NLP tasks, such as morphological analysis, POS-Tagging, etc., are either parts of commercial applications or not freely accessible over the web. Therefore, there is a shortcoming of NLP, IE and IR tools and resources.

Despite this shortcoming, access to some of the tools developed in the theses produced at the Department of Computer Engineering at Dokuz Eylül University is available. The Msc. And Ph.D. theses produced in this department have generated many NLP tools; however these are disparate studies and require a fair amount of effort to combine into a coherent algorithm. Also the studies conducted by Dokuz Eylül University Natural Language Processing Group (DEUNLPG) contributed the tools generated by these theses.

The shortcoming of ready resources such as a machine readable lexicon, a WordNet, etc. has diverted us towards developing a rule-based system. Also the linguistic support provided by the linguists in the DEUNLPG had us to decide this way.

The rule-based text summarization system developed in this thesis, like other automatic text summarization systems comprises of three phases: the pre-processing phase, the processing phase, and a post-processing phase. These phases and the sub-phases of them will be explained below.

4.3.1 Pre-Processing for Rule-Based Text Summarization in Turkish

The pre-processing phase of rule-based text summarization takes UTF-8 coded text documents and starts with a sentence-boundary detection algorithm. After this step, a tokenizer splits the sentences generated by the previous step into words. A stop word elimination algorithm removes the stop words from the texts. Later a morphological analyzer parses these words into their stems and suffixes. Named entity recognition is left out since there is not any reliable resource for named entities, and there is only one study (Küçük & Yazıcı, 2009) which developed a rule-based named entity recognition
algorithm for news texts; however the use of this algorithm in other genres yields very low results. POS-tagging is also left out since POS-taggers for Turkish are not freely available for use and have different tag sets and rules.

4.3.1.1 Sentence Boundary Detection Algorithm

We used the sentence boundary detection algorithm developed by Aktaş & Çebi (2013). Rule-Based Sentence Detection Method for Turkish (RBSDM) uses a rule list, a rule parser and an abbreviations parser to detect the sentence boundaries.

![Main scheme for RBSDM](image)

The rules for End of Sentence (EOS) are determined by linguists and stored in an XML file. RBSDM first splits the text into paragraphs by placing "\n" at the end of the paragraphs; then it checks for punctuation marks listed in the rule list. Normally this approach should detect the EOS correctly; however, the representation of ordinals, e.g. 1st as 1., the use of full stop for some acronyms such as T.B.M.M reduce the accuracy of the algorithm. In order to overcome this issue, RBSDM uses a list of abbreviations obtained from the Turkish Linguistics Association (TLA) and converted into XML format; this list contains 597 abbreviations.

Another issue in sentence boundary detection is the use of hyphen, “-”, to indicate conversations in written texts. Hyphens, along with some other symbols are also used
in bulleted lists. To solve this ambiguity RBSDM uses a novel approach. It assumes that all bulleted lines belong to the same sentence. Thus, the hyphen or the bulleted lists positioned after a colon “:” or a semicolon “;” are considered as parts of a single sentence.

Karasal iklimin genel özellikleri şunlardır: (4.9)
- yazlar kurak ve ılık geçer,
- yağış miktarı azdır,
- günlük ısı farkı yüksektir.

RBSDM considers the text in (4.9) as a single sentence. The accuracy rate of RBSDM is 99.78%, when the misspelled words are removed from the texts.

4.3.1.2 Tokenizer

The tokenizer takes the sentences from the RBSDM and removes the punctuation marks. Then separates these strings into tokens by considering a string of characters with white spaces at the beginning and at the end as a token.

4.3.1.3 Stop Word Elimination

In this step the algorithm takes the output of the tokenizer and using the stop word list presented in Table 4.2 and removes these words from the output.

<table>
<thead>
<tr>
<th>Table 4.2 Some portion of Turkish stop word list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop words</td>
</tr>
<tr>
<td>altmış</td>
</tr>
<tr>
<td>altu</td>
</tr>
<tr>
<td>ve</td>
</tr>
</tbody>
</table>
4.3.1.4 Morphological Parser

After the text is parsed into its tokens, the morphological parser takes this data and conducts a morphological analysis to find the roots and the suffixes of the tokens. The morphological parser uses a lexicon comprising of the following:

i) A list of roots obtained from the TLA, and checked and revised by linguists in the DEUNLPG.

ii) A list of morphemes, both inflectional and derivational,

iii) A set of tags belonging to the morphemes

iv) A list of rules determining the obligatory, allowed and disallowed combinations of roots and suffixes, written by the linguists in the DEUNLPG.

The morphological parser also has a morphophonemics module which generates the allomorphs of the morphemes, i.e. different surface forms of the morphemes which vary according to phonological environment of the morphemes. For instance, the vowel harmony in Turkish requires a morpheme added to a word with a front vowel in the last syllable to have a surface form with a front vowel as in (4.10).

\[
\text{o\d e\v} – \{l\text{Ar}\} \rightarrow \text{o\d e\v-l\text{er}} \quad (4.10)
\]

homework – PL homework-PL

‘homeworks’

The list of roots was obtained from the Turkish Linguistics Association and the list was checked for accuracy with regard to the daily use of modern Turkish. For instance the word yeşil (green) was not included in the original root list, because TLA considers this word as derived as yaş (fresh) + íl (derivational morpheme). However, this root+derivational morpheme combination is an archaic one. It is impossible to track yeşil to yaş today, and thus yeşil should be considered as a root. The root list contains 20,138 roots.
The list of morphemes were obtained by reviewing the literature on Turkish Linguistics. The main resources were the books “Türkiye Türkçesinde Biçimbirimler” (Morphemes in Modern Turkish) by Oya Adalı (Adalı, 1979) and “Biçimbilim: Temel Kavramlar” (Morphology: Basic Concepts) by Nadir Engin Uzun (Uzun, 2000). There are total 174 morphemes, 96 derivational and 78 inflectional in the list of morphemes. Table 4.3 presents some of the morphemes in the list, with their tags and explanations.

Table 4.3 Morphemes, tags and explanations

<table>
<thead>
<tr>
<th>Morpheme</th>
<th>Tag</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>{-lAr}</td>
<td>DuASayC</td>
<td>Nominal Number - Plural</td>
</tr>
<tr>
<td>{-(y)A}</td>
<td>DuADurYon</td>
<td>Dative Case</td>
</tr>
<tr>
<td>{-(A, I)r}</td>
<td>DuEZGen</td>
<td>Aorist</td>
</tr>
<tr>
<td>-lik</td>
<td>TBAA:lik</td>
<td>Derivational Suffix – Noun to Noun</td>
</tr>
</tbody>
</table>

The tags are developed following the scheme in (4.11):

\[
\begin{align*}
DU & \rightarrow \text{Dilbilgisel Ulam (Grammatical Category)} \\
A & \rightarrow \text{Ad (Noun)} \\
Dur & \rightarrow \text{Durum (Case)} \\
Yon & \rightarrow \text{Yönelme (Dative)}
\end{align*}
\]

The list of rules determines the root-suffix and suffix-suffix combinations. The rules include “must rules”, “allowed rules”, and “disallowed rules”.

\[
E, \text{ DuEK/DuEG/DuEZ}
\]

The rule in (4.12) indicates that a verb (E) must be followed by any of the morphemes in the DuEK (Grammatical Category, Verb, Mood), DuEG (Grammatical Category, Verb, Aspect) or DuEZ (Grammatical Category, Verb, Tense). This rule is a must rule because verbs do not appear, other than very limited cases, in their bare forms in the sentence. Thus they have to carry at least one Tense, Aspect or Mood morpheme. Also, the personal agreement suffixes cannot be added to the verb root.
An allowed rule indicates that a morpheme can follow the root or another suffix.

\[
A, \text{TBAE} \quad (4.13)
\]

The rule in (4.13) asserts that a noun (A) can be followed by a derivational suffix which derives a verb from a noun.

A disallowed rule indicates that a morpheme cannot follow a root or another suffix.

\[
A, \text{DuE/TBEE/TBEA/TBSA/TBSE} \quad (4.14)
\]

The rule in (4.14) asserts that a noun (A) cannot be followed by any of the morphemes in DuE category (Grammatical Category, Verb), TBEE (derivational suffixes which derive verbs from verbs), TBEA (derivational suffixes which derive nouns from verbs), TBSA (derivational suffixes which derive noun from adjectives) or TBSE (derivational suffixes which derive adjectives from verbs).

The algorithm, presented in Figure 4.2, starts with analyzing tokens, after tokenization and stop word removal.

i) It takes one token, a character string between two white spaces, and starts the checks the leftmost character against the entries in the root list. If the character string matches to a root in the list, algorithm records this as a possible root in the first root record.

ii) Then the algorithm starts checking the remaining character string against the morpheme list. If a part of the remaining character string matched with any morphemes the algorithm adds this to another record, the first suffix record. The procedure continues with the remaining characters and checks them against the list of morphemes and the first suffix record. If the second identified morpheme is not compatible with the one in the suffix record, the algorithm adds another character to the string in the first morpheme and starts the second step again.
iii) Then the suffix record is checked against the root record, to affirm whether the suffixes in the record can follow the root in the root record.

iv) If the suffix combination is not allowed to follow the root, then the algorithm eliminates the first root record and adds the next character in the string and checks it against the root list.

v) This procedure continues until all possible roots and suffix combinations are tried.

The procedure explained above can be exemplified as in Table 4.4:

Table 4.4 Morphological parser output

<table>
<thead>
<tr>
<th>Token: okumam</th>
<th>Possible root</th>
<th>Possible Suffixes</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>O he</td>
<td>kumam</td>
<td>Invalid suffix combination</td>
<td></td>
</tr>
<tr>
<td>ok arrow</td>
<td>umam</td>
<td>Invalid suffix combination</td>
<td></td>
</tr>
<tr>
<td>Oku read</td>
<td>mam</td>
<td>2 Results</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>{-mA} + {m}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: TBEA (derivational suffix that derives nouns from verbs) + First person singular possessive suffix</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: Negative suffix + First Person Singular Agreement Suffix</td>
<td></td>
</tr>
</tbody>
</table>
Figure 4.2 Morphological parser algorithm
CHAPTER FIVE
PROCESSING FOR RULE-BASED TEXT SUMMARIZATION IN TURKISH

5.1 Introduction

After the preprocessing phase rule-based text summarization system initiates the processing phase. The processing phase, just like the pre-processing phase, has some resources, such as a semantic relations network comprising of synonyms, quasi-synonyms and antonyms of the roots. This module was developed using NLP techniques on the Turkish Dictionary of TLA. The processing phase also has a list of meta-discourse markers list. Meta-discourse refers to the authors’ linguistic choices in a text. Meta-discourse can be realized through a range of linguistic devices from punctuation and typographic marks (such as parentheses to signal clarifications or underlining to mark emphasis), to whole clauses and sentences (Hyland, 1998). The meta-discourse marker list is presented in Table 5.1. However, the algorithm developed in this thesis does not use typographic marking as meta-discourse marker, because the system can process only UTF-8 coded plain text documents which are stripped from font styling.

Table 5.1 Some of the meta-discourse markers

<table>
<thead>
<tr>
<th>Meta-Discourse Markers</th>
<th>As a result</th>
<th>Thuse</th>
<th>Then</th>
<th>Therefore</th>
<th>Thus</th>
<th>Consequently</th>
<th>Because of this</th>
<th>Because of this reason</th>
<th>In other words</th>
<th>For that purpose</th>
<th>Because</th>
<th>Because</th>
</tr>
</thead>
<tbody>
<tr>
<td>sonuçta</td>
<td>As a result</td>
<td>Thuse</td>
<td>Then</td>
<td>Therefore</td>
<td>Thus</td>
<td>Consequently</td>
<td>Because of this</td>
<td>Because of this reason</td>
<td>In other words</td>
<td>For that purpose</td>
<td>Because</td>
<td>Because</td>
</tr>
<tr>
<td>böylece</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o zaman</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bu nedenle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>böylelikle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sonuç olarak</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bundan ötürü</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bu nedenden dolayı</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yani</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bu amaçla</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>çünkü</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zira</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The algorithm starts with obtaining keywords from the user. If the user enters keywords, the algorithm first sends these to the morphological parser and determines...
the roots and suffixes. If the user does not enter keywords, the algorithm seeks for the
title of the document. The title of the document is sent to the pre-processing phase and
the tokens are used as keywords. If there is not any title for the text document, the
algorithm runs a frequency search on the annotated tokens and the most frequent 5
words are taken as keywords.

The algorithm then determines synonyms, quasi-synonyms and antonyms of these
keywords to check them against the annotated tokens in the pre-processed text
document. Then the algorithm ranks the matches between keywords and in-text tokens
and gives points to each occurrence of the keywords, their synonyms, antonyms and
quasi-synonyms.

The algorithm then searches for meta-discourse markers in the sentences and
increases the ranks of the sentences which contain a meta-discourse marker.

The sentences are also ranked with regard to their positions in the text. The
algorithm determines the positions of the sentences, i.e. the position of the paragraphs
in the text and the position of the sentence in the paragraph. The first and last
paragraphs are considered more important and the sentences in these paragraphs are
ranked higher. Also the first and the last sentences of the paragraphs are considered
more important and ranked respectively.

The algorithm then orders the sentences in terms of their ranking score and selects
the sentences with higher rankings for summary, and sends them to the post-processing
module. Figure 5.1 gives the algorithm of the automatic summarizer developed in this
thesis.
Figure 5.1 Automatic text summarization algorithm for Turkish
5.2 Post-Processing for Rule-Based Text Summarization in Turkish

The post-processing phase only comprises of a sorter algorithm which sorts the sentences selected for summary in terms of their original position in the input text. The processing algorithm orders the sentences in a descending order in terms of their ranks. However, this could impair the coherence of the summary generated. In order to achieve this problem, the sentences are sorted following their original positions in the text. The post-processing approaches explained in Section 3.4 are not used in this thesis, because developing a syntactic parser for Turkish which has an almost free word order, in addition to its rich morphology and intricate relationship between morphology and syntax is a task that transcends beyond the scope and limitations of this thesis.

5.3 The User Interface

The algorithm explained in Section 5.2 runs as a web service on a web server. The graphical user interface, presented in Figure 5.1 allows users to input their texts, titles of the texts, and keywords, if available.

Figure 5.2 The GUI of the automatic text summarization software
5.3.1 Adding Text Files

In the graphical user interface, the user starts by adding a UTF-8 coded plain text file by clicking the select file button and then selecting the file from the browser window.

Figure 5.3 A screenshot of a sample text file

5.3.2 Entering the Title of the Text

In the next step the user may enter the title of the text. If any title is not entered in the input box in the graphical user interface, the software uses the word frequencies to determine the most important concepts in the text files.

Figure 5.4 A screenshot of the GUI after title input
5.3.3 Entering Keywords

Following the input of title of the text to be summarized, the user may enter keywords in the keyword input box in the graphical user interface.

![Automatic Text Summarization for Turkish](image)

Figure 5.5 A screenshot of the GUI after keyword input

5.3.4 Raw and Processed Text Data

After entering the keywords, the software shows the input text in another text box without any processing on it to the user as a control measure, to allow the user to check whether the selected file is the right one. Yet in another text box the software shows the text parsed into sentences in XML format.

![Text parsed into sentences](image)

Figure 5.6 A screenshot of the GUI showing raw and processed text data
5.3.5 Sentence Ranking

The graphical user interface later shows the sentence ranking, with rank points placed to the left of the sentences, with regard to the text as a whole and with regard to the paragraphs. The algorithm ranks the sentences and then lists the sentences in an ascending order from the highest ranking to the lowest ones. The graphical user interface also shows sentence ranking with regard to paragraphs. The algorithm ranks the sentences in the paragraphs then sends this data to the graphical user interface. And the user interface shows sentence ranking in an ascending order for each paragraph in the text, starting from the first paragraph.

![Sentence Ranking (Whole Text)](image1)
![Sentence Ranking (Paragraph)](image2)

Figure 5.7 A screenshot of the GUI showing sentence rankings

5.3.6 Summary Output

The algorithm send the highest ranked sentences to the post-processing module to sort these sentences according to their original positions in the text. When the sentences with the highest ranked are combined without considering the original positions, the generated summary would be incohesive and difficult to understand. Thus the algorithm sorts the highest ranked sentences to generate a more cohesive and understandable text.

1. **8-Odev, belli bir konu veya ünite ile ilgili olarak öğrencilerden yapmaları istenen zihinsel veya bedensel çalışmalarla denir.**
2. **8-Öte yandan, bu çalışmada geliştirilen tutum ölçüğü, Türkiye’nin değişik yörelerinde okumaktadır, yetişimleri farklı öğrencilere uygulanarak ölçülür, madde yapısı tekrar incelenebilir.**
3. - Tutum ölçeğinde yer alan maddelerin tümünün birinci faktördeki yük değerinin 0.30'un üzerinde olması ve tek faktörle açıklanan varyans miktarının %55'e ulaşması, ölçeğin tek faktörlü olarak yorumlanabileceği göstermektedir.

4. - Ölçek, likert tipi beşli derecelene ölçekte hazırlanmış 17 maddeden oluşmuştur.

5. - Eskişehir Fatih Fen lisesinde okuyan öğrenciler üzerinde gerçekleştilen bu araştırmının bulgularına göre, ev ödevine karşı öğrenci tutumlarını belirlemek üzere hazırlanan tutum ölçeği, öğrencilerin bu konudaki tutumlarını ölçmeye yarayan tek boyutlu güvenilir ve geçerli bir ölçme aracıdır.

As it can be seen in Figure 5.8, when sentences are listed according to their ranks the generated summary is not cohesive, i.e. it is difficult the links between sentences, thus to follow the meaning of the text. However, when the sentences are sorted in term of their original positions in the text, as in, the generated summary becomes more cohesive and understandable.

Figure 5.8 Output of the software, sentences not ordered in terms of their original position in the text

Figure 5.9 Output of the software, sentences ordered in terms of their original position in the text

If the reference summary, written by the author of the text (presented in Figure 5.10), is compared to the output of the automatic summarizer, it can be seen that the summary generated by the automatic summarizing algorithm for Turkish, contains more information about the text than the reference summary.

Figure 5.10 Reference summary, the summary written by the author of the text
CHAPTER SIX
EVALUATION

6.1 Introduction

A critical task in developing an automatic text summarization is the evaluation of the system performance in various ways. The system performance may be evaluated in terms of speed of the operation as a whole, or of the operations one by one, in the automatic summarization system. Such an evaluation would be used to improve the use of resources more efficiently; however it would not improve the quality of the summaries generated by the system.

The purpose of this chapter is not to perform a speed test for the software but to evaluate the quality of the summaries generated by the automatic text summarization algorithm for Turkish.

There are numerous methods for evaluating automatic text summarization, and an abundant literature on automatic summarization evaluation techniques. It would be beyond the scope of this thesis to address all of these methods; therefore only two of the methods used in the literature will be elaborated here and applied to test data. One automatic evaluation method will be used since human evaluation is considered labor-intensive and sometimes inconsistent (Liu & Liu, 2009). In addition to this automatic evaluation method a human evaluation method developed by Liu & Liu (2009) will be adapted to the aims of this study.

6.2 ROUGE-N: Automatic Evaluation of Summaries

ROUGE is the acronym for “Recall-Oriented Understudy for Gisting Evaluation”. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans (Lin, 2004). ROUGE-N, on the other hand is a N-gram co-occurrence statistics used to automatically evaluate
the quality of summaries. Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries (Lin, 2004). The formula for calculating ROUGE-N score is as follows:

$$\text{ROUGE - N} = \frac{\sum_{\text{Se}(\text{ReferenceSummaries})} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{Se}(\text{ReferenceSummaries})} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}$$

(6.1)

Here, $n$ is the length of the n-gram, and $\text{Count}_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurring in a candidate summary, i.e. the summary generated by the automatic text summarization algorithm and a reference summary, i.e. the summary written by a human.

### 6.3 Human Evaluation

Liu & Liu (2009) developed a human evaluation experiment setup for comparing human summaries and automatic text summarization algorithm summaries. They created a survey comprising of two sections for evaluating meeting summaries. The first section includes nine statements about the summaries. The second section includes two statements about the disfluencies in the summaries and their effects on their evaluation. The statements in Liu & Liu’s survey is presented in Figure 6.1:

<table>
<thead>
<tr>
<th>Section 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1: The summary reflects the discussion flow in the meeting very well.</td>
</tr>
<tr>
<td>S2: Almost all the important topic points of the meeting are represented.</td>
</tr>
<tr>
<td>S3: Most of the sentences in the summary are relevant to the original meeting.</td>
</tr>
<tr>
<td>S4: The information in the summary avoids redundancy.</td>
</tr>
<tr>
<td>S5: The relationship between the importance of each topic in the meeting and the amount of summary space given to that topic seems appropriate.</td>
</tr>
<tr>
<td>S6: The relationship between the role of each speaker and the amount of summary speech selected for that speaker seems appropriate.</td>
</tr>
<tr>
<td>S7: Some sentences in the summary convey the same meaning.</td>
</tr>
<tr>
<td>S8: Some sentences are not necessary (e.g., in terms of importance) to be included in the summary.</td>
</tr>
<tr>
<td>S9: The summary is helpful to someone who wants to know what is discussed in the meeting.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section 2</th>
</tr>
</thead>
</table>
The disfluencies affect the readability and your comprehension of the summary very much in your evaluation.

S10: The disfluencies affect the readability and your comprehension of the summary very much in your evaluation.

S11: The disfluencies affect very much the scores you give to the evaluation statements.

Figure 6.1 Survey for human evaluation of summaries (F. Liu & Y. Liu, 2009)

F. Liu & Y. Liu (2009) directed the statements in Figure 6.2 to five participants and requested them to rate the statements using a Likert-scale from 1 to 5, according to the extent of agreement. Following F. Liu & Y. Liu (2009), a similar survey is created for human evaluation of the summaries generated by the automatic text summarization algorithm for Turkish developed in this thesis. This survey is comprised of two sections. The first include seven section questions the quality of the summaries; and the second section include two questions questioning whether the problems in the summaries affected the participants’ evaluations. The statements of the survey to be used in this thesis are presented in Figure 6.2:

**Section 1**

S1: The summary reflects the topics in the original text very well.
S2: Almost all the important topic points of the original text are represented.
S3: Most of the sentences in the summary are relevant to the original text.
S4: The information in the summary avoids redundancy.
S5: Some sentences in the summary convey the same meaning.
S6: Some sentences are not necessary (e.g., in terms of importance) to be included in the summary.
S7: The summary is helpful to someone who wants to know what is written in the original text.

**Section 2**

S8: The disfluencies affect the readability and your comprehension of the summary very much in your evaluation.
S9: The disfluencies affect very much the scores you give to the evaluation statements.

Figure 6.2 Statements in the automatic text summarization evaluation survey

Using the statements in Figure 6.2, survey form comprising of five-level Likert items is created (1-strongly disagree, 2-disagree, 3-neutral, 4-agree, 5-strongly agree). The survey form (Appendix A) is in Turkish since it is planned to be applied Turkish native speakers.
6.4 Data Set

Evaluation of the automatic text summarization algorithm for Turkish, developed in this thesis, is conducted using a data set comprising of five text documents. Only five documents are selected for the data set, because human evaluation is a time consuming process for participants. Three scientific articles from various disciplines, and two news texts are included in the data set. All of these text document have summaries generated by their respective writers. These summaries are used as reference summaries to make comparisons with the candidate summaries, i.e. the ones generated by the automatic text summarization algorithm. Table 6.1 provides information about the data set used in the evaluation process of the automatic text summarization algorithm developed in this thesis.

<table>
<thead>
<tr>
<th>No</th>
<th>Genre</th>
<th>Number of Words (NOW)</th>
<th>Number of Sentences (NOS)</th>
<th>Number of Paragraphs (NOP)</th>
<th>NOS in the reference summary</th>
<th>NOS in the candidate summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text1</td>
<td>Scientific Article</td>
<td>3697</td>
<td>194</td>
<td>34</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Text2</td>
<td>Scientific Article</td>
<td>3416</td>
<td>179</td>
<td>15</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Text3</td>
<td>News Article</td>
<td>233</td>
<td>13</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Text4</td>
<td>News Article</td>
<td>141</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Text5</td>
<td>Scientific Article</td>
<td>1735</td>
<td>75</td>
<td>30</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

6.5 ROUGE Score of the Automatic Text Summarization Algorithm

The ROUGE-N score for the automatic text summarization algorithm developed here is found by using the formula in 6.1 by placing in the formula the number of n-grams of the reference summaries, i.e. the summaries written by the authors of the texts in the data, and the number of number n-grams that match in the reference and candidate summaries, i.e. the summaries generated by the automatic text summarization algorithm.

ROUGE-N metrics are calculated separately for each summary in the data set. The number of n-grams in the reference summaries is calculated first; and then the number of matching n-grams is found. Then these values are placed in the formula and the
calculation is completed. This procedure is repeated for unigrams, bigrams, trigrams and four-grams for all summary pairs in the data set. The ROUGE-N metrics are basically recall metrics, higher ROUGE scores would indicate that the algorithm returned most of the sentences that are relevant to the reference summaries.

When the ROUGE-N metrics for all the texts in the data set are calculated, the following tables are obtained.

Table 6.2 ROUGE-N metrics for text1

<table>
<thead>
<tr>
<th>Text1</th>
<th>Match</th>
<th>Total</th>
<th>ROUGE Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigram</td>
<td>53</td>
<td>153</td>
<td>0.2746</td>
</tr>
<tr>
<td>bigram</td>
<td>12</td>
<td>197</td>
<td>0.0609</td>
</tr>
<tr>
<td>Trigram</td>
<td>7</td>
<td>205</td>
<td>0.0341</td>
</tr>
<tr>
<td>Four-gram</td>
<td>6</td>
<td>207</td>
<td>0.0289</td>
</tr>
</tbody>
</table>

Table 6.3 ROUGE-N metrics for text2

<table>
<thead>
<tr>
<th>Text2</th>
<th>Match</th>
<th>Total</th>
<th>ROUGE Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigram</td>
<td>56</td>
<td>179</td>
<td>0.3128</td>
</tr>
<tr>
<td>bigram</td>
<td>13</td>
<td>246</td>
<td>0.0528</td>
</tr>
<tr>
<td>Trigram</td>
<td>8</td>
<td>250</td>
<td>0.0320</td>
</tr>
<tr>
<td>Four-gram</td>
<td>7</td>
<td>251</td>
<td>0.0278</td>
</tr>
</tbody>
</table>

Table 6.4 ROUGE-N metrics for text3

<table>
<thead>
<tr>
<th>Text3</th>
<th>Match</th>
<th>Total</th>
<th>ROUGE Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigram</td>
<td>8</td>
<td>16</td>
<td>0.5000</td>
</tr>
<tr>
<td>bigram</td>
<td>5</td>
<td>16</td>
<td>0.3125</td>
</tr>
<tr>
<td>Trigram</td>
<td>3</td>
<td>15</td>
<td>0.2000</td>
</tr>
<tr>
<td>Four-gram</td>
<td>3</td>
<td>14</td>
<td>0.2142</td>
</tr>
</tbody>
</table>

Table 6.5 ROUGE-N metrics for text4

<table>
<thead>
<tr>
<th>Text4</th>
<th>Match</th>
<th>Total</th>
<th>ROUGE Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigram</td>
<td>7</td>
<td>23</td>
<td>0.3043</td>
</tr>
<tr>
<td>bigram</td>
<td>5</td>
<td>22</td>
<td>0.2272</td>
</tr>
<tr>
<td>Trigram</td>
<td>4</td>
<td>21</td>
<td>0.1904</td>
</tr>
<tr>
<td>Four-gram</td>
<td>4</td>
<td>20</td>
<td>0.2000</td>
</tr>
</tbody>
</table>
Table 6.6 ROUGE-N metrics for text5

<table>
<thead>
<tr>
<th></th>
<th>Match</th>
<th>Total</th>
<th>ROUGE Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigram</td>
<td>24</td>
<td>86</td>
<td>0.2790</td>
</tr>
<tr>
<td>bigram</td>
<td>15</td>
<td>100</td>
<td>0.1500</td>
</tr>
<tr>
<td>Trigram</td>
<td>10</td>
<td>100</td>
<td>0.1000</td>
</tr>
<tr>
<td>Four-gram</td>
<td>7</td>
<td>99</td>
<td>0.0707</td>
</tr>
</tbody>
</table>

When all the tables are examined in combination, it can be seen that the algorithm returned better results for Text3 and Text4. These texts are short news articles and the language used in these texts is close to the daily spoken language without intricate grammatical structures and special terminology. On the other hand Text1, Text2, and Text5 are scientific articles and they contain bulleted lists, formulaic expression, special terminology and longer sentences. The average sentence length for Text3 and Text4 is 17 words per sentence, while it is 20 words per sentence in Text1, Text2, and Text3. Scientific articles may also contain borrowed words which could not be easily analyzed by the morphological parser.

The results obtained after the ROUGE metrics analysis are similar to the studies in the literature (Wang, Stokes, Doran, Newman, Carthy & Dunnion, 2005). However, in order to better evaluate the summaries generated by the automatic text summarization algorithm in this thesis, more analyses should be conducted using larger data sets and perhaps using other automatic text summarizers for comparison.

### 6.6 Human Evaluation of the Automatic Text Summarization Algorithm

ROUGE metrics are intrinsic and quantitative, i.e. they use the text data and recall scores. However, the results of this analysis can be misleading, because high rates of matching between the linguistic units between two documents do not always mean a better summary. In order to conduct a qualitative analysis of the summaries generated by the automatic text summarization algorithm, a survey is applied to 10 participants (n=10), students of the Department of Linguistics Department in Dokuz Eylül University. These participant are selected for evaluation since they have a better understanding of language in general and Turkish in specific due to their training in
linguistics. The participants have read the original texts and the relevant summaries, generated by the algorithm and evaluate them. The participants evaluate the summaries by rating the extent of agreement with the statements in the survey form. The results obtained from the participants are examined and the averages for each text and for each statement are calculated. When the results are examined closely it can be seen that, the average scores for Statement1 (The summary reflects the topics in the original text very well.) is very high, i.e. the participants strongly agreed with the statement. Especially in Text3 and Text4, it is seen that the summary generated by the algorithm has been evaluated as reflecting the topics in the original text very well. Considering these texts are short news articles, the results overlap with the results of the ROUGE metrics.

Statement2 (Almost all the important topic points of the original text are represented.) has got different averages for different texts. Scientific articles in the data set are long texts with complex grammatical structures and terminology. Averages of Statement3 (Most of the sentences in the summary are relevant to the original text) indicate that the algorithm generates summaries which contain sentences relevant to the original text.

Statement4 (The information in the summary avoids redundancy) is an indicator of originality that the summary does not contain non-relevant sentences. Statement5 (Some sentences in the summary convey the same meaning.) averages show that the summaries generated by the algorithm do not contain sentences with similar meanings, which is the case for most scientific texts, since they contain similar sentences in the introduction, discussion and conclusion sections.

Statement6 scores (Some sentences are not necessary (e.g., in terms of importance) to be included in the summary) indicate that, especially in scientific articles, some sentences are not necessary for the summaries, considering their significance for the topics mentioned in the original text. However, this score is very low for news articles.
The scores for Statement7 (The summary is helpful to someone who wants to know what is written in the original text) is very high, indicating that the summaries generated by the algorithm fulfill their purpose and inform the participant about what is said in the text document. This result can be considered the most important one, since automatic text summarization aims at providing understandable and comprehensible compressed versions of the original text. Statement8 and Statement9 are about the self-evaluation of the participants for evaluating the effects of the summary to their judgment about it. The scores for both statements are low indicating that either there are not any disfluences in the summaries or they are not so significant to affect the decisions of the participants.

Table 6.5 Average Scores for the statements in the survey form

<table>
<thead>
<tr>
<th></th>
<th>Text1</th>
<th>Text2</th>
<th>Text3</th>
<th>Text4</th>
<th>Text5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement1</td>
<td>4.5</td>
<td>4.1</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Statement2</td>
<td>4.3</td>
<td>2.8</td>
<td>4.6</td>
<td>4.4</td>
<td>2.6</td>
</tr>
<tr>
<td>Statement3</td>
<td>4.3</td>
<td>3.7</td>
<td>4.4</td>
<td>4.3</td>
<td>3.4</td>
</tr>
<tr>
<td>Statement4</td>
<td>3.5</td>
<td>4.1</td>
<td>5</td>
<td>4.8</td>
<td>3.9</td>
</tr>
<tr>
<td>Statement5</td>
<td>2.1</td>
<td>2.5</td>
<td>1</td>
<td>1</td>
<td>2.4</td>
</tr>
<tr>
<td>Statement6</td>
<td>3.6</td>
<td>3</td>
<td>1.3</td>
<td>1.3</td>
<td>3.1</td>
</tr>
<tr>
<td>Statement7</td>
<td>4.6</td>
<td>4.3</td>
<td>5</td>
<td>5</td>
<td>4.4</td>
</tr>
<tr>
<td>Statement8</td>
<td>1.5</td>
<td>1.8</td>
<td>1</td>
<td>1</td>
<td>1.9</td>
</tr>
<tr>
<td>Statement9</td>
<td>1.2</td>
<td>1.4</td>
<td>1</td>
<td>1</td>
<td>1.5</td>
</tr>
</tbody>
</table>

The results obtained in human evaluation of the summaries generated by automatic text summarization algorithm indicate that the algorithm is a promising one. However, conducting analyses with five text documents and 10 participants is not enough to be conclusive. There is more to be achieved, yet more to be tried. Again, as it is mentioned in Section 6.5, the analyses should be repeated with larger data sets with more participants, and comparisons should be made with the summaries generated by other automatic text summarization algorithms to provide a clearer picture of where this algorithm stands.
CHAPTER SEVEN
CONCLUSION

Automatic text summarization has become a popular research topic recently due to the excessive generation and distribution of data over the Internet and the need for processing this huge amount of data for commercial, political and scientific purposes. Turkish also has its share, albeit very late compared to other languages, from this trend towards natural language processing and automatic text summarization. Although there are studies and tools for NLP, IE, IR and ATS, the resources are still sparse. Annotated corpora are limited and resources such as WordNet, FrameNet, VerbNet are not present or not accessible freely.

In this context, this thesis was an attempt to develop an automatic text summarization tool for Turkish. First an elaborate definition of what a summary or abstract is provided, then a brief historical perspective on automatic text summarization in the world and in Turkey, especially is given. After explaining the text summarization methodology with its pre-processing, processing and post-processing phases, the preprocessing phase for an automatic text summarization tool for Turkish is introduced. Following a detailed account of Turkish language, a robust morphological parser, developed by Birant (2008) and improved towards in this thesis, is introduced. Then the automatic text summarization algorithm are and its graphical user interface is developed. The evaluation of the algorithm developed in this thesis is conducted via ROUGE-N analysis using unigrams, bigrams, trigrams and four-grams. In addition to this quantitative analysis, a qualitative analysis depending on human evaluation is conducted. It is seen in the analysis that the algorithm produce summaries similar to the original summaries written by authors of the texts.

This thesis can be considered as an achievement of its goal of developing a rule-based automatic text summarization tool for Turkish. The question “what is the advantage of the algorithm developed in this thesis among others?” may come into mind. It can be argued that it has some advantages:
i. The resources used in the algorithms, such as the root list, morpheme list, rule list, meta-discourse marker list, stop word list, are either generated or checked by linguists.

ii. The morphological parser which was developed in Birant (2008), but used here with improvements is another asset. It was developed and improved with regard to the latest studies in the linguistics literature and the computer science literature.

iii. The automatic text summarization algorithm uses meta-discourse markers, in addition to other linguistic clues, to rank the sentences. This is a novel approach for automatic text summarization in Turkish.

The summaries generated by the algorithm were evaluated both quantitatively via ROUGE-N metrics and qualitatively via human evaluation using a survey form. The results of the analyses were promising; they indicated that the summaries generated by the algorithm were similar to the reference summaries, i.e. the summaries written by the authors of the corresponding texts. The human evaluation showed that the automatic summaries were cohesive and understandable and did not include unnecessary or irrelevant sentences. These analyses should be conducted again and again to improve the performance of the algorithm.

It can be argued that these are important achievements; however there is much progress to be made. This rule-based text summarization system for Turkish could be improved by providing support for text documents containing styling data such as fonts and colors. Also a robust syntactic parser would enable a deep parsing of the text and would give way to better post-processing, such as paraphrasing, sentence fusion or sentence compression.

Automatic Text Summarization is a complex process which require knowledge and experience from different areas of research such as computer science, linguistics, and statistics. A single researcher could not be expected to master all three areas of research, thus automatic text summarization studies, and natural language processing
in general, should be studied with an interdisciplinary perspective avoiding the exclusion of experts of other areas from the studies.

Lastly, it can be argued that the algorithm developed in thesis can be adapted to other Turkic languages such as Kyrgyz, Uzbek, Kazakh, etc. Since the algorithm is a rule based one, comprising of rule-based resources such as roots list, suffix list, rule list, stop-word list, macrostructure marker list, etc. In case these resources are provided, the algorithm can easily be adapted to the aforementioned languages.
REFERENCES


APPENDICES

APPENDIX A – SURVEY FORM


Sormacaya katkıdınız için teşekkür ederiz.

Araş, Gör. Çağdaş Can BIRANT
Dokuz Eylül Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı

A-KATILIMCI BİLGİLERİ

1. Cinsiyetiniz: □ Kadın □ Erkek
2. Yaşınız:
3. Mesleğiniz:

Sormacının İkinci Bölümü Diğer Sayfadadır
APPENDIX - A SURVEY FORM (CONTINUED)

RÖZET DEĞERLENDİRME

Metin No: 1

1. Özet özgün metinde yer alan konuların önemli çoğunlukla iyi bir biçimde yansıtmaktadır.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

2. Özgün metindeki hemen hemen bütün önemli konular özette yer alıyor.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

3. Özetteki cümlelerin çoğun özgün metinde ilgili.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

4. Özette tekrarlanan bilgi yok.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

5. Özetteki bazı cümleler aynı anlama taşıyor.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

6. Özetteki bazı cümlelerin özete eklenmesi gerekmez (örneğin, önem açısından).
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

7. Özet, özgün metinde sadece bahsedildiğiini anlamada yardımcı oldu.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

C-ÖZ. DEĞERLENDİRME

1. Özette akıcılığı bozandan ifadeler okunabilirliği ve anlaşılabilirliği büyük ölçüde etkiliyor.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

2. Özette akıcılığı bozandan ifadeler İkinci Bölümdeki özet değerlendirme puanlarını büyük ölçüde etkiledi.
   ○ Kesinlikle katlıyorum  ○ Katlıyorum  ○ Fikrim yok  ○ Katılmıyorum  ○ Kesinlikle katılmıyorum

Sormacaya katkıdığımız için teşekkür ederiz.
Okumanın en temel amacı okuduğunu anlamaktır. Okuduğunu anlam düzeyini artırmak için bazı stratejiler kullanabilir. Bu stratejilerden biri de özetlemedir. Özetleme, okuma sonrasında okuyucunun metindeki ana fikri bulması, gereksez ayrıntıları çıkarması, metnin yapısını ve düşünce akışını bozmadan kendi kelime ve cümleleriyle metni kısaltmasıdır. Okuyucu orijinal metin üzerinde bazı stratejiler kullanarak özet yapmalıdır.


APPENDIX C – CANDICATE SUMMARY 1

APPENDIX D – REFERENCE SUMMARY 2

Yabancı dil öğretiminde dil öğrenicilerinin dilsel ve iletişimSEL becerilerinin geliştirilmesinde sözcük öğretiminin payı büyüktür. Öğrencilerin anlama becerilerinin gelişmesinde de sözcükler önemli bir yere sahiptir. Çünkü sözcüklerin anlamlandırılması sayesinde etkili ve amaca yönelik iletişim sağlanabilmekte, bir bağlam ancak hedef sözcüklerin bilinmesi durumunda kolaylıkla anlaşılabilirmektedir.


APPENDIX E – CANDIDATE SUMMARY 2

Muğla'nın Fethiye İlçesi'nin il olması amacıyla 38 sivil toplum kuruluşunun desteğiyle 60 bin imza toplandı.

Radikal Gazetesi accessed on: 10.03.2015 from: http://www.radikal.com.tr/turkiye/82_il_mi_gelivor-1310147
Fethiye Belediyesi'nin önderliğinde ilçedeki 38 sivil toplum kuruluşunun desteğiyle geçen 23 Şubat'ta başlatılan 'Fethiye İl Olsun' kampanyasına destek her geçen gün artıyor.
APPENDIX H – REFERENCE SUMMARY 4

Halk TV'de Baransu'nun tutuklanması konusundaki yorumuna moderatör Ruhat Mengi'nin gösterdiği tavır yüzünden sınırlenen Metin Feyzioğlu canlı yayını terk etti.

Cumhuriyet Gazetesi accessed on: 08.03.2015 from: http://www.cumhuriyet.com.tr/haber/turkiye/229243/Metin_Feyzioglu_canli_yayini_terk_etti.html
Halk Tv'ye Ruhat Mengi ile Her Açıdan programına konuk olan Türkiye Barolar Birliği Başkanı Metin Feyzioğlu, Moderatör Ruhat Mengi tarafından sözünün kesilmesinden rahatsız olarak canlı yayını terk etti.
Günümüzde nitelikli işgücü yetiştirmeye yönelik arayışların en son güncel ürünlerinden biri olan Aktif Eğitim modelinin belkemiğini oluşturan Probleme Dayalı Öğrenim (PDÖ) oturumları çağdaş bilgi toplumunun gerektirdiği nitelikli birey ve işgücünü yetiştirmede etkili bir model olabileceğinin ipuçlarını vermeye başlamıştır. Ancak bu umut verici öğrenim yönteminin, daha etkili olabilmesi için yalnızca yüz yüze ortamlarda gerçekleşmesi gibi bir kısıtlamayı aşmak gerekmektedir. Bu amaç doğrultusunda bu çalışmada PDÖ oturumlarının sanal ortamda nasıl geçireştirilebileceğine ilişkin bir uygulama geliştirilmeye çalışılmıştır. Geliştirilen sanal-PDÖ henüz uygulama fırsatı bulamasa ve bazı noktalarda eksiklikleri bulunsa da getirdiği birçok artı değer nedeniyle üzerinde çalışmaya ve iyileştirmeye değer bir uygulama olduğunu göstermektedir.