ABSTRACTIVE TEXT SUMMARIZATION FOR MORPHOLOGICALLY RICH LANGUAGES

by

Batuhan Baykara

B.S., Computer Engineering, Bilkent University, 2013

M.S., Computer Science, University of Tampere, 2015

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

Graduate Program in Computer Engineering

Boğaziçi University

2023

ABSTRACTIVE TEXT SUMMARIZATION FOR MORPHOLOGICALLY RICH LANGUAGES

APPROVED BY:

Prof. Tunga Güngör (Thesis Supervisor)	.....
Assoc. Prof. Arzucan Özgür	.....
Prof. Mehmet Fatih Amasyalı	.....
Prof. Fikret Gürgen	.....
Asst. Prof. Rahim Dehkharghani	.....

DATE OF APPROVAL: DD.MM.YYYY

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Tunga Güngör. He has always guided me with great wisdom and care throughout my PhD. His continued support and understanding were one of the most fundamental aspects that have helped me to complete this journey. I am truly grateful to have worked with such a great supervisor and would like to thank him for all the effort he has put in me and this thesis.

I would like to thank my thesis committee members Assoc. Prof Arzucan Özgür and Prof. Mehmet Fatih Amasyalı for their valuable advice that have helped me to improve my research and come up with new ideas. Moreover, I would like to thank Prof. Fikret Gürgen and Asst. Prof. Rahim Dehkharghani for kindly accepting to be part of my thesis jury and their valuable comments.

I would like to thank my parents and sister for always supporting me throughout my life. They have always encouraged me to aim higher and trusted in me.

Finally, but the most importantly, I would like to thank my family and especially my dear wife Dilem for her patience and support whom have helped me share this burden throughout all these years. This journey would have not been possible without her continued support and encouragements. My warmest thanks go to my lovely daughter Birce. Every time I see her, I am filled with the energy that has kept me going no matter what. The thought of having the support of this wonderful family has given me the strength to always continue on this long and hard path.

ABSTRACT

ABSTRACTIVE TEXT SUMMARIZATION FOR MORPHOLOGICALLY RICH LANGUAGES

The exponential growth in the number of documents available on the Web has turned finding the relevant piece of information into a challenging, tedious, and time-consuming activity. Accordingly, automatic text summarization has become an important field of study by gaining significant attention from the researchers. Recent progress in deep learning shifted the research in text summarization from extractive methods towards more abstractive approaches. The research and the available resources are mostly limited to the English language, which prevents progress in other languages which especially differ in terms structure and characteristics such as the morphologically rich languages (MRLs). In this thesis, we mainly focus on abstractive text summarization on two MRLs, Turkish and Hungarian, and address their important challenges. Firstly, we tackle the resource scarcity problem by curating two large-scale datasets for Turkish (TR-News) and Hungarian (HU-News) aimed for text summarization, but are also suitable for other tasks such as topic classification, title generation, and key phrase extraction. Then, we utilize the morphological properties of these languages and adapt them to summarization where we show improvements upon the existing models. Later, we make use of pretrained multilingual sequence-to-sequence models and provide state-of-the-art models for abstractive text summarization and title generation tasks. Evaluation of text summarization for MRLs is very limited. Thus, we show how preprocessing can drastically influence the evaluation results through a case study in Turkish. Finally, morphosyntactic methods are proposed for text summarization evaluation and a human judgement dataset is curated. It is shown that morphosyntactic tokenization processes during evaluation increase correlation with human judgements. All the work and the curated datasets are made publicly available.

ÖZET

Biçim Bilimsel Açıdan Zengin Dillerde Soyutlamalı Özetleme

İnternet ortamında bulunan belge sayısındaki yoğun artış, aranan bilgiye ulaşımı zorlu, sıkıcı ve zaman alıcı bir faaliyet haline getirmiştir. Bu doğrultuda otomatik metin özetleme, araştırmacıların büyük ilgisini çekerek önemli bir çalışma alanı haline gelmiştir. Derin öğrenme alanındaki son gelişmeler, metin özetleme alanındaki araştırmaları çıkarımsal yöntemlerden daha soyut yaklaşımlara doğru kaydırmıştır. Araştırmalar ve mevcut kaynaklar çoğunlukla İngilizce diliyle sınırlıdır, bu da özellikle biçim bilimsel açıdan zengin diller gibi yapısı ve özellikleri bakımından farklılık gösteren diğer dillerde ilerlemeyi engellemektedir. Bu tezde, ağırlıklı olarak Türkçe ve Macarca soyut metin özetleme üzerine odaklandık ve önemli zorluklarını inceledik. İlk olarak, Türkçe (TR-News) ve Macarca (HU-News) için metin özetleme alanında kullanımı amaçlayan, ancak konu sınıflandırması, başlık oluşturma ve anahtar kelime öbeği çıkarma gibi diğer görevler için de uygun olan iki büyük ölçekli veri kümesini oluşturarak kaynak kıtlığı sorununu ele aldık. Daha sonra, bu dillerin biçim bilimsel özelliklerini metin özetlemeye uyarlayarak mevcut modeller üzerine iyileştirmeler gerçekleştirdik. Bir sonraki aşamada, önden eğitilmiş çok dilli diziden diziye modellerden yararlanarak, soyut metin özetleme ve başlık oluşturma görevleri için son teknoloji modeller oluşturduk. Biçim bilimsel açıdan zengin diller için metin özetleme değerlendirmesi çalışmaları oldukça sınırlıdır. Bu nedenle, ön işlemenin değerlendirme sonuçlarını nasıl büyük ölçüde etkileyebileceğini Türkçe bir çalışmayla gösterdik. Son olarak, metin özetleme değerlendirmesi için morfosentaktik yöntemler önerip buna ek olarak bir insan yargısı veri kümesi derledik. Değerlendirme sırasında morfosentaktik yöntemlerin insan yargıları üzerindeki korelasyonu artırdığını gözlemledik. Tez kapsamında yapılan tüm çalışmalar ve veri kümeleri açık kaynak olarak kullanıma sunulmuştur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . x
LIST OF TABLES . . . xi
LIST OF SYMBOLS . . . xix
LIST OF ACRONYMS/ABBREVIATIONS . . . xxi
1. INTRODUCTION . . . 1
1.1. Contributions . . . 4
1.2. Publications . . . 6
1.3. Thesis Outline . . . 6
2. BACKGROUND . . . 8
2.1. Text Summarization . . . 8
2.2. Morphologically Rich Languages: Turkish and Hungarian . . . 9
2.3. Neural Abstractive Summarization Components . . . 12
2.3.1. Bidirectional LSTM . . . 12
2.3.2. Attention Mechanism . . . 13
2.3.3. Transformer Network . . . 14
2.3.4. Neural Text Generation . . . 15
2.4. Evaluation Methods . . . 16
2.4.1. Human Evaluation . . . 16
2.4.2. Automatic Evaluation . . . 17
2.4.2.1. ROUGE . . . 18
2.4.2.2. METEOR . . . 18
3. RELATED WORK . . . 20
3.1. Abstractive Summarization . . . 20
3.2. Pretrained Sequence-to-Sequence Models . . . 21
3.3. Turkish Text Summarization . . . 22

3.4. Hungarian Text Summarization 23
3.5. Summarization Evaluation 23
3.6. Tokenization 24
4. MORPHOLOGY-BASED ABSTRACTIVE TEXT SUMMARIZATION AND LARGE-SCALE DATASETS FOR AGGLUTINATIVE LANGUAGES TURKISH AND HUNGARIAN 26
4.1. Introduction 26
4.2. Datasets 28
4.2.1. Dataset Compilation 28
4.2.2. Statistics 31
4.3. Methodology 37
4.3.1. Models 37
4.3.1.1. Pointer-Generator Network with Coverage Mechanism 37
4.3.1.2. BERT + Transformer 38
4.3.2. Morphological Tokenizers 39
4.3.2.1. Turkish 41
4.3.2.2. Hungarian 43
4.4. Experimental Settings 43
4.4.1. Experiment 1 - Pointer-Generator Network and Morphological Tokenizers 43
4.4.2. Experiment 2 - BERT-based Abstractive Text Summarization . 44
4.5. Results 45
4.5.1. Quantitative Analysis 45
4.5.1.1. Experiment 1 Results 46
4.5.1.2. Experiment 2 Results 49
4.5.2. Qualitative Analysis 52
4.6. Discussion 59
5. TURKISH ABSTRACTIVE TEXT SUMMARIZATION USING PRETRAINED SEQUENCE-TO-SEQUENCE MODELS 61
5.1. Introduction 61
5.2. Models 63

5.2.1. BERT2BERT 64
5.2.2. mBART 65
5.2.3. mT5 67
5.3. Datasets 69
5.4. Experiments 72
5.4.1. Tokenization Analysis 73
5.4.2. Experiment 1 - Summary Generation 76
5.4.3. Experiment 2 - Title Generation 77
5.4.4. Novelty Analysis 78
5.5. Results 79
5.5.1. Quantitative Results 79
5.5.1.1. Experiment 1 - Summary Generation 79
5.5.1.2. Experiment 2 - Title Generation 82
5.5.1.3. Novelty Analysis 84
5.5.1.4. Cross Dataset Evaluations 85
5.5.1.5. Generation Parameters: Beam size and early-stopping 86
5.5.1.6. ROUGE Assessment Variations 88
5.5.2. Qualitative Results 90
5.5.2.1. Summary Generation 90
5.5.2.2. Title Generation 96
5.6. Discussion 99
6. MORPHOSYNTACTIC EVALUATION FOR MORPHOLOGICALLY RICH LANGUAGES: A CASE STUDY FOR TURKISH 101
6.1. Introduction 101
6.2. Methodology 102
6.2.1. Morphosyntactic Variations 102
6.2.2. Evaluation Metrics 106
6.3. Dataset, Models, and Annotations 107
6.3.1. Dataset and Models 108
6.3.2. Human Judgment Annotations 108
6.4. Correlation Analysis 109

6.5. Discussion 112
7. CONCLUSION 114
REFERENCES 116
APPENDIX A: ADDITIONAL TABLES REGARDING CHAPTER 4 132
APPENDIX B: ADDITIONAL TABLES REGARDING CHAPTER 5 136
APPENDIX C: ADDITIONAL TABLES REGARDING CHAPTER 6 140

LIST OF FIGURES

Figure 2.1. Seq2Seq Architecture based on LSTM 16

Figure 4.1. Crawling process using Scrapy 30

Figure 4.2. N-gram novelty comparison between content and abstract 36

Figure 4.3. Pointer-generator network 38

Figure 5.1. A high-level transformer-based encoder-decoder network. 64

Figure 5.2. A number of noising methods experimented in the BART model. T1-T6 denote tokens. The box that the arrows point to shows the denoised text. 66

Figure 5.3. Various downstream tasks such as machine translation, semantic textual similarity, and text summarization on mT5 framework shown with examples in Turkish. 67

Figure 5.4. Average number of tokens generated by the tokenizers of the models for content, abstract, and title. 75

LIST OF TABLES

Table 2.1. Morphological parses of the word *karın* 11

Table 4.1. Comparison of summarization datasets with respect to overall corpus size, sizes of training, validation, and test sets, average content and abstract lengths (in terms of words and sentences) 31

Table 4.2. Comparison of summarization datasets with respect to vocabulary size and type-token ratio of both content and abstract. 31

Table 4.3. Two news articles selected from TR-News and HU-News. All the collected fields are shown: URL, title, abstract, content, topic, tags, date, author, and source. 33

Table 4.4. Turkish morphological tokenization methods 41

Table 4.5. Hungarian morphological tokenization methods 42

Table 4.6. Results of pointer-generator models with different tokenization methods on CNN/Daily Mail, TR-News, and HU-News datasets in terms of ROUGE-1, ROUGE-2, and ROUGE-L F-measure. "-" denotes result is not available. Bold values show the highest scores obtained in the experiments per dataset. 46

Table 4.7. Novel n-gram ratios for the models in Experiment 1. N1, N2, and N3 respectively represent n-grams (n=1,2,3). 48

Table 4.8. OOV analysis for the models in Experiment 1. OOV and OOV ratio denote, respectively, the average number of OOV words and the percentage of OOV per summary. 48

Table 4.9. BERT+Transformer results on CNN/Daily Mail, TR-News, and HU-News datasets in terms of ROUGE-1, ROUGE-2, and ROUGE-L F-Measure. "-" denotes data is not available. Bold values show the highest scores obtained in the experiments per dataset. 49

Table 4.10. OOV analysis results for the models in Experiment 2. OOV and OOV ratio denote, respectively, the average number of OOV words and the percentage of OOV per summary and content. 50

Table 4.11. Novel n-gram ratios for the models in Experiment 2. N1, N2, and N3 respectively represent n-grams (n=1,2,3). 51

Table 4.12. First example document and generated summaries from TR-News for qualitative analysis. 52

Table 4.13. Second example document and generated summaries from TR-News for qualitative analysis. 54

Table 5.1. Comparison of summarization datasets with respect to sizes of training, validation, and test sets, and average content, abstract, and title lengths (in terms of words and sentences) 69

Table 5.2. Comparison of summarization datasets with respect to vocabulary size and type-token ratio of content, abstract, title, and overall. . . 70

Table 5.3. Two news articles selected from TR-News and MLSum (TR) . . . 71

Table 5.5. Tokenization outputs of the methods for a given Turkish sentence which translates to "If one day, my words are against science, choose science.". 74

Table 5.6. Novelty ratios of the datasets with respect to the summary generation and title generation tasks. N1, N2, and N3 denote uni-gram, bi-gram and tri-gram ratios, respectively. 78

Table 5.7. Text summarization results of pretrained encoder-decoder models on TR-News, MLSum (TR), and Combined-TR datasets. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) scores are given in F-measure. "-" denotes result is not available. Bold values show the highest scores obtained in the experiments per dataset. 80

Table 5.8. Title generation (abstract as input) results of pretrained encoder-decoder models on TR-News, MLSum (TR), and Combined-TR datasets. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) are given in F-measure. Bold values show the highest scores obtained in the experiments per dataset. 81

Table 5.9. Title generation (LEAD-3 as input) results of pretrained encoder-decoder models on TR-News, MLSum (TR), and Combined-TR datasets. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) are given in F-measure. Bold values show the highest scores obtained in the experiments per dataset. 82

Table 5.10. Title generation LEAD sentences ablation study results. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) are given in F-measure. 83

Table 5.11. Novelty ratios of the summaries generated by the models per dataset. N1, N2, and N3 denote uni-gram, bi-gram, and tri-gram ratios, respectively. Bold values show the highest scores obtained in the experiments per dataset (the mBERT-uncased results are misleading and are ignored due to the high number of unknown tokens output). 83

Table 5.12. Novelty ratios of the titles (abstracts are given as input) generated by the models per dataset. N1, N2, and N3 denote uni-gram, bi-gram, and tri-gram ratios, respectively. Bold values show the highest scores obtained in the experiments per dataset. 84

Table 5.13. Cross-dataset evaluation results for the summary generation and the title generation (abstract as input) tasks. The values correspond to ROUGE-1 scores. 85

Table 5.14. Results for the summary generation and title generation (abstract as input) tasks with various beam sizes and early-stopping method. The values correspond to ROUGE-1 scores. Bold values show the highest scores obtained in the experiments per dataset. 87

Table 5.15. ROUGE scores calculated with different preprocessing settings. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations. 88

Table 5.16. An example from the test set of TR-News accompanied with the summaries generated by the models. The spelling and grammatical errors in the original texts are left as is. News article's content is given as the input and the reference summary is the abstract of the article. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models, whereas the underlined texts are for reference in the discussion. 90

Table 5.17. An example from the test set of MLSum (TR) accompanied with the summaries generated by the models. News article's content is given as the input and the reference summary is the abstract of the article. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models, whereas the underlined texts are for reference in the discussion. 93

Table 5.18. An example from the test set of TR-News accompanied with the titles generated by the models. News article's abstract is given as the input and the title of the article is expected as the output. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models, whereas the underlined texts are for reference in the discussion. 97

Table 5.19. An example from the test set of MLSum (TR) accompanied with the titles generated by the models. News article's abstract is given as the input and the title of the article is expected as the output. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models. 98

Table 6.1. Morphological analysis of an example sentence. 103

Table 6.2. Proposed methods based on morphosyntactic variations of words. . 104

Table 6.3. Average scores and inter-annotator agreement scores for the models. In the first row, the averages of the two annotators are separated by the / sign. 108

Table 6.4. Pearson correlation results of the morphosyntactic methods with prefix tokens for the BERTurk-cased summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column. 110

Table 6.5. Pearson correlation results of the morphosyntactic methods with prefix tokens for the mT5 summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column. 110

Table A.1. A detailed example for Hungarian morphological parsing and disambiguation. 132

Table A.2. A detailed example for Turkish morphological parsing and disambiguation. 134

Table B.1. Cross-dataset evaluation results for the summary generation task. . 136

Table B.2. Cross-dataset evaluation results for the title generation (abstract as input) task. 136

Table B.3. The analysis results for the summary generation task given various beam sizes and early-stopping method. 136

Table B.4. The analysis results for the title generation (abstract as input) task given various beam sizes and early-stopping method. 137

Table B.5. ROUGE scores with different preprocessing settings for the summary generation task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations. 137

Table B.6. ROUGE scores with different preprocessing settings for the title generation (abstract as input) task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations. 138

Table B.7. ROUGE-1 scores of all the models calculated under different preprocessing settings on the TR-News dataset for the text summarization task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations. 138

Table B.8. ROUGE-1 scores of all the models calculated under different preprocessing settings on the MLSum (TR) dataset for the text summarization task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations. . . 139

Table B.9. ROUGE-1 scores of all the models calculated under different preprocessing settings on the Combined-TR dataset for the text summarization task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations. . . 139

Table C.1. Pearson correlation results of the morphosyntactic methods without prefix tokens for the BERTurk-cased summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column. 140

Table C.2. Pearson correlation results of the morphosyntactic methods without prefix tokens for the mT5 summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column. 141

LIST OF SYMBOLS

a	alignment model of attention mechanism
a_i^t	attention distribution in pointer generator network
a_{ij}	weight of the alignment i,j in attention mechanism
b_i	bias of input gate
b_f	bias of forget gate
b_o	bias of output gate
b_c	bias of memory gate
b_{ptr}	bias of pointer generator network
C_t	memory state at timestep t
$\widetilde{C}_t$	new memory state at timestep t
c_i	context vector at timestep i of attention mechanism
c_t	context vector at timestep t in pointer generator network
d	dimension of the key/value vector in transformer network
f_t	forget gate at timestep t
h_t	hidden state at timestep t
i_t	input gate at timestep t
K	key vector in transformer network
λ	lambda value in Poisson distribution
o_t	output gate at timestep t
p_{gen}	probability of generating a vocabulary token in pointer generator network
p_{vocab}	probability of copying a word from input in pointer generator network
s_t	decoder state at timestep t in pointer generator network
σ	sigmoid function
s_i	hidden state at timestep i of attention mechanism
Q	query vector in transformer network
$tanh$	hyperbolic tangent function
V	value vector in transformer network

W_i	weight matrix of input gates
W_o	weight matrix of output gates
W_f	weight matrix of forget gates
W_c^T	weight matrix of context at timestep t in pointer generator network
W_s^T	weight matrix of decoder state at timestep t in pointer generator network
x_t	input at timestep t
y_i	output at timestep i of attention mechanism

LIST OF ACRONYMS/ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
BPE	Byte Pair Encoding
C4	Colossal Clean Crawled Corpus
CC	Common Crawl
CNN	Convolutional neural networks
CSS	Cascading Style Sheets
DUC	Document Understanding Conference
GRU	Gated Recurrent Unit
HTML	HyperText Markup Language
HU-News	Hungarian News Dataset
IDF	Inverse Document Frequency
LCS	Longest Common Sub-Sequences
LM	Language Model
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
LSTM	Long Short-Term Memory
mBERT	Multilingual Bidirectional Encoder Representations from Transformers
mC4	Multilingual Colossal Clean Crawled Corpus
MDS	Multi Document Summarization
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MLSum	Multilingual Summarization Dataset
MRL	Morphologically Rich language
mT5	Multilingual Text-to-Text Transfer Transformer
NER	Named Entity Recognition
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NNLM	Neural Network Language Model

OOV	Out-of-Vocabulary
POS	Part-of-Speech
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
Seq2Seq	Sequence-to-Sequence
SDS	Single Document Summarization
SMD	Sentence Mover Distance
T5	Text-to-Text Transfer Transformer
TAC	Text Analysis Conference
TR-News	Turkish News Dataset
TTR	Type-Token Ratio
TULAP	Turkish Language Processing Platform
WMD	Word mover distance
XPath	XML Path Language
XSum	Extreme Summarization Dataset

1. INTRODUCTION

With the emergence of the Web, there has been an exponential increase in the number of documents made available online from sources such as websites, news, blogs, books, scientific papers, and social media. In parallel to this, it has become increasingly difficult for users to find the information they are interested in due to repetitive and irrelevant content. Moreover, the time and effort that are required to comprehend all these sources are immense. There is a need to automatically digest and extract the essence of all this information since it is impractical for humans to comprehend this vast amount of information through manual efforts. In this regard, text summarization has become an inevitable necessity and a very popular field of study in the past few decades.

Text summarization aims at automatically generating a concise piece of text from a long document, which is capable of portraying the most important information in a fluent and salient way [1,2]. There are two main approaches in automatic text summarization: extractive text summarization [3,4] and abstractive text summarization [5–7]. Extractive text summarization produces summaries by selecting the most relevant sentences or phrases from the input text without reflecting any changes. Abstractive text summarization, on the other hand, is a more challenging task where the aim is to generate a human like summary through making use of complex natural language understanding and generation capabilities.

Recent advances in deep learning have enabled significant progress in natural language understanding and generation tasks. Especially, breakthroughs such as the attention mechanism [8] and the recent Transformer model [9] have been crucial in sequence to sequence (Seq2Seq) tasks. The encoder-decoder architecture has been the most widely used method for Seq2Seq tasks, including abstractive summarization [5, 6, 10–13]. Accordingly, such methods have helped abstractive summarization to progress very rapidly in the past years. Building on the Transformer model, pretrained

language models like BERT [14] have been widely applied to various tasks and have shown to be very effective producing state-of-the-art results for text summarization [15]. Later, studies leveraged the pretraining for Seq2Seq models [16–20] to further improve upon the language generation tasks. Accordingly, pretraining Seq2Seq models especially on large scale datasets has shown to perform very well, reaching state-of-the-art results in neural abstractive summarization for English [7, 21].

Although there have been a great number of research in abstractive text summarization for the English language, the amount of studies was very limited for morphologically rich languages (MRLs). Agglutinative languages which are also MRL such as Turkish and Hungarian differ from other languages in the sense that the word formation process heavily depends on affixation. Each affix added to the word can have a great impact leading to complete change in the form and meaning of words. The morpho-syntactic properties of these languages enable the word to carry much more information compared to other languages such as English. In this work, we focus on abstractive text summarization for morphologically rich languages. Although our study focuses mostly on Turkish and Hungarian, it can easily be expanded to other MRLs. Turkish was chosen since it is native to the author. Hungarian was chosen because it is a commonly spoken MRL and it was one of the languages which had suitability for data curation as explained in Section 4.2.1. There are various challenges and limitations for morphologically rich languages in abstractive text summarization. The most important of these are listed below:

- **Limited research on abstractive summarization.** Despite significant advances in abstractive summarization, these were mostly limited to English since most of the resources and studies are available in this language. For Turkish, almost all of the studies are extractive [22–25] except a very few recent ones which have been published after we have started working on this topic [26, 27]. Hungarian text summarization has been studied even less than Turkish. It has been mostly employed on speech data [28, 29] and to the best of our knowledge, there hasn't been any study for abstractive text summarization in Hungarian. Addi-

tionally, it is not safe to assume that state-of-the-art models in English will have similar performances in other languages. The results might differ depending on the language and methods can be validated only through comprehensive study.

- **Resource scarcity.** Due to lack of studies, the data resources are also very limited. This is an important obstacle preventing progress in resource-scarce languages that needs to be overcome. Resource curation is an important but expensive process in terms of time and effort. In Turkish text summarization, almost all of the works done are in extractive manner where small scale datasets are utilized. The dataset sizes range from 50 [22] to 120 [23] documents. The sizes of the datasets are especially critical for abstractive summarization where mostly deep learning-based approaches are utilized. Large amount of data is needed to be able to train an adequate model which is capable of generating reasonable summaries. The only large-scale dataset for Turkish to the best of our knowledge is a recent work where a large-scale multilingual corpus including Turkish has been released [26]. For Hungarian, we are not aware of any large-scale dataset that can be utilized for abstractive text summarization.
- **Morphological Complexity.** MRLs can express the same information with fewer words compared to English and even a sentence can be represented with just one word (see Section 2.2). However, this leads to a very large vocabulary which gives rise to the out of vocabulary (OOV) problem especially for text generation tasks. Therefore, the complexity of morphology is a significant challenge for agglutinative languages. Moreover, utilizing morphology correctly has shown to be effective for several tasks such as named entity recognition (NER) [30], part-of-speech (POS) tagging [31], learning word embeddings [32, 33], and machine translation [34]. To the best of our knowledge, the effect of morphology has not been studied in abstractive text summarization for neither Turkish nor Hungarian.
- **Limitations of evaluation methods.** Evaluation of summarization methods is critical to assess and benchmark their performance. The success of a summarization system can only truly be reflected by using an adequate evaluation method. The main objective of evaluation is to observe how well the output sum-

mary is able to reflect the reference summaries. The commonly used evaluation methods in summarization such as ROUGE [35] and METEOR [36] are based on n-gram matching strategy. For instance, ROUGE computes the number of overlapping word n-grams between the reference and system summaries in their exact (surface) forms. While the exact matching strategy is not an issue for extractive summarization where the words are directly copied, it poses a problem for abstractive summarization where the generated summaries can contain new words or the same words in different forms. In the abstractive case, this strategy is very strict especially for morphologically rich languages in which the words are subject to extensive affixation and thus carry syntactic features. It severely punishes the words that have even a slight change in their forms. Hence, taking the morphosyntactic structure of these morphologically rich languages into account is important for the evaluation of text summarization. The studies which aim to utilize morphology has been very limited for text summarization [37].

In this thesis we addressed these challenges and limitations. Accordingly, we curated large-scale datasets that are suitable for abstractive text summarization. Then, we provided state-of-the-art models for both text summarization and title generations tasks. We showed the importance of morphological information in abstractive text summarization for morphologically rich languages by incorporating this information to model training and evaluation. The contributions are explained in more detail in Section 1.1.

1.1. Contributions

Although Chapters 4-6 contain detailed explanation of the work, outcomes, and contributions, we provide the most important contributions as a summary in this section to give a better overall understanding of the thesis. Accordingly, the contributions of the thesis which address the challenges listed in the previous section are the following.

(i) Two large-scale publicly available text summarization datasets[1] for two resource-scarce agglutinative languages, Turkish and Hungarian are released publicly. The datasets also contain information that can be used in other tasks such as title generation, topic classification, key phrase extraction, and author detection. To the best of our knowledge, HU-News is the first large-scale summarization dataset for Hungarian. The datasets[2] [3] can also be easily accessed through a well known open source platform.

(ii) Two types of morphological tokenization approaches (SeperateSuffix and CombinedSuffix) are proposed for both Turkish and Hungarian. Through these tokenization methods, the effect of morphology is studied on both datasets using the pointer generator model. The SeperateSuffix method achieves the highest ROUGE-1 F-Score on the TR-News dataset amongst all the models used in Chapter 4 surpassing BERT-based models and provides promising results on the HU-News dataset. The morphological tokenization library is implemented in a flexible and extendable manner so that more methods and also languages can be added easily. It is made publicly available [4] .

(iii) For Hungarian, we show that a transformer-based encoder-decoder network that utilizes multilingual cased BERT model as encoder and standard transformer network as decoder reaches the state-of-the-art results on HU-News dataset.

(iv) We show that pretrained sequence-to-sequence models reach state-of-the-art on the TR-News and MLSum datasets for summary generation and title generation tasks. The first study that utilizes the titles of both datasets (TR-News and MLSum) for the Turkish language is conducted. Comprehensive and strong baselines for the title generation task is provided.

(v) Evaluation of text summarization is studied comprehensively. The importance of preprocessing the text such as removing punctuations or applying stemming before evaluation and how drastically such operations can influence the end results is shown through a case study in Turkish. Morphosyntactic preprocessing

[1] https://github.com/batubayk/datasets
[2] https://huggingface.co/datasets/batubayk/TR-News
[3] https://huggingface.co/datasets/batubayk/HU-News
[4] https://github.com/batubayk/MorphologicalTokenizers

methods are adopted for several commonly used evaluation metrics and its affects are analyzed on the TR-News dataset using the state-of-the-art models trained in Chapter 5. In order to evaluate the proposed methods, a manually annotated human judgement dataset was curated and made publicly available[1]. It is shown that morphosyntactic tokenization processes during evaluation is more correlated with human judgements and contributes to the evaluation process positively.

(vi) A Turkish abstractive text summarization tool[5] within TULAP (Turkish Language Processing Platform) is created using the state-of-the-art model obtained in Chapter 5.

1.2. Publications

The following papers have been published and submitted as part of the thesis work.

(i) Baykara, Batuhan, and Tunga Güngör. "Abstractive text summarization and new large-scale datasets for agglutinative languages Turkish and Hungarian." Language Resources and Evaluation 56.3 (2022): 973-1007. (Chapter 4)

(ii) Baykara, Batuhan, and Tunga Güngör. "Turkish abstractive text summarization using pretrained sequence-to-sequence models." Natural Language Engineering (2022): 1-30. (Chapter 5)

(iii) Baykara, Batuhan, and Tunga Güngör. "Morphosyntactic Evaluation for Text Summarization in Morphologically Rich Languages: A Case Study for Turkish" Proceedings of the 28th international conference on applications of natural language to information systems (NLDB) (2023). Accepted (Chapter 6)

1.3. Thesis Outline

The rest of the thesis is outlined as follows. Background information on text summarization, MRLs, abstractive summarization components, and evaluations methods

[5]https://tulap.cmpe.boun.edu.tr/demo/bounsumm

are briefly given in Chapter 2. The related work in terms of abstractive text summarization, Turkish and Hungarian text summarization, evaluation methods, and tokenization methods are provided in Chapter 3. Chapter 4 describes the curation process and the statistical details of the datasets TR-News and HU-News released in this thesis. Then, the proposed morphological tokenization methods are explained and utilized in models to show the effectiveness of morphology for summarization. In Chapter 5, pretrained Seq2Seq models are finetuned for both text summarization and title generation tasks showing state-of-the-art results on the TR-News dataset. Later in Chapter 6, morphosyntactic features are used to propose alternative evaluation methods. Finally, the thesis is concluded in Chapter 7.

2. BACKGROUND

2.1. Text Summarization

Text summarization is a very broad topic where a number of distinctions are made depending on several aspects such as approach, input size, intent, and input language. These distinctions are briefly explained in this section.

There are two main approaches in text summarization: extractive and abstractive. Extractive text summarization produces summaries by selecting the most relevant sentences or phrases from the input text without reflecting any changes and orders these without any modification. Abstractive text summarization, on the other hand, is a more challenging task where the aim is to generate a human like summary through making use of complex natural language understanding and generation capabilities. The generated summaries aim to have new and original content which is not present in the source document.

The number of input documents is a factor that determines the type of the summarization. In the case there is a single source document to be summarized, the task is called single document summarization (SDS). The number of input documents can also be greater than one and in this case the task is called multi document summarization (MDS). The most important difference of MDS compared to SDS is to reduce the repetitive content which is caused by the increased number of input documents and also create a salient summary in harmony by incorporating all the key information from multiple documents.

Language is another determining factor of the type of summarization. A summarization system is considered to be monolingual when both the input content and the output summary belongs to a single language. In the case where where the input content consists of several languages and a summary is generated in all of these languages,

it is called a multilingual summarization system. Lastly, it is called a cross-lingual system when the input content is from one language and the output summary is in another language.

The intent of a summary is also important because in majority of the cases, summarization systems are intended to be used in specific tasks. For instance, generating a summary of a book and summarizing a news article are two very different type of tasks which have their own requirements and limitations. Therefore, summarization systems are mostly categorized depending on the intended task that they are going to be used on. The most common type of summarization tasks in the literature are news summarization, headline generation, and long document summarization.

Summarization tasks can be further broken into domain specific and general. Domain specific summarization systems are specialized systems that aim to summarize documents in certain domains which might require very specific knowledge. On the other hand, general summarization system can be utilized in domain independent, common knowledge.

2.2. Morphologically Rich Languages: Turkish and Hungarian

Morphologically rich languages are capable of expressing a very broad range of information through small grammatical units which are referred to as morphemes within the word level. These types of languages mostly belong to either fusional (e.g. Spanish, Arabic, and Hebrew) or the agglutinative (e.g. Turkish, Hungarian, Czech, Finnish, and Korean) language groups which are in the family of synthetic languages [38]. This study focuses on Turkish and Hungarian, hence both languages will be explained further.

Turkish is an agglutinative language which makes use of suffixation extensively. A root word can take several suffixes in a predefined order as dictated by the morphotactics of the language. It is common to find words affixed with 5-6 suffixes. During

the affixation process, the words are also subject to a number of morphophonemic rules such as vowel harmony, elisions, or insertions. Turkish morphology has very few number of prefixes and it is regarded as having no prefixes in natural language processing (NLP) studies. There are two types of suffixes as inflectional suffixes and derivational suffixes. While the inflectional suffixes do not alter the core meaning of a word as in (1), the derivational suffixes can change the meaning or the part-of-speech of the word as in (2)[6] .

1) *göz* + -lAr = *gözler*	2) *göz* + -lHk = *gözlük*
eye PLUR eyes	eye NESS eyeglasses

Affixation can lead to very long words like *Çekoslovakyalılaştıramadıklarımızdanmışsınızcasına*, which means "In the manner of you being one of those that we apparently couldn't manage to convert to Czechoslovakian". Although this example is an uncommon word in Turkish, possibilities of deriving and inflecting words from a single root can be immense [39]. Additionally, such morphologically rich languages also contain ambiguity on the word level which adds another level of complexity. Table 2.1 displays the possible morphological parses of the word *karın* which contain its various inflected and derived forms. Hence, a morphological disambiguation is required to choose the correct form of a given word.

koşuyorum = [koşmak:Verb] koş:Verb + uyor:Prog1 + um:A1sg

The main word order in Turkish is subject-object-verb but the order can be altered depending on the focus of the sentence and all the six word orders are possible [40]. There is no gender in its grammar and the gender does not affect the word forms. Below is the morphological analysis of an example word *koşuyorum* (I am running) that has been analyzed with Zemberek[7] , a Turkish morphological parser and disambiguator that we use in this study. The analysis yields the root of the word as *koş* which is a verb

[6]The upper case letters within suffixes indicate that the sound is phonologically conditioned. 'A' stands for the low vowels 'e' (front) and 'a' (back). 'H' stands for the high vowels 'i' (front unrounded), 'ı' (back unrounded), 'ü' (front rounded), and 'u' (back rounded).

[7]https://github.com/ahmetaa/zemberek-nlp

Table 2.1. Morphological parses of the word *karın*

Root Form	Part of Speech	Parse	Meaning (Translation)
karmak	Verb	kar:Verb+Imp+ın:A2pl	shovel!
karın	Noun	karın:Noun+A3sg	stomach
kar	Noun	kar:Noun+A3sg+ın:Gen	snow's
kar	Noun	kar:Noun+A3sg+ın:P2sg	your snow
kâr	Noun	kar:Noun+A3sg+ın:Gen	profit's
kâr	Noun	kar:Noun+A3sg+ın:P2sg	your profit
karı	Noun	karı:Noun+A3sg+n:P2sg	your wife
karımak	Verb	karı:Verb|n:Pass→Verb+Imp+A2sg	age!
karınmak	Verb	karın:Verb+Imp+A2sg	mix together!

and its infinitive case is *koşmak*. The morpheme *-uyor* is progressive tense morpheme and *-um* denotes first person singular form.

Hungarian is similar to Turkish as it is also highly agglutinative and makes use of affixes. The most commonly used word order is subject-verb-object, which is different from Turkish. Topic-comment structure is effective in determining the word order. The topic of the dialogue and the response that will be given can change the order of words within a sentence. Hungarian is also a genderless language and does not feature grammatical and pronominal gender [41]. In this study, we use the POS tagger PurePOS [8] and the morphological parser emMorph[9] (Humor) for Hungarian language. Below is an example which shows the analysis of the word *játszhatnak* (They can play). The word *játszik* corresponds to the verb *to play* and the morpheme *-hat* is tagged as modal whereas the morpheme *-nak* denotes present tense, indefinite conjugation and third person plural.

játszhatnak = játszik[/V]=játsz + hat[Mod/V] + nak[Prs.NDef.3Pl]

[8]https://github.com/dlt-rilmta/purepospy
[9]https://github.com/dlt-rilmta/emmorphpy

2.3. Neural Abstractive Summarization Components

In recent years there have been many advances in text generation tasks such as machine translation, text summarization, and dialogue generation. Especially deep learning approaches have been very effective in such domains compared to more traditional methods. Text summarization methodologies have also shifted from traditional to neural approaches. Therefore, it is important to briefly go over the building blocks of the more recently used neural networks and methods.

2.3.1. Bidirectional LSTM

Long short-term memory (LSTM) [42] networks are an upgrade over Recurrent Neural Networks (RNNs) which are neural network cells that recur onto themselves for a number of time steps. An RNN cell works like a memory cell in the sense that it tends to remember important parts of the information that stream through the earlier parts of the sequence and also to forget or eliminate the unnecessary information. It passes the salient information to the next time steps. The main problems in this network type are the exploding and vanishing gradients [43]. LSTM cells have been proposed to overcome such issues in vanilla RNN cells. Below are the equations that define an LSTM cell.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.1}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.3}$$

$$\widetilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.4}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{2.5}$$

$$h_t = o_t * tanh(C_t) \tag{2.6}$$

In the equations σ denotes the sigmoid function, $tanh$ the hyperbolic tangent function, and $*$ is used for element wise multiplication. The input, output and forget gates are denoted as respectively, i, o and f. The memory states are shown with C and $\widetilde{C}$ where C is used to show the content of the memory cell and $\widetilde{C}$ for the new memory content.

Bi-RNN emerged as an improvement over the RNN [44]. It was applied to different kinds of RNN structures like LSTM and Gated Recurrent Unit (GRU) [45]. Two sets of LSTM networks are used mutually where one of them reads the input sequence in the forward direction whereas the other reads the sequence backwards. The hidden layers of the last timestep are usually used as the output of the LSTMs and given to another network as an input (e.g. a linear layer for a classification task). In the case of Bi-LSTMs, both are mostly concatenated to form a combined output as $h_t = [\overrightarrow{h_t} : \overleftarrow{h_t}]$.

2.3.2. Attention Mechanism

Although being successful for tasks that work on short sequences, LSTM-based sequence-to sequence (Seq2Seq) architectures have difficulties in remembering contextual information from early timesteps for long sequences. To overcome this problem, attention mechanism was introduced in a machine translation study [8], but is applicable to any Seq2Seq task. The attention mechanism is able to incorporate information from the whole input sequence by learning weights which state the importance of each token for the given timestep of the decoder. Hence, loss of contextual information for long sequences is prevented by attending to each input during the decoding stage. In machine translation, this means aligning the words from the source language to the target language by learning to give more importance to more probable alignments.

The formulation of the attention mechanism [8] is shown below. At timestep i of the decoder, the current hidden state s_i of the decoder is computed as a function of the previous time step's hidden state s_{i-1}, output y_{i-1}, and the current context vector c_i. Importantly, the current context vector is calculated by summing each hidden state

h_j of the encoder for the input sequence with respect to its weight α_{ij}. The alignment model a, from which the weights are computed, is modelled as a feed-forward neural network where the parameters define how important each source hidden state is with respect to a target token at each timestep.

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2.7}$$

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \tag{2.8}$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.9}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.10}$$

Following the Bahdanau attention, soft and hard attentions were introduced in the computer vision domain [46] which determine whether the attention has access to, respectively, the entire image or a local portion of the image. Then, local and global attention mechanisms were proposed [47] in NLP. Self attention [48], also referred to as intra-attention, calculates an attention between the current word and the whole sequence. It has been proven to be useful in various text generation tasks including abstractive summarization [49] where capturing the most essential portions of text is important.

2.3.3. Transformer Network

Once the attention mechanism has proven its impact, research shifted towards attention-based architectures. A network architecture called transformer that only relied on attention was introduced [9]. It is entirely built on the self-attention mechanism without using a sequential architecture like RNN. The most basic component in the network is the scaled dot-product attention. The transformer interprets the encoded

input as a set of key-value pairs (K,V) and maps these values with the query (Q) when calculating the output as shown below, where d denotes the dimension of the key/value vector.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{2.11}$$

When this is repeated several times for different (Q,K,V) values, it is called a multi-headed attention mechanism. The mechanism allows the network to learn a variety of different representations by applying different linear transformations and consequently enriching itself with more diverse information. The network is formed of an encoder and a decoder similar to that of the RNN-based encoder-decoder architecture. However, the internal design is different. The encoder part is formed of a multi-head attention mechanism and a simple feed forward neural network with normalization and residual connections after each of them. The decoder part also makes use of the same components but the encoder's output is also fed as an input. The last layer of the decoder is a linear layer which outputs the token probabilities through a softmax function.

2.3.4. Neural Text Generation

Neural text generation has been dominated by Seq2Seq models which was first introduced and applied in the machine translation task [10]. The main approach is to transform an input sequence to a target sequence, both of which are variable length sequences. The architecture of Seq2Seq models is based on two main components: an encoder and a decoder. The encoder part is responsible for learning a complete representation for the input sequence and compresses all the information into a vector of fixed size. Then, the vector is passed into the decoder part where an output sequence is generated based on the given contextual information.

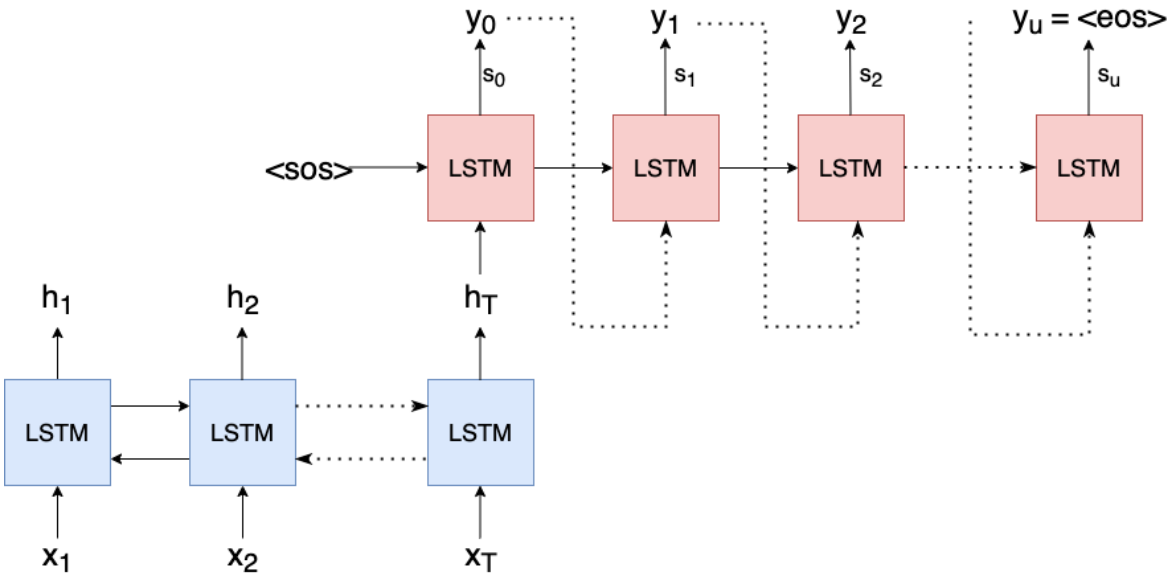

Figure 2.1. Seq2Seq Architecture based on LSTM

An example Seq2Seq architecture is given in Figure 2.1. A bidirectional LSTM is used in the encoder part and a unidirectional LSTM is used in the decoder part where a token is generated at each timestep. The encoder and decoder components of the Seq2Seq architecture can also be constructed by using different models. For instance, the encoder can utilize a transformer-based model such as BERT [14] and the decoder can utilize a RNN-based model such as GRU.

2.4. Evaluation Methods

There are two main ways to evaluate text summarization output: human evaluation and automatic evaluation. In this section both approaches will be explained.

2.4.1. Human Evaluation

The most straightforward and reliable way of evaluating summaries is to have a human evaluation. For this reason, several organisations have been created in the past where judges would evaluate system summaries and provide certain feedbacks. Document Understanding Conference (DUC)[10] organised by National Institute of Standards

[10]https://www-nlpir.nist.gov/projects/duc/data.html

and Technology (NIST) and Text Analysis Conference (TAC) [11] were one the most important events in the field of text summarization. DUC later became a summarization track in the TAC. The outputs of these organisations were curated datasets which were used in evaluating extractive summaries automatically.

Although humans enable high quality evaluations, it is also very challenging and time consuming. For instance, human annotator agreement is a problem since the evaluation is very subjective. In order to reduce the subjectivity and provide a more comprehensive feedback, the evaluation is done based on several criteria such as the following.

- **Readability:** The linguistic quality is checked and made sure that the content is easily understandable.
- **Structure and coherence:** The output is well organized and the flow is correct with each sentence relating to each other.
- **Grammar:** The summary should be correct in terms of grammar and spelling rules.
- **Coverage:** The summary should be able to cover the most important aspects cts of the input document.
- **Conciseness and redundancy:** The output should be as brief as possible while capturing the salient information. Content repetition should not be allowed.

The articles are given scores within a range (most commonly 1-5 where 1 is strongly disagree and 5 is strongly agree) based on each of these criteria.

2.4.2. Automatic Evaluation

Human evaluation is a tedious task and requires a lot of time and effort. Automatic evaluation has been introduced to replace human evaluation and adress these problems while aiming to keep the evaluation quality as high as possible. The most

[11] https://tac.nist.gov/data/forms/index.html

dominantly used automatic evaluation metrics are explained in this section.

2.4.2.1. ROUGE. Recall-Oriented understudy for Gisting Evaluation (ROUGE) [35] is a set of metrics which aims to measure the quality of the summary by comparing it to a reference summary. There are several approaches within ROUGE that aim to measure different aspects of the summaries. ROUGE-N calculates the overlapping n-grams between the candidate summary and a set of reference summaries hence, it is considered to be a recall oriented metric.

$$ROUGE - N = \frac{\sum_{S \in (referenceSummaries)} \sum_{ngram \in (S)} Count_{match}(ngram)}{\sum_{S \in (referenceSummaries)} \sum_{ngram \in (S)} Count(ngram)} \tag{2.12}$$

ROUGE-N is a strict exact matching strategy which requires consecutive matches. ROUGE-L aims to overcome this shortcoming by allowing in-sequence matches into its calculation by making use of longest common sub-sequences (LCS). Accordingly, it is able to capture the sentence and summary level structures in a better way. ROUGE-S metric measures the overlap of any two word in the sentence which follow each other and allows certain gaps. Some restrictions in terms of distance between the two words are imposed so that spurious matches are avoided. ROUGE-SU is an extension to ROUGE-S which eliminates the problem of not giving any importance to a candidate sentence in the case where the sentence does not have any word pair occurring together with its reference. Unigram are also incorporated into the metric to overcome this issue.

In the literature, ROUGE-1, ROUGE-2, and ROUGE-L are the most commonly used metrics for text summarization evaluation.

2.4.2.2. METEOR. The Metric for Evaluation of Translation with Explicit Ordering (METEOR) [36] has been initially proposed for the evaluation of machine translation

task. Hence, it aimed to solve several shortcomings of the commonly used machine translation metric BLEU [50] such as lacking recall in the evaluation procedure. Thus, METEOR makes use of both unigram precision and recall by computing a harmonic mean of two metrics and aims to take longer matches into consideration also by enabling a penalty mechanism. METEOR also includes stemming (Porter Stemmer [51]) and synonym matching (through synsets of Wordnet [52]) in its processes to further increase the matches. However, these modules are not available for the majority of languages.

3. RELATED WORK

3.1. Abstractive Summarization

Recent abstractive summarization methods are based on neural approaches and are conceptualized as Seq2Seq models. Rush et al. [5] were one of the first studies to apply an encoder-decoder architecture using a neural network language model (NNLM) to the title generation task as part of the abstractive summarization problem. Then, Chopra et al. [53] replaced the NNLM with recurrent neural networks. Later, a feature-rich (a vector with word embedding, POS (Part-of-speech) and NER (Named entity recognition) tags, and TF-IDF values) encoder was used to capture important keywords accompanied with a switching pointer-generator to model out of vocabulary (OOV) words and a hierarchical attention model to capture the hierarchy in documents [11]. Importantly, CNN/Daily News dataset was also released in this work to set a benchmark for such abstractive models. The pointer-generator network was later enhanced with a mechanism controlled with a soft switch that allows copying words from the source to eliminate the OOV problem [6]. A coverage mechanism was also introduced in this work that was able reduce word repetition.

Later, a model that uses multiple encoders to represent a document and a hierarchical attention mechanism at decoding time was trained using reinforcement learning [54]. In another work, intra-attention that attends to previous decoded words was proposed to handle the coverage problem [49]. The authors also use trigram blocking to reduce phrase repetition at inference time. Different from other methods, a bottom up approach was proposed that first determines the phrases to be extracted from the input document and then copies the selected phrases into the summary at the decoding step [55]. Convolutional neural networks (CNN) are utilized on the extreme summarization (XSum) dataset which also makes use of the topics in the news articles [56]. After pretrained language models [14] that were trained on huge corpora were introduced, these were also utilized in both extractive and abstractive summa-

rization [15]. For the extractive model, the pretrained BERT models were finetuned by inserting sentence markers in the input to learn the sentence representations and also by differentiating the segment embeddings of the sentences. Hence, the model is capable of learning which sentences to choose from the input text. For the abstractive model, the BERT model was used as the encoder and a transformer model was fitted as the decoder to output words when generating the summary. Recently, the pretrained Seq2Seq models have shown to perform very well for neural abstractive summarization which are further explained in Section 3.2.

3.2. Pretrained Sequence-to-Sequence Models

In recent years, transfer learning in NLP has proven to be very effective and has enabled state-of-the-art results in a large variety of tasks. The concept of pretraining a language model that is capable of learning task-agnostic knowledge through various pretraining objectives and then transferring this knowledge to downstream tasks has been especially successful in natural language understanding [14,57,58]. However, tasks that require both natural language understanding and natural language generation such as machine translation and text summarization could not benefit from these pretrained encoder models as much, leading to pretrained sequence-to-sequence models.

Song et al. [17] proposed MASS, a masked Seq2Seq generation model, that is able to reproduce part of a sentence when the remaining parts are provided. UniLM [16] employed simultaneous training on three types of language modelling objectives: unidirectional, bidirectional, and sequence-to-sequence. In BART, Lewis et al. [19] followed various denoising objectives to first corrupt an input text and then reconstruct it using an autoencoder. T5 [20] introduced a generalized text-to-text framework capable of handling a variety of NLP tasks using solely text as its input and output, and is pretrained on various supervised and unsupervised objectives including summarization. Lastly, the multilingual variations of T5 and BART, respectively mT5 [59] and mBART [60], were released. PEGASUS [7] was specifically pretrained for the abstractive text summarization task and made use of masking whole sentences from a

document and generating these gap-sentences as the pretraining objective. ProphetNet [21] introduced a novel self-supervised objective named as future n-gram prediction and the n-stream self-attention mechanism. Unlike traditional Seq2Seq models which optimize one-step ahead prediction, it optimizes n-steps ahead predicting the next n tokens simultaneously based on previous context tokens at each time step.

3.3. Turkish Text Summarization

The research in Turkish text summarization is mostly based on extractive approaches where more traditional methods are utilized. In an early work, a rule-based system which aims to summarize news articles through various heuristics has been proposed [61]. For instance, more importance is given to the sentences that contain positive sentiments or that are at the introduction or conclusion parts of the input text. Other studies made use of features that are commonly used in extractive text summarization such as term frequency, title similarity, key phrases, and sentence position and centrality to select the most relevant sentences [23, 24]. Özsoy et al. [22] proposed variations to the commonly applied latent semantic analysis (LSA) such as finding the main topics of the text and then selecting the sentences that have the highest scores amongst those topics. Query-biased summarization was studied in the Web information retrieval domain to further improve snippet quality by utilizing the document structure [62]. Güran et al. [25] made use of non-negative matrix factorization and applied various preprocessing methods such as detecting consecutive words, removing stopwords, and stemming. Later, a hybrid extractive summarization system was proposed which uses semantic features extracted from Wikipedia in conjunction with the commonly used structural features [63]. The datasets used in all these studies are limited in size ranging from 50 [22] to 120 [23] documents.

The number of studies on Turkish abstractive text summarization is very limited as well as the applications of pretrained Seq2Seq models on Turkish text summarization and title generation tasks. Scialom et al. [26] evaluated the recent Seq2Seq models (pointer-generator [6] and UniLM [16]) on the MLSum dataset that they have released,

which consists of five different languages including Turkish. Karakoç and Yılmaz [64] employed a plain LSTM-based encoder-decoder network for the title generation task.

3.4. Hungarian Text Summarization

Hungarian text summarization has been studied even less than Turkish. It has been mostly employed on speech data. In a work, highly spontaneous speeches were summarized in an extractive manner using traditional scoring methods such as TF-IDF, latent semantic indexing (LSI), and sentence position [28]. In another work, extractive speech summarization was used as an external evaluation strategy where the aim was to assess the semantic space bias caused by automatic speech recognition [29]. To the best of our knowledge, there hasn't been any study and there is no dataset available for abstractive text summarization in Hungarian at the time of our work.

3.5. Summarization Evaluation

Most of the evaluation methods used in text summarization and other NLP tasks are more suitable for well-studied languages such as English. ROUGE [35] is the most commonly applied evaluation method in text summarization which basically calculates the overlapping number of word n-grams. It also employs other strategies that take longest common sub-sequences or skip-grams into account. Although initially proposed for machine translation, METEOR [36] is also used in text summarization evaluation. METEOR follows the n-gram based matching strategy which builds upon the BLEU metric [50] by modifying the precision and recall computations and replacing them with a weighted F-score based on mapping unigrams and a penalty function for incorrect word order.

Recently, neural evaluation methods have been introduced which aim to capture semantic relatedness. Word mover distance (WMD) [65] and its modified version WMDo [66] which takes word order into account in essence make use of Word2Vec [67] embeddings to represent the input text. Sentence mover distance (SMD) tries to im-

prove WMD by using sentence level embeddings rather than word level embeddings. BERTScore [68] makes use of the BERT model [14] to compute a cosine similarity score between the given reference and system summaries.

There has been very limited research in summarization evaluation for Turkish which has quite different morphology and syntax compared to English. Most of the studies make use of common metrics such as ROUGE and METEOR [22,26]. Recently, Fikri et al. [27] utilized various semantic similarity metrics including BERTScore to semantically evaluate the Turkish summaries on the MLSum [26] dataset.

3.6. Tokenization

Tokenization is a preprocessing step required in almost every NLP application. It is an important step that determines the inputs which will be fed into the model. Thus, the model is directly affected by the tokenization method and its outputs. This is more apparent in natural language generation tasks where the output space, i.e. the size of the vocabulary, heavily depends on the tokenization. In most languages, this is not a serious problem since the number of affixes a word can take and thus the number of word forms are limited whereas this is not true for agglutinative languages. These languages possess a rich morphological process where a high number of affixes can be appended to a root. This issue usually results in high vocabulary sizes compared to other languages given the same amount of text. High vocabulary size in generation tasks increases the number of model parameters since the number of operations required in the output layer of a predictive model is proportional to the vocabulary size. This makes most language generation tasks more challenging in morphologically rich languages.

There have been studies that aimed to reduce the vocabulary size such as byte pair encoding (BPE) [69], unigram language model (unigram LM) [70], WordPiece [71], and Morfessor [72]. These models are trained on a corpus to determine the subword units and hence they are dependent on the corpus they are trained. Each model follows a different methodology. BPE and WordPiece are similar and make use of the frequencies

of character n-gram pairs to merge the most frequent ones at each iteration, whereas the Unigram LM follows a pruning-based approach until a predefined vocabulary size is reached. Subword methods have been mostly applied to neural machine translation and shown to be effective. Recently, a work on comparing subword methods on language model pretraining has been published [73]. It has been shown that unigram LM is capable of capturing morphology better than the BPE model, hence would be more suitable for languages with rich morphology.

Besides subword methods, morphological tokenization has not been studied well for agglutinative languages in neural generation tasks. There has been some work on morphology-based segmentation for machine translation in Arabic, German, Uyghur, and Turkish [34, 74, 75]. To the best of our knowledge, there is no work involving morphological tokenization in the abstractive summarization task, which is different from machine translation in the sense that the input and output texts are from the same language and are not sentences but much longer texts. For Hungarian, splitting of the inflectional suffixes and removing the remaining morphemes in order to reduce word perplexity in language modelling was proposed [76]. However, such an approach that discards affixes is not applicable to tasks such as abstractive summarization or neural machine translation, where the outputs of the decoder must be words in surface form.

4. MORPHOLOGY-BASED ABSTRACTIVE TEXT SUMMARIZATION AND LARGE-SCALE DATASETS FOR AGGLUTINATIVE LANGUAGES TURKISH AND HUNGARIAN

4.1. Introduction

In this chapter, we curate two large-scale datasets (TR-News and HU-News) that can serve as benchmarks in the abstractive summarization task for Turkish and Hungarian. The datasets are primarily compiled for text summarization, but are also suitable for other tasks such as topic classification, title generation, and key phrase extraction.

Morphology is important for these agglutinative languages since meaning is carried mostly within the morphemes of the words. We utilize these morphological properties for tokenization to retain the semantic information and reduce the vocabulary sparsity introduced by the agglutinative nature of these languages. Using the datasets compiled, we propose linguistically-oriented tokenization methods (SeperateSuffix and CombinedSuffix) and evaluate them on the state-of-the-art abstractive summarization models. The SeperateSuffix method achieves the highest ROUGE-1 score on the TR-News dataset and provides promising results on the HU-News dataset. In another experiment, we show that the multilingual cased BERT model outperforms monolingual BERT models for both languages and reaches the highest ROUGE-1 score on the HU-News dataset. Lastly, we provide qualitative analysis of the generated summaries on the TR-News dataset.

More specifically, the contributions of this chapter are summarized below:

(i) We release two large-scale publicly available text summarization datasets[12] for two resource-scarce agglutinative languages, Turkish and Hungarian. The datasets also contain information that can be used in other tasks such as title generation, topic classification, key phrase extraction, and author detection. To the best of our knowledge, HU-News is the first large-scale summarization dataset for Hungarian.

(ii) We provide strong baselines for both datasets.

(iii) Two types of morphological tokenization approaches (SeperateSuffix and CombinedSuffix) are proposed for both Turkish and Hungarian. Through these tokenization methods, the effect of morphology is studied on both datasets. The SeperateSuffix method achieves the highest ROUGE-1 F-Score on the TR-News dataset amongst all the models used in this study. The code for morphological tokenization is made publicly available [13] .

(iv) In addition to the pointer-generator model, we use a BERT-based summarization approach and test it with multilingual and monolingual BERT models. It is shown that the multilingual cased BERT model outperforms the monolingual BERT models and achieves the highest ROUGE-1 F-Score on HU-News amongst all the models used in the study.

This chapter is organised as follows. In Section 4.2, the dataset construction phase and statistics about the datasets are presented in detail. Section 4.3 provides explanations for the summarization models and the morphological tokenization methods used in this study. Experimental setups are given in Section 4.4. Both the quantitative and the qualitative results of the experiments are presented in Section 4.5, which is followed by conclusions in Section 4.6.

[12]https://github.com/batubayk/datasets
[13]https://github.com/batubayk/MorphologicalTokenizers

4.2. Datasets

In NLP, most of the research is based on the English language. Accordingly, more resources are available for English compared to the other languages. This is an obstacle for progressing and carrying out recent research methodologies in other languages with scarce data resources. Text summarization is also suffering from data scarcity and it is an important but expensive process to create such resources. In Turkish text summarization, almost all of the works done are in extractive manner where very limited data resources are utilized. Most datasets contain a total of 50-120 data samples where none of them acts as a benchmark and are rather used once per study. Similar remarks are also valid for Hungarian where even less study is done in text summarization.

The sizes of the datasets are especially critical for abstractive summarization where mostly deep learning-based approaches are utilized. A large amount of data is needed to be able to train an adequate model which is capable of generating reasonable summaries. To tackle this problem, two large-scale datasets (in Turkish and Hungarian) aimed for text summarization were prepared and released within the scope of this work. Moreover, both datasets were compiled in a manner to make them suitable for other NLP tasks such as topic classification, author identification, headline generation, and various others. In this section, the dataset preparation methodology and statistical information regarding the datasets will be provided.

4.2.1. Dataset Compilation

Newspapers are valuable sources of information for numerous NLP tasks. In text summarization, newspapers are also important since such sources provide data in public domain and are easily accessible. In English text summarization, almost all corpora are based on news articles such as DUC-2003[14] , DUC-2004[15] , and other larger corpora that are more recent and commonly used in abstractive text summarization;

[14]https://duc.nist.gov/duc2003
[15]https://duc.nist.gov/duc2004

Gigaword [5], XSum [56], NY Times [77], and CNN/Daily Mail [11]. In this work, a similar approach used in the compilation of the CNN/Daily Mail dataset was adopted. The news articles with highlights/abstracts were selected and abstracts were used as reference summaries.

Firstly, the agglutinative languages and sources from which the datasets were going to be collected have been determined. Turkish was the first chosen language since it was native to the authors. Other candidate languages were Hungarian, Czech, and Finnish since these are the mostly studied agglutinative languages in the literature. All the publicly available newspapers for these languages were gathered from Wikipedia and were sorted according to the establishment dates in ascending order (assuming more data would be available for older ones) and the ones that did not have websites or required subscriptions were eliminated. Next, websites that did not have the abstract field in the articles were eliminated. Articles were randomly sampled from the remaining news websites to check if the abstract field was consistently available in most of the articles and the inconsistent websites were eliminated. Additionally, content and abstract lengths, HTML markup quality, accessibility, and also content diversity and quality were amongst the elimination criteria. At the end of these processes, there were three news sites from both Turkish and Hungarian that were suitable for curating the datasets.

A web crawler that was capable of continuously gathering news articles from the selected web sources and automatically extracting several selected fields that were mostly common to all the sources was implemented. These fields were URL, title, abstract, content, date of publish, author, source, topic, and tags. Figure 4.1 shows an overview of the crawling process. The Scrapy [16] tool was used as the crawler framework. A separate spider was implemented for each website. Each spider was given the main page of the website as the seed and links were extracted through the Link Extractor module. The responses were parsed and the outgoing links within the requested page have been extracted. Only the links that belong to the website domain

[16]https://scrapy.org/

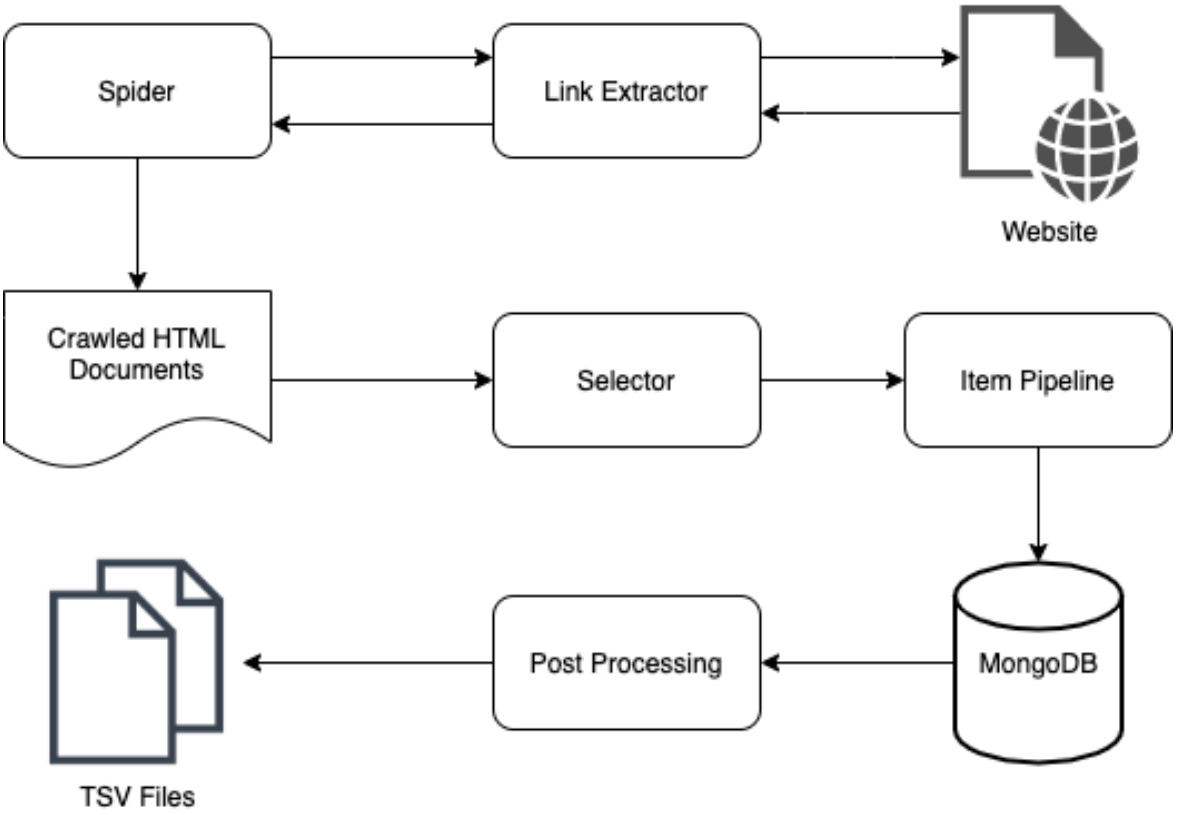

Figure 4.1. Crawling process using Scrapy

were used, others were eliminated. Once the spider extracted the HTML content from the responses, Selector module was utilized to extract the necessary fields from the webpages. XPath and CSS selectors were used in the process of extraction. The extracted pages were converted into objects which are called Items. Each item is a custom defined object where in our case an item is a news document consisting of the fields (e.g url, title, content) stated earlier. For storage, MongoDB[17] , a No-SQL database, has been utilized. Item pipeline was responsible for inserting each item object into the database. The whole data collection process took nearly one month.

Once all the data had been collected, the documents were further processed to eliminate the ones which had missing values in their content or abstract fields. This operation was essential for data integrity of the text summarization task. Lastly, the data was exported into TSV (tab-seperated values) files for easier use. The code of the

[17]https://www.mongodb.com/

crawler[18] and datasets are publicly available.

Table 4.1. Comparison of summarization datasets with respect to overall corpus size, sizes of training, validation, and test sets, average content and abstract lengths (in terms of words and sentences)

Datasets	Num docs (train/val/test)	Avg. content length		Avg. abstract length	
		words	sentences	words	sentences
CNN/Daily Mail	287,226/13,368/11,490	766.00	29.74	53.00	3.72
NY Times	589,284/32,736/32,739	800.04	35.55	45.54	2.44
XSum	204,045/11,332/11,334	431.07	19.77	23.26	1.00
TR-News	277,573/14,610/15,379	290.93	16.03	25.74	1.49
HU-News	211,860/11,151/11,738	423.89	17.78	36.73	1.88

Table 4.2. Comparison of summarization datasets with respect to vocabulary size and type-token ratio of both content and abstract.

Datasets	Vocabulary size		Type-token ratio	
	content	abstract	content	abstract
CNN/Daily Mail	869,526	240,868	0.0036	0.0146
NY Times	1,399,358	294,011	0.0027	0.0099
XSum	399,147	81,092	0.0041	0.0154
TR-News	990,863	230,772	0.0111	0.0292
HU-News	1,900,854	439,441	0.0191	0.0510

4.2.2. Statistics

Both the Turkish and the Hungarian datasets (which we refer to as TR-News and HU-News, respectively) have been collected primarily for the text summarization task. However, both datasets contain additional fields such as topic, title, and tags that can be used in other tasks like topic classification, title generation, and key-phrase (tag) extraction, respectively.

TR-News was constructed from three different sources: NTV, Cumhuriyet, and Habertürk websites. We collected a total of 307,562 articles with the article count from

[18]https://github.com/batubayk/newscrawler

each source being, respectively, 222,301, 44,990, and 40,271. The articles' date varies in the range of 2009-2020. The dataset contains articles from a total of 121 various different domains (e.g. Domestic, World, Sports, Economy, Health, Life, Art, Technology, Education, Politics) where some being sub-categories of others. The dataset was randomly split into training (90%, 277,573), validation (5%, 14,610), and test (5%, 15,379) sets.

HU-News was also constructed from three different sources: Origo, Magyar Nemzet, and Index websites. A total of 234,749 articles were collected where the count per source is, respectively, 152,129, 73,017, and 9,603. The dates of the articles vary from 2009 to 2020 and the articles are categorized into a total of 49 various different domains (e.g. Economy, World, Domestic, Tech, Culture, Autos) where some being synonyms of each other. The dataset was randomly split into training (90%, 211,860), validation (5%, 11,151), and test (5%, 11,738) sets.

Table 4.1 compares TR-News and HU-news with the CNN/Daily Mail, NY Times, and XSum benchmark datasets. It is evident that both TR-News and HU-News contain a substantial amount of instances, where TR-News has a similar size as the CNN/Daily Mail dataset and HU-News as the XSum dataset. The content and abstract lengths of both datasets are around the half of the sizes of CNN/Daily Mail and NY Times both in terms of words and sentences. This is partly due to the agglutinative nature of these languages where the same information can be expressed with fewer words when compared to other languages such as English. Moreover, the amount of information entered into the websites also limits the content and abstract lengths of the datasets.

In Table 4.2, it can be seen that the vocabulary size (number of distinct words) of both TR-News and HU-News are much higher than the English datasets when the dataset sizes are taken into consideration. This is also due to the agglutinative nature of the languages where a very large number of words can be derived from a single root form. Type-token ratio (TTR) is a simple measure of lexical diversity and is used in the approximation of morphological complexity of languages [78]. TTR is calculated by

dividing the vocabulary size to the total number of words. As expected, Turkish and Hungarian datasets show much higher TTR ratio when compared with all the other datasets.

Table 4.3. Two news articles selected from TR-News and HU-News. All the collected fields are shown: URL, title, abstract, content, topic, tags, date, author, and source.

	TR-News	HU-News
URL	https://www.haberturk.com/buyuk-onder-ataturk-un-ebediyete-intikalinin-80-inci-yili-2214706	https://www.origo.hu/techbazis/20200708-nyujthato-kijelzo-telefon-laptop-hajtogathato.html
Title	Büyük önder Atatürk'ün ebediyete intikalinin 80'inci yılı	A nyújtható kijelző lehet a következő nagy csoda
Topic	gündem	techbazis
Tags	['büyük önder atatürk', 'türkiye cumhuriyeti', '10 kasım', 'mustafa kemal atatürk', '10 kasım haberleri']	['laptop', 'telefon', 'nyújtható kijelző', 'kihajtható kijlező']
Date	10.11.2018 - 01:47	2020.07.09. 10:23
Author	AA	Haraszi Tibor
Source	haberturk	www.origo.hu

Table 4.3 (cont.)

Abstract	Türkiye Cumhuriyeti'nin kurucusu, cesur ve unutulmaz önderi Mustafa Kemal Atatürk, Kurtuluş Savaşı'nı başarıyla yöneten komutan olmasının yanı sıra gerçekleştirdiği devrimlerle de dahi bir devlet adamı olarak tarihe geçti. Mustafa Kemal Atatürk'ü sevgiyle ve özlemle anıyoruz...	A színes, nagy felbontású és jó képminőséget produkáló kijelzők mostanra elválaszthatatlanná váltak a hétköznapoktól. És nem csak a tévékre, számítógépekre gondolunk ilyenkor, hiszen ilyen megjelenítők találhatóak a telefonokban, az okosórákban, az okoskarkötőkben, és egyre inkább meghódítják a járműipart, a háztartási gépeket, sőt felbukkantak már a villamosmegállókban, boltok kirakatában is. Ez pedig még csak a kezdet.

Table 4.3 (cont.)

Content	Türkiye Cumhuriyeti'nin kurucusu, cesur ve unutulmaz önderi Mustafa Kemal Atatürk, 80 yıl önce 10 Kasım 1938'de Dolmabahçe Sarayı'nda saat dokuzu beş geçe hayata gözlerini yumdu. Atatürk'ün vefatı Anadolu'nun yanı sıra bütün dünyada da üzüntüyle karşılandı. Mustafa Kemal Atatürk'ü sevgiyle ve özlemle anıyoruz... Büyük Önder Atatürk, 1881'de Selanik'te halihazırda müzeye dönüştürülen üç katlı evde doğdu. Babası Ali Rıza Efendi, annesi Zübeyde Hanım olan Atatürk, ilkokulu Selanik'te Şemsi Efendi Mektebi'nde okudu. Öğrenimini Selanik Askeri Rüştiyesi ve Manastır Askeri İdadisi'nde sürdüren Atatürk, 1899'da girdiği İstanbul Harbiye Mektebi'ni 1902 yılında piyade teğmeni rütbesiyle Harp Akademisi'ni de 1905'te kurmay yüzbaşı olarak bitirdi. (...)	Egyre többen foglalkoznak a korábbi korlátokon átlépő speciális kijelzők fejlesztésével, amelyek korábban sosem látott területekre vihetik el az ilyen eszközöket. Így születtek meg a feltekerhető és hajlítható képernyők, amelyek elképesztő lehetőségeket nyitottak meg a mérnökök és dizájnerek előtt. A nyújtható kijelzők pedig egy újabb szintlépést jelenthetnek. Ez nem egy idétlen sci-fi film, hanem a valóság. Ténylleg számíthatunk a nyújtható képernyők megjelenésére, ráadásul már a közeljövőben. A dél-koreai állam kezdeményezésére létrejött egy munkacsoport, amelynek feladata ezen különlegesnek ható technológia kifejlesztése. A projekt vezetésével a hatalmas LG konglomerátum kijelzőgyártó és fejlesztő leányvállalatát, az LG Displayt bízták meg. (...)

The novelty ratio is the percentage of the number of words in reference summaries that do not occur in the source documents. It is usually used for assessing the abstractiveness of the datasets. Figure 4.2 shows the novelty ratios in terms of n-grams

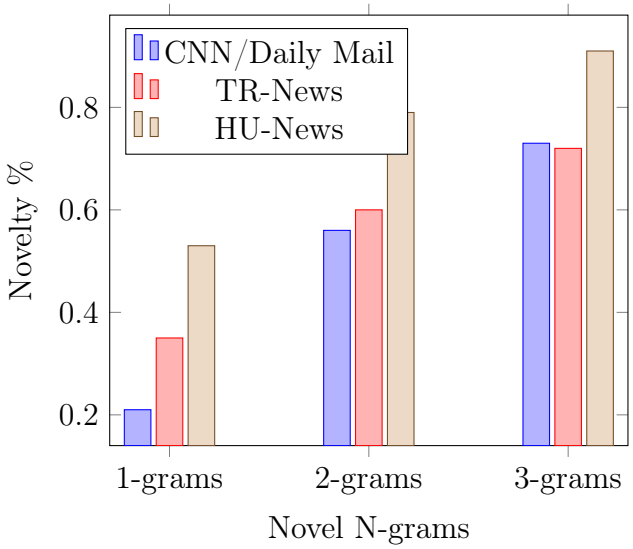

Figure 4.2. N-gram novelty comparison between content and abstract

(1-grams, 2-grams, and 3-grams) for the datasets used in this work. As can be seen, CNN/Daily Mail contains the least amount of novel unigrams, followed by TR-News and then HU-News. The Hungarian dataset has the highest novelty ratio which partially means that it would be the most challenging dataset amongst others. This is especially the case for the models that tend to copy text from the input such as the pointer-generator network (see Section 4.3.1.1) used in this study. Consequently, such features of agglutinative languages discussed in this section bring higher complexities when dealing with NLP tasks [39].

Lastly, an example article from each dataset is shown in Table 4.3. Most of the fields in the table are self-explanatory. Each news has a single topic and may have multiple tags. Tags can be considered as key phrases that can be extracted from the news content. The author field contains the author of the news article, although in some cases it is populated with news agency names. For the sake of space, the contents of the examples were cropped.

4.3. Methodology

In this section we describe the models and the tokenizers used in this chapter. The morphological tokenization methods proposed for both Turkish and Hungarian are explained in detail.

4.3.1. Models

Two models have been used in this study for text summarization, which are the pointer-generator model and the BERT+Transformer model.

4.3.1.1. Pointer-Generator Network with Coverage Mechanism. As the baseline model we have chosen the pointer-generator model [6]. It is an encoder-decoder network that is capable of deciding whether to point to a word from the input sequence or to generate a new word from the vocabulary as shown in Figure 4.3. The encoder is composed of a bidirectional LSTM and the decoder makes use of a unidirectional LSTM with attention mechanism [8]. The model manages to lower the OOV (out of vocabulary) cases through a mechanism called *copy*. OOV is a critical problem in text generation tasks. This model manages to overcome this issue and is also able to generate abstractive summaries although copying from the source text is allowed.

$$p_{gen} = \sigma(W_c^T c_t + W_s^T s_t + W_x^T x_t + b_{ptr}) \tag{4.1}$$

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \tag{4.2}$$

At a timestep t, the probability of generating a vocabulary token (p_{gen}) is calculated by applying a sigmoid function after adding the dot products of the context vector

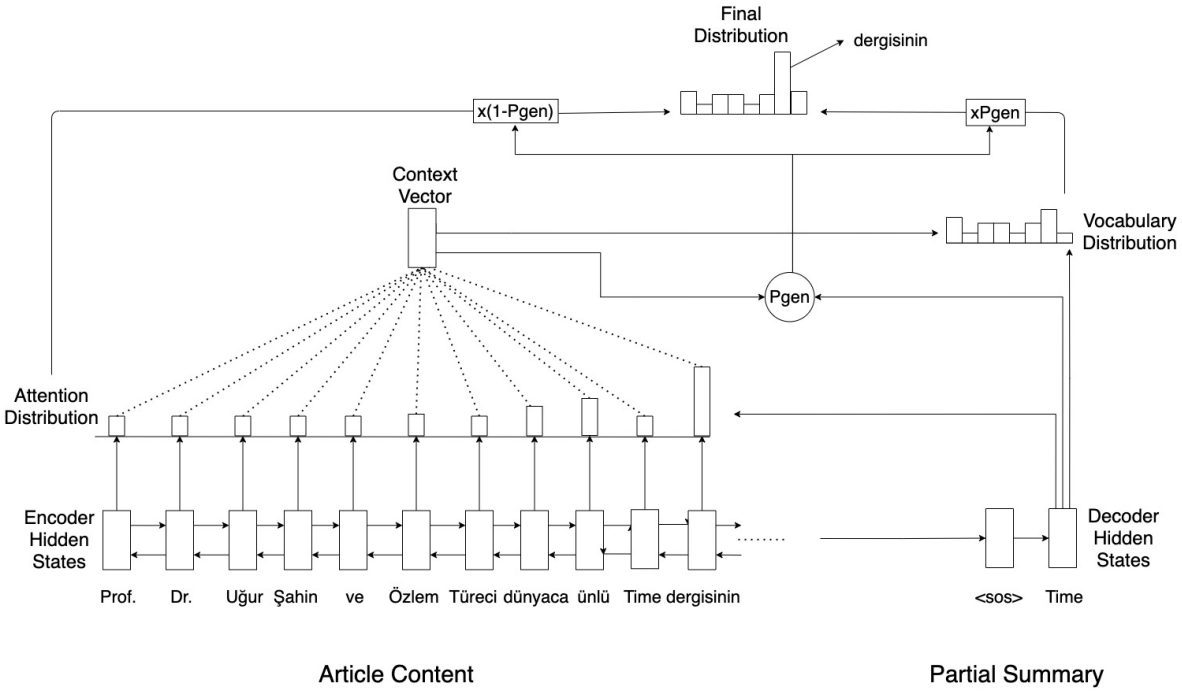

Figure 4.3. Pointer-generator network

c_t, the decoder state s_t, the decoder input x_t (decoder output of the previous timestep) with their respective weight matrices and adding a bias b_{ptr}. The final vocabulary distribution $P(w)$ is calculated as a weighted sum controlled by the p_{gen} parameter. A balance between the probability of choosing a word from the vocabulary (P_{vocab}) or copying a word from the input document based on the attention distribution a^t is achieved.

Seq2Seq models also tend to have repetition problems and to solve this issue, a mechanism called *coverage* is proposed [6]. An additional parameter is introduced to the attention mechanism that is capable of keeping track of the earlier generated tokens and thus reducing repetition. Additionally, the loss function is altered and a separate loss function for coverage is added.

4.3.1.2. BERT + Transformer. Recently, pretrained models have become very effective in NLP. BERT (Bidirectional Encoder Representations from Transformers) [14], pretrained using masked language model and next sentence prediction objectives, particularly has proven to be very successful in many NLP applications, especially in nat-

ural language understanding tasks. In this work, we have utilized an encoder-decoder architecture that makes use of BERT as the encoder and a 6-layered transformer network as the decoder [15]. Two separate optimizers with different initial learning rates are used to avoid unstable model training which might be caused due to the encoder being a pretrained model and the decoder being a randomly initialized network. The authors propose two different methods to initialize the encoder; the first being a pretrained BERT (BERTSumAbs) and the second being a BERT model finetuned on extractive summarization task (BERTSumExtAbs). We follow the first approach and use the standard pretrained BERT models in our experiments.

4.3.2. Morphological Tokenizers

In this work, we implement two different morphology-based tokenizers for Turkish and Hungarian. The approaches we use are more linguistically-oriented when compared to unigram LM and BPE. Rather than splitting the word based on statistical methods, we aim to leverage the true morphological structure within the words. In essence, this is what subword methods are also trying to achieve. The two methods implemented in this study are both based on the roots of the words and the suffixes. In the first method (SeperateSuffix) all morphemes (root and suffixes) are considered separately, whereas in the second one (CombinedSuffix) the word is divided into two parts as the root and all the suffixes in concatenated form. There could have been other tokenization strategies such as only using inflectional suffixes or derivational suffixes, but since the tokens need to be reconstructed in full form during generation these methods are not applicable to language generation tasks.

The morphological tokenization process is formed of three main components; vocabulary construction, text encoding, and text decoding. Firstly, the vocabulary needs to be constructed from the corpus. In this phase, sentence detection algorithms are used to extract the sentences from the text and each sentence is fed to the morphological analysis tools. In most cases, the words contain ambiguity in the sense that a word can be split into various combinations of morphemes. These ambiguities need

to be resolved in order to obtain the correct root form and suffixes. In our case, we used the Zemberek tool which is able to automatically solve such ambiguities for Turkish. However, for Hungarian the tools did not automatically solve the morphological ambiguities. Thus, we matched POS tags annotated using the tool PurePOS with the possible morphological parses obtained from the tool emMorph in order to disambiguate the morphemes. Once the morphological analysis and disambiguation is completed, depending on the tokenization approach stated above, each morpheme is processed separately or suffixes within the word are concatenated.

As the roots and the suffixes are collected, they are sorted in descending order with respect to their frequencies in the corpus and the top n entries are taken to form the vocabulary. In this way, the vocabulary is built to represent the most frequent roots and morphemes within the corpus. Splitting the words into morphemes reduces data sparsity and increases coverage when compared to standard whitespace tokenization. Although being able to represent the vast majority of the tokens in the language, these methods cannot handle the OOV problem as efficiently as BPE or unigram LM.

The second component, encoding, makes use of the same sentence detection and morphological analysis steps to obtain the morphemes. Then these units are looked up in the vocabulary and for each one an id is returned if it exists in the vocabulary and the id of the token used to represent the unknown words is returned otherwise. The last component, decoding, takes in a sequence of ids and constructs a text by concatenating the morphemes corresponding to these ids as encoded during the encoding process. If an unknown word id occurs in the encoded input, the predefined token for the unknown words (e.g. [UNK]) is output. This process is responsible for constructing the morphologically correct surface form of the words that are decoded.

Lastly, although English is not an agglutinative language, we aimed to observe if splitting the words in English would result in a similar effect to that of morphology. Accordingly, we used the Porter stemmer algorithm [51] implemented in NLTK to split the words into two parts (as the stem and the suffixes) and refer to this method as

StemSuffix. The three tokenization components are identical to the morphology-based methods described earlier. The only difference is the morphological disambiguation step. It is not required since the stemmer returns a unique form.

The tools and the methods used in this process for Turkish and Hungarian are explained further below.

Table 4.4. Turkish morphological tokenization methods

Method	Operation	Output
		Sentence: şampiyon yüzücünün abd kongre baskınındaki görüntüleri ortaya çıktı. (The photos of the champion swimmer taken during the US congress raid have been revealed.)
SeperateSuffix	tokenize	['şampiyon', 'yüz', '##ücü', '##nün', 'abd', 'kongre', 'baskın', '##ı', '##nda', '##ki', 'görüntü', '##ler', '##i', 'orta', '##ya', 'çık', '##tı', '.']
	encode	[829, 234, 2317, 522, 207, 1158, 2151, 7, 21, 49, 927, 10, 8, 241, 57, 107, 69, 6]
	decode	şampiyon yüzücünün abd kongre baskınındaki görüntüleri ortaya çıktı .
CombinedSuffix	tokenize	['şampiyon', 'yüz', '##ücünün', 'abd', 'kongre', 'baskın', '##ındaki', 'görüntü', '##leri', 'orta', '##ya', 'çık', '#tı', '.']
	encode	[823, 201, 2342, 183, 1321, 2347, 272, 964, 23, 220, 52, 77, 106, 5]
	decode	şampiyon yüzücünün abd kongre baskınındaki görüntüleri ortaya çıktı .

4.3.2.1. Turkish. Zemberek [79] is an open source NLP framework for Turkic languages. It provides a range of capabilities such as sentence boundary detection, tokenization, text normalization, and morphological parsing and disambiguation. As

stated earlier, the morphological parser is capable of automatic disambiguation (see Appendix A for detailed examples). Table 4.4 shows an example that illustrates the three components of both of the tokenization methods. First the input is tokenized into morphemes where each suffix is concatenated with a prefix ## (similar to WordPiece) to denote that it is not the root form. This information is utilized during decoding. The encoding step assigns an id to each token from the vocabulary. Lastly, the decoding step concatenates all the subwords containing ## and forms an output.

Table 4.5. Hungarian morphological tokenization methods

Method	Operation	Output
		Sentence: a tanulók igényeihez kell igazodniuk a nyelvvizsga követelményeinek is . (The requirements of the language exam must also be adapted to the needs of the students.)
SeperateSuffix	tokenize	['a', 'tanuló', '##k', 'igény', '##ei', '##hez', 'kell', 'igazod', '##niuk', 'a', 'nyelvvizsga', 'követelmény', '##ei', '##nek', 'is', '.']
	encode	[4, 3297, 23, 939, 112, 179, 94, 6026, 1241, 4, 13435, 3760, 112, 33, 18, 6]
	decode	a tanulók igényeihez kell igazodniuk a nyelvvizsga követelményeinek is .
CombinedSuffix	tokenize	['a', 'tanuló', '##k', 'igény', '##eihez', 'kell', 'igazod', '##niuk', 'a', 'nyelvvizsga', 'követelmény', '##einek', 'is', '.']
	encode	[4, 3316, 31, 939, 5889, 74, 6236, 1720, 4, 14658, 3803, 823, 15, 6]
	decode	a tanulók igényeihez kell igazodniuk a nyelvvizsga követelményeinek is .

4.3.2.2. Hungarian. For Hungarian we first tried to utilize the e-magyar system [80–85] which provides tools such as tokenizer, morphological analyser, POS tagger, dependency parser, and NP (Noun Phrase) chunker. However, processing the texts took noticeably long computation times although we were only utilizing the sub-components we required, which are the morphological analyser and the POS tagger. Hence, we decided to use the underlying components as standalone libraries instead of using the e-magyar system. Consequently, we implemented our Hungarian tokenizers by making use of the emMorph and PurePOS tools where the first is the morphological analyzer and the latter is the POS tagger. A performance increase was obtained in terms of computation time. The tokenization process for Hungarian is composed of the same steps as the Turkish tokenization and an example of morphological disambiguation is given in the Appendix A. Table 4.5 displays an example for both of the tokenization methods.

4.4. Experimental Settings

In this work, we performed two set of experiments. The first experiment aims to measure the effect of morphology in abstractive summarization by utilizing various tokenization methods. The second experiment evaluates the performance of a state-of-the-art abstractive summarization model on Turkish and Hungarian. For both experiments, TR-News, HU-News, and CNN/Daily Mail datasets have been used.

4.4.1. Experiment 1 - Pointer-Generator Network and Morphological Tokenizers

The pointer-generator model [6] has been proven to be a very effective model based on the LSTM encoder-decoder architecture with copy and coverage mechanisms. In the original model, whitespace tokenization has been used. In this experiment, in addition to the whitespace tokenization that serves as a baseline, we use three other tokenization methods. Two of them (SeperateSuffix and CombinedSuffix) are linguistically-oriented and one (unigram LM) is statistical but capable of capturing

morphological information. Importantly, the model does not require any costly pretraining operation when the tokenizer changes unlike the BERT-based models. Therefore, this model is more appropriate for morphological experiments given the computational resource and time restrictions of pretraining a BERT model.

Following the original paper [6], we set the hidden unit size of the models as 256 and the embedding dimension as 128. Similarly, the vocabulary size was chosen as 50K for a fair comparison. The optimization method was changed from Adagrad [86] to Adam [87] due to unstable loss values in training for both TR-News and HU-News. The learning rate was set to 0.001 and gradient clipping with gradient norm 2 was utilized. The best models were chosen based on the validation losses and early stopping was employed. During training, the encoder was limited to 400 tokens and the decoder to 100. At test time the decoding limit was increased to 120 tokens. The authors have stated that such an option would lead to performance gains [6].

Each model was trained on a single Tesla V100 GPU with a batch size of 32. Training the models with morphological tokenizers took more computation time compared to whitespace and unigram LM models. Moreover, Hungarian morphological models took more time (7 days for 7 epochs) when compared to Turkish morphological models (1 day for 7 epochs) due to the performance difference in tokenization.

4.4.2. Experiment 2 - BERT-based Abstractive Text Summarization

In order to see the performance of a state-of-the-art summarization model and compare its performance with the baseline model, we conducted a second set of experiments on the Turkish and Hungarian languages. We utilized the multilingual uncased BERT model but we suspected possible character encoding problems, thus we made additional evaluations with the cased variation of the multilingual BERT model. In addition to the multilingual BERT models, we also experimented with monolingual uncased BERT models which are BERTurk [88] for Turkish and huBERT [89] for Hungarian. Moreover, the small and large variants of BERTurk were also utilized to see if a

larger model with more vocabulary would affect the performance. Unfortunately, there is only a single sized pretrained BERT model for Hungarian so we couldn't conduct vocabulary related experiments for that language.

For this experiment, we used the publicly available code[19] of the study [15] that we follow for our BERT-based models. We followed the same configurations but had to increase the number of warmup steps of the decoder to 20K for the BERTurk-uncased-128k model due to unstable training. All the models were trained for 200K iterations and the best models were chosen according to validation loss. We also had to alter the vocabulary of BERTurk and huBERT to insert special tokens (e.g. start of sentence and end of sentence tokens) since there were no reserved tokens in the vocabularies of these monolingual BERT models, unlike the BERT and multilingual BERT models. Tokens that did not appear in TR-News and HU-News were replaced with our special tokens to avoid collisions during training.

4.5. Results

In this section we evaluate our findings both quantitatively and qualitatively for Experiment 1 and Experiment 2.

4.5.1. Quantitative Analysis

The models described in Section 4.4 were evaluated with the ROUGE metric [35] which is commonly used in text summarization. ROUGE-1, ROUGE-2 and ROUGE-L scores were calculated. The ROUGE-n score measures the informativeness of the generated summaries by counting the number of common n-grams between the generated summary and the reference summary. ROUGE-L calculates the number of n-grams based on the longest common sub-sequences and measures the fluency of the generated summaries. In addition to the ROUGE metrics, we also computed the novelty ratios of the generated summaries in terms n-grams (n=1,2,3). Moreover, the OOV ratios were

[19]https://github.com/nlpyang/PreSumm

calculated for each experiment in order not to misinterpret the novelty ratios that can also be caused by the OOV tokens. We also calculated the commonly-used baselines LEAD-2 and LEAD-3 by selecting, respectively, the first two sentences and the first three sentences in the content.

Table 4.6. Results of pointer-generator models with different tokenization methods on CNN/Daily Mail, TR-News, and HU-News datasets in terms of ROUGE-1, ROUGE-2, and ROUGE-L F-measure. "-" denotes result is not available. Bold values show the highest scores obtained in the experiments per dataset.

Model	CNN/Daily Mail			TR-News			HU-News		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
LEAD-2	38.42	15.74	34.28	31.37	17.91	26.92	24.34	7.87	17.61
LEAD-3	40.43	17.62	36.66	28.64	16.21	24.07	23.7	7.78	16.75
See et al. [6]	39.53	17.28	36.38	-	-	-	-	-	-
Whitespace(No copy)	35.36	14.41	32.65	27.35	14.86	25.8	13.62	2.99	12.43
Whitespace	**39.09**	**17.33**	**35.84**	31.61	18.55	29.57	22.92	7.69	19.78
Unigram LM	38.57	17.16	35.37	33.38	19.77	31.15	**24.33**	**8.25**	**20.91**
StemSuffix	38.60	17.10	35.42	-	-	-	-	-	-
SeperateSuffix	-	-	-	**34.94**	**20.89**	**32.56**	23.86	8.10	20.53
CombinedSuffix	-	-	-	33.93	20.07	31.57	23.57	7.97	20.23

4.5.1.1. Experiment 1 Results. Table 4.6 summarizes the results of the first experiment on the test sets. In the first part of the table, the performance of the LEAD-2 and LEAD-3 baselines are given. The LEAD baselines are commonly-used in text summarization and are considered to be strong baselines. Also the results reported in the reference study [6] are shown for comparison. The second part of the table shows the performances of the tokenization methods. Whitespace refers to the model used in the reference study [6]. Whitespace (No copy) is the same model without the copy mechanism, which is used here to observe the effect of the copy mechanism. SeperateSuffix and CombinedSuffix are the morphology-based tokenization methods proposed in this work for text summarization. We also devised another method, StemSuffix, to observe the effect of morphological tokenization in English. Note that, although Whitespace is a replication of the reference study [6], the results we obtained are slightly differ-

ent than those given in the paper for the CNN/Daily Mail dataset. We believe that the difference is due to using different optimization methods and preprocessing steps. We have made use of NLTK[20] in our preprocessing whereas the authors have utilized the Stanford NLP toolkit[21] . For TR-News, SeperateSuffix tokenization has achieved the highest score with +3.33 and +1.56 ROUGE-1 scores compared to, respectively, whitespace and unigram LM. For HU-News which is a more challenging dataset in terms of vocabulary size and novelty ratio, Unigram LM obtained the highest score and outperformed SeperateSuffix method with 0.47 ROUGE-1 points. These results show that subword methods are effective for both of the agglutinative languages when compared to whitespace tokenization. When we compare the two morphology-based methods, we see that SeperateSuffix outperforms CombinedSuffix method for both Turkish and Hungarian. For English, whitespace tokenization yields the best results and StemSuffix does not contribute to the performance in this language. Moreover, the whitespace method without copy mechanism shows a decrease for all datasets where the performance loss for HU-News being the most dramatic.

Taking the average abstract sentence lengths of TR-News and HU-News into consideration (see Table 4.1), LEAD-2 is a more suitable baseline for these datasets. This claim is supported by the scores in Table 4.6 where the ROUGE-1 score of LEAD-2 is higher than LEAD-3 for both TR-News and HU-News. Lastly, we observe that morphology-based models significantly outperform the LEAD-2 baseline for TR-News whereas the LEAD-2 of HU-News cannot be outperformed by any model in terms of ROUGE-1 score. This outcome also supports the findings in Figure 4.2 which points HU-News being a more challenging dataset.

To assess the abstractiveness of the generated summaries, novel n-grams that occur in the generated summary but not in the corresponding news article are counted. This metric is especially relevant to models that make use of the copy mechanism in abstractive summarization [6, 55], which highly relies on copying text from the input. In Table 4.7, novel n-gram (unigram, bigram and trigram) ratios are given for all

[20]https://www.nltk.org/
[21]https://nlp.stanford.edu/software/

Table 4.7. Novel n-gram ratios for the models in Experiment 1. N1, N2, and N3 respectively represent n-grams (n=1,2,3).

Models	CNN/Daily Mail			TR-News			HU-News		
	N1	N2	N3	N1	N2	N3	N1	N2	N3
Whitespace(No copy)	3.58	14.46	25.4	16.87	37.80	50.95	43.98	84.15	96.75
Whitespace	0.35	02.78	6.72	3.27	7.50	11.35	6.85	16.46	24.29
Unigram LM	0.64	03.92	8.70	4.66	9.90	14.27	8.99	17.44	24.86
StemSuffix	3.68	08.94	15.71	-	-	-	-	-	-
SeperateSuffix	-	-	-	11.06	23.40	31.31	7.10	15.53	22.79
CombinedSuffix	-	-	-	10.17	22.20	30.00	7.49	15.53	22.72

models in Experiment 1. According to the results, all methods perform poorly on CNN/Daily Mail and this aspect is in parallel to the findings of the original paper [6]. The summaries generated on TR-News and HU-News seem to be more abstractive when compared to CNN/Daily Mail. For TR-News, morphology-based tokenizers managed to produce more novel summaries.

Table 4.8. OOV analysis for the models in Experiment 1. OOV and OOV ratio denote, respectively, the average number of OOV words and the percentage of OOV per summary.

Dataset	CNN/Daily Mail		TR-News		HU-News	
	OOV	OOV ratio (summary)	OOV	OOV ratio (summary)	OOV	OOV ratio (summary)
Whitespace(No copy)	1.39	2.62	4.83	8.48	20.75	30.62
Whitespace	0.03	0.05	0.10	0.25	5.07	6.78
Unigram LM	0.03	0.07	0.04	0.10	0.02	0.05
StemSuffix	0.03	0.05	-	-	-	-
SeperateSuffix	-	-	0.01	0.05	1.37	2.28
CombinedSuffix	-	-	0.01	0.03	1.93	2.68

However, the novelty metric can be deceiving in cases where the number of OOV words (the special UNK token) generated in the summary is high, since the UNK token does not appear in the content but is counted as a novel word. This case is reflected in

the novelty ratios for the whitespace method without the copy mechanism. In Table 4.8, the average number of OOV words and the ratio of OOV words per summary are given on the test set of each dataset. OOV ratio is calculated as the number of OOV words over the total number of words in the summary. This value is calculated for each summary and then the average is taken over the test set. The whitespace method with no copy mechanism seems to output high amount of OOV words especially on HU-News which explains the high novelty ratio in this dataset. Besides the model without copy mechanism, the other models do not suffer from OOV on CNN/Daily Mail and TR-News. However, whitespace and morphology-based tokenizers seem to be affected from this issue on HU-News. This might partially explain the SeperateSuffix and CombinedSuffix methods having slightly lower ROUGE scores than Unigram LM on HU-News.

Table 4.9. BERT+Transformer results on CNN/Daily Mail, TR-News, and HU-News datasets in terms of ROUGE-1, ROUGE-2, and ROUGE-L F-Measure. "-" denotes data is not available. Bold values show the highest scores obtained in the experiments per dataset.

Models	CNN/Daily Mail			TR-News			HU-News		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
LEAD-2	38.42	15.74	34.28	31.37	17.91	26.92	24.34	7.87	17.61
LEAD-3	40.43	17.62	36.66	28.64	16.21	24.07	23.7	7.78	16.75
Liu and Lapata [15]	41.72	19.39	38.76	-	-	-	-	-	-
BERT-uncased	41.95	19.24	38.95	-	-	-	-	-	-
mBERT-uncased	-	-	-	21.70	8.95	18.41	21.88	4.51	17.62
mBERT-cased	-	-	-	**30.99**	**18.09**	**26.54**	**26.54**	9.72	**19.51**
BERTurk-uncased-32K	-	-	-	27.4	15.60	23.36	-	-	-
BERTurk-uncased-128K	-	-	-	26.92	15.25	22.96	-	-	-
huBERT-uncased	-	-	-	-	-	-	25.40	**10.03**	18.54

4.5.1.2. Experiment 2 Results. In the second experiment, we test the performance of the BERT-based models on the three datasets. The ROUGE results are shown in Table 4.9. In the first part of the table, LEAD-2, LEAD-3, and the results of the BERTSumAbs model in the reference study [15], which is the model we use in this work, for CNN/Daily Mail are given, as reference. The second part of the table

shows the performances of the BERT-based models implemented. The BERT-uncased model is the replication of BERTSumAbs on the CNN/Daily Mail dataset. For Turkish and Hungarian, we employed both the multilingual BERT models (mBERT-uncased and mBERT-cased) and the monolingual pretrained BERT models (BERTurk and huBERT).

Table 4.10. OOV analysis results for the models in Experiment 2. OOV and OOV ratio denote, respectively, the average number of OOV words and the percentage of OOV per summary and content.

Model	Dataset	OOV	OOV Ratio (Summary)	OOV Ratio (Content)
BERT-uncased	CNN/Daily Mail	0.01	0.01	0.01
mBERT-uncased	TR-News	10.83	15.73	26.45
mBERT-cased	TR-News	0.18	0.003	0.004
BERTurk-uncased-32K	TR-News	1.29	1.76	1.16
BERTurk-uncased-128K	TR-News	1.66	2.19	1.16
mBERT-uncased	HU-News	3.33	9.54	41.99
mBERT-cased	HU-News	0.22	0.004	0.01
huBERT-uncased	HU-News	0.2	0.26	0.46

We see that the multilingual cased BERT model outperforms all the other BERT models for both Turkish and Hungarian. On the other hand, the uncased variation is outperformed by all the models on both datasets with a large margin. We believe the main cause of this to be the difference between the amount of OOV words. Table 4.10 shows the average number of OOV words and OOV ratios of the generated summaries. The OOV ratios in the summaries for Turkish and Hungarian are, respectively, 1.76%/2.19% and 0.26% in monolingual models. Interestingly, the mBERT-cased shows a very small amount of OOV words for both languages, even less than the monolin-

gual models. However, these ratios jump to, respectively, 15.73% and 9.54% for the uncased multilingual model. OOV words can also be present in the source documents (content) if a specific word or a subword is not present within the tokenizer vocabulary. The table also shows the OOV ratios for the source documents, which are even higher than the ratios of the summaries for the multilingual uncased models. These figures explain the low ROUGE scores obtained for the multilingual uncased models. An interesting observation is that the 32K model gives better results than the 128K model for Turkish. Table 4.10 shows that both models have the same OOV ratios for the source documents, but the larger model has produced summaries with a higher OOV ratio. This may be attributed to the case that 32K model might have fitted the dataset better.

We observe that the best BERT models for Hungarian and English outperform both LEAD baselines and the pointer-generator models (see Table 4.6). This is not the case for Turkish where the best BERT model falls behind the best pointer-generator model with more than 3.95 ROUGE-1 scores.

Table 4.11. Novel n-gram ratios for the models in Experiment 2. N1, N2, and N3 respectively represent n-grams (n=1,2,3).

Models	CNN/Daily Mail			TR-News			HU-News		
	N1	N2	N3	N1	N2	N3	N1	N2	N3
BERT-uncased	1.52	10.81	19.50	-	-	-	-	-	-
mBERT-uncased	-	-	-	11.38	26.34	37.63	16.28	50.49	70.80
mBERT-cased	-	-	-	10	21.33	30.29	17.60	35.32	49.64
BERTurk-uncased-32K	-	-	-	17.03	32.25	43.07	-	-	-
BERTurk-uncased-128K	-	-	-	21.79	39.23	51.21	-	-	-
huBERT-uncased	-	-	-	-	-	-	26.29	46.25	61.23

Table 4.11 shows the novel n-gram ratios for the BERT-based models. When compared with Table 4.7, we see that BERT-based models are able to produce more novel summaries, especially for Turkish and Hungarian. In our observations we note that the results of the mBERT-uncased model are misleading on both datasets due to high number of OOV words which increase the novelty ratio. However, other mod-

els including mBERT-cased correctly reflect their novelty ratios. It is notable that monolingual models are able to generate summaries with higher novelty ratios when compared to multilingual models.

4.5.2. Qualitative Analysis

In addition to the quantitative analysis, we provide a qualitative analysis on the generated summaries for both experiments. Quantitative analysis yields numerical values related to the informativeness and fluency of the summaries, however such results do not reflect well the coherence and cohesion. Hence, we provide here an analysis of example summaries from the TR-News dataset.

Table 4.12. First example document and generated summaries from TR-News for qualitative analysis.

Content	detroitli baba hal vaughen , 24-25 aralık noel tatilini kızı pierce vaughan ile birlikte geçirmek istiyordu . ancak kızının bu iki günde de çalışmak zorunda olması üzerine kızını yalnız bırakmamamak için farklı bir çözüm buldu ve kızının görevli olduğu 6 uçuşun tamamına bilet aldı . 2 gün içinde 6 kez uçak yolculuğu yapan baba hal 'ın bu jesti detroit uçuşunda yanına oturan mike levy isimli yolcunun facebook 'taki paylaşımıyla viral oldu . mike 'ın " eve dönüş yolculuğumda hal 'ın yanına oturma onurunu yaşıyorum . hal , noel 'de çalışma zorunda kalan uçak hostesimiz pierce 'ın babası . noel tatilini kızıyla birlikte geçirmeye karar vermiş . bugün ve yarın kızının görevli olduğu tüm uçuşlarda onunla birlikte . ne harika bir baba ! " paylaşımı 38 bin kezden daha fazla paylaşıldı
Reference	abd 'de bir baba , noel tatilini kızıyla birlikte geçirebilmek için kızının delta havayolları 'nda hosteslik yaptığı tüm uçuşlara bilet aldı .

Table 4.12 (cont.)

Whitespace	**ingiltere 'nin başkenti londra 'da yaşayan** bir **adam** , noel tatilini kızı pierce vaughan ile birlikte geçirmek istiyordu . ancak kızının bu iki günde de çalışmak zorunda olması üzerine kızını yalnız bırakmamamak için farklı bir çözüm buldu ve kızının görevli olduğu 6 uçuşun tamamına bilet aldı
UnigramLM	detroitli baba hal vaughan 'ın kızı pierce vaughan 'ın kızı pierce vaughan , facebook 'taki paylaşımıyla viral oldu . 'ın bu jesti detroit uçuşunda yanına oturan mike levy isimli yolcunun facebook 'taki paylaşımı 38 bin kezden daha fazla paylaşıldı
SeperateSuffix	detroitli baba hal vaughen , 24-25 aralık noel tatilini kızı pierce vaughan ile birlikte geçirmek istiyordu ve kızının görevli olduğu 6 uçuşun tamamına bilet aldı
CombinedSuffix	noel tatilini kızı pierce vaughan ile birlikte geçirmek **isteyen** detroitli baba hal vaughen , kızının görevli olduğu 6 uçuşun tamamına bilet aldı .
BERTurk-uncased-32k	**abd** 'li baba hal vaughen , noel tatilini kızı pierce vaughan ile birlikte geçirmek istiyordu . kızını yalnız bırakmamamak için farklı bir çözüm buldu ve kızının görevli olduğu 6 uçuşun tamamına bilet aldı . bu jesti bir yolcunun facebook 'taki paylaşımıyla viral oldu . paylaşımı 38 bin kezden daha fazla paylaşıldı . . **paylaşımında** " ne harika bir baba ! " **ifadesine yer verildi** .

Table 4.12 (cont.)

BERTurk-uncased-128k	**abd** 'li baba , noel tatilinde kızını yalnız bırakmamamak için farklı bir çözüm buldu ve kızının görevli olduğu 6 uçuşun tamamına bilet aldı . bu **hediye** 38 bin kezden daha fazla paylaşıldı . **paylaşımında** , " eve dönüş **kalkışında** hal 'ın yanına oturma onurunu yaşıyorum " **ifadelerine yer verildi** . . paylaşımı 38 bin kez **tıklandı** [UNK] 'ın paylaşımı **sosyal medyada milyonlarca kez tıklandı** ! [UNK] . . [UNK] aralık 'ta kızını yalnız **bırakmamaması** için **özel** uçak yolculuğu yapan baba ise bu kez de **ilginç** bir **yöntem** facebook 'taki paylaşımıyla viral
mBERT-uncased	2 [UNK] [UNK] 6 kez [UNK] [UNK] yapan hal vaughen 'ın bu jesti facebook 'taki [UNK] viral oldu . [UNK] 38 bin kezden daha fazla [UNK] [UNK] . hal 'ın kızı pierce 'ın babası , noel 'de [UNK] zorunda kalan [UNK] hostesimiz pierce 'in babası . ne harika bir baba ! [UNK] dedi . **peki** hal 'ın **neler** ? hal 'ın yanına oturma onurunu ? hal , noel tatilini kızıyla birlikte karar ve **yarı**
mBERT-cased	**amerika birleşik devletleri 'nde** noel tatilini kızıyla birlikte geçirmek **isteyen** baba hal vaughen , kızının bu iki günde de çalışmak zorunda olması üzerine kızını yalnız bırakmamak için farklı bir çözüm buldu ve kızının görevli olduğu 6 uçuşun tamamına bilet aldı . kızının bu jesti detroit uçuşunda yanına oturan mike levy isimli yolcunun facebook 'taki paylaşımıyla viral oldu

Table 4.13. Second example document and generated summaries from TR-News for qualitative analysis.

Table 4.13 (cont.)

Content	peru 'da , 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . peru kültür bakanlığı tarihin tam olarak doğrulanması halinde , söz konusu tapınağın , dünyanın en eski yerleşim yerleri arasına girebileceğini açıkladı . bakanlık , keşfin çok önemli olduğuna vurgu yaparak , lima ve çevresinin insanlığın en eski tarihlerinde bir uygarlık merkezi olabileceğini belirtti . el paraiso piramidinin kanatlarından birini oluşturan ve arkeologlarca , " ateş tapınağı " adı verilen tapınakta bulunan fırının , ateş yakma törenlerinin işaretçisi olduğuna inanılıyor . araştırma ekibinden marco guillen , o dönemde insanların tanrıyla duman aracılığıyla iletişim kurduğunu ve fırının da bu amaçla kullanılmış olabileceğini kaydetti . peru 'nun tam orta noktasında yer alan 50 hektar alana sahip el paraiso , dünyanın en büyük yerleşim alanlarından biri olarak kabul ediliyor .
Reference	perulu arkeologlar , başkent lima 'nın kuzeyindeki ünlü el paraiso arkeolojik yerleşim alanında 5000 yıl öncesine ait bir tapınak keşfetti .
Whitespace	peru 'da , 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . , dünyanın en eski yerleşim yerleri arasına girebileceğini açıkladı . , dünyanın en büyük yerleşim alanlarından biri olarak kabul ediliyor .
UnigramLM	peru 'da , 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . tarihin tam olarak doğrulanması halinde , söz konusu tapınağın , dünyanın en eski yerleşim yerleri arasına girebileceğini açıkladı .
SeperateSuffix	peru ' da 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . , dünyanın en eski yerleşim yerleri arasına **girebilecek** .

Table 4.13 (cont.)

CombinedSuffix	peru ' da , 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . , dünyanın en büyük yerleşim alanlarından biri olarak kabul ediliyor .
BERTurk-uncased-32k	peru 'da , 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . **tapınaktaki** fırının , ateş yakma törenlerinin işaretçisi olduğuna inanılıyor . . [UNK] **yılları arasında** peru **'daki** el paraiso piramidinin kanatlarından birini oluşturan ve arkeologlarca , " ateş tapınağı " adı verilen fırının , dünyanın en eski yerleşim yerleri arasına **girebileceği belirtildi** .
BERTurk-uncased-128k	peru 'da , 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . . **mezarlığa** çok önemli olduğuna **inanılan eser** , dünyanın en eski yerleşim alanlarından biri olarak kabul ediliyor . [UNK] . peru 'nun tam orta **noktasına** yer alan 50 hektar alana sahip olan el paraiso **kentinin** tam orta **noktada** yer alan el paraiso **camisi 'ndeki festivalin** , ateş yakma **görevliye işaret olabileceği belirtiliyor** .
mBERT-uncased	peru 'da **500** yıl [UNK] ait [UNK] [UNK] bir tapınak [UNK] . **arkeologlar** , [UNK] en eski tarihlerinde bir uygarlık merkezi [UNK] belirtti . [UNK] konusu [UNK] , peru 'nun en eski [UNK] yerleri **arasında** yer **alıyor** . peru [UNK] [UNK] tarihin tam olarak [UNK] halinde , lima ve [UNK] [UNK] en [UNK] yerleri arasına [UNK] **bildirildi** . **arkeologlarınca** , " [UNK] [UNK] " adı verilen tapınakta bulunan fırının , [UNK] yakma inanılıyor . **arkeologlar**

Table 4.13 (cont.)

mBERT-cased	peru 'da , 5000 yıl öncesine ait olduğu düşünülen bir tapınak keşfedildi . kültür bakanlığı , söz konusu tapınağın dünyanın en eski yerleşim yerleri arasına girebileceğini açıkladı . peru 'da bulunan fırının , ateş yakma törenlerinin işaretçisi olduğuna inanılıyor . **pınak** 'nın bu amaçla kullanılmış olabileceği belirtiliyor . **pınaktaki** fırının , " ateş yakma **t**

Two documents from the test set were randomly selected and their summaries generated by each method were analyzed in terms of accuracy and intelligibility. The news article, the reference summary, and the generated summaries for these examples are shown in Tables 4.12 and 4.13. The text in bold indicate the novel words generated by the models that are not present in the content of the articles and the text in the tables which we refer to in the discussion below are shown underlined. In the first example, we see the effects of copying text from the source document for all the summaries produced by the pointer-generator models. The whitespace tokenization summary is longer than the others and has conveyed the main points of the article well. However, it also includes false information that does not exist in the article. The first line of the summary is read as "ingiltere 'nin başkenti londra'da yaşayan bir adam" (a man living in London, the capital of England), which does not match with the information in the content ("detroitli baba" (a father from Detroit)). Unigram LM, on the other hand, missed the main point of the article and produced an irrelevant summary. The summaries generated by the morphological tokenization methods, SeperateSuffix and CombinedSuffix, are very similar to each other. We observed that this is the case for most of the documents in the test set. Both summaries seem to have captured the salient information in the article. An interesting point is that the CombinedSuffix method, as the verb of the embedded clause in the sentence, produced a morphological variant ("isteyen" (one who wants)) rather than copying the original verb ("istiyordu" (he wanted)) from the source document. It is worth-noting that the

summaries generated by the two morphological tokenization approaches do not contain word formation errors or syntactical errors although they were formed of subwords instead of tokens. This shows that the SeperateSuffix and CombinedSuffix models are successful in language generation where the output text is formed of combinations of morphemes.

BERT-based models produced more novel summaries although they also tend to copy from the input document. For instance, both BERTurk models have paraphrased the text "detroitli baba" (a father from Detroit) as "abd 'li baba" (a father from the USA) which is closer to the text in the reference summary ("abd 'de bir baba" (a father in the USA)). Similarly, the mBERT-cased model has expanded the abbreviation "abd'de" (in the USA) as "amerika birleşik devletleri 'nde" (in the United States of America). Interestingly, mBERT-cased was also able to generate the morphological variant of "istiyordu" (he wanted) as "isteyen" (one who wants). However, it has output the word "kızının" (his daughter's) and corrupted the meaning of the sentence where it should have output a phrase similar to "babasının" (her father's). The summary of the 32K model is better in terms of intelligibility and has less syntactical errors than the larger 128k model. In addition, BERT-based models produced longer summaries compared to the pointer-generator models. Lastly, we observed that the mBERT-uncased model was not able to represent words that mostly contained Turkish specific characters. This causes its summary to be highly affected by unknown words.

In the second example (Table 4.13), the pointer-generator models produced similar summaries by heavily copying from the input document as in the previous example. All have copied the first sentence directly from the article and none of them was able to capture the core information "başkent lima 'nın kuzeyindeki ünlü el paraiso arkeolojik yerleşim alanında" (in the famous archaeological site Paraiso which is on the north of the capital Lima) as presented in the reference summary. In the SeperateSuffix summary, we see a paraphrase of the term "girebileceğini açıkladı" (he/she announced that it will be able to enter) as "girebilecek" (it will be able to enter), which is similar to the morphological variation in the previous example. Unlike the other methods, using

morpheme information has enabled the system to output new words.

BERT-based methods produced more novel summaries in line with the findings in Tables 4.11 and 4.7, and being longer as in the previous example, they could capture the salient information "el paraiso". However, they can also output unsupported information or corrupt the meaning in the content. For instance, BERTurk-uncased-32k attempted to output a time interval "[UNK] yılları arasında" (between the years [UNK]) that does not exist in the original text, and a subjectival noun phrase ""ateş tapınağı" adı verilen fırının" (of the oven named as "fire temple") that is a corrupted form of the original subject. The mBERT-cased model generated non-existent words "pınak", "pınaktaki", and "t" which are respectively suspected to be "tapınak" (temple), "tapınaktaki" (in the temple) and the first character of "töreni" (the ceremony). The mBERT-uncased model again suffers from words with Turkish specific characters.

For both examples, we see that BERTurk-uncased-32k and mBERT-cased have generated summaries that are more meaningful and syntactically correct when compared to other BERT-based summaries. This finding is also supported by the ROUGE results in Table 4.9. However, as reflected in Table 4.11, BERTurk-uncased-32k was able to generate more novel words when compared to mBERT-cased and this behavior might have caused the model to obtain a lower ROUGE score. In general, we observe that pointer-generator models produce less novel but more accurate summaries due to the copy mechanism, while the BERT-based models produce more abstractive summaries which may include unsupported information.

4.6. Discussion

In this chapter, we compiled two large-scale datasets aimed at text summarization for the agglutinative languages Turkish and Hungarian that suffer from resource scarcity. To the best of our knowledge, HU-News is the first large-scale text summarization dataset in Hungarian. The collected datasets also contain other valuable information that can be leveraged for various other tasks such as topic classification, au-

thor detection, key phrase extraction, and title generation. The effect of morphology on abstractive summarization was demonstrated through different tokenization methods including two linguistically-oriented approaches (SeperateSuffix and CombinedSuffix) proposed for both Turkish and Hungarian. The SeperateSuffix method achieved the highest ROUGE-1 score on the TR-News dataset amongst all the models. The BERT-based state-of-the-art models are also utilized to further expand the baselines provided in the work. The highest ROUGE-1 score on the HU-News dataset was obtained by the multilingual cased BERT model. Moreover, we showed that the multilingual cased BERT model outperforms the monolingual BERT models (BERTurk and huBERT) on both the TR-News and HU-News datasets in terms of ROUGE score.

In future works we plan to extend the morphological tokenization methods used in this study with other variations such as utilizing only the root and derivational suffixes or the root and the last suffixes of words. Since these variations may not be applied to text generation tasks, we plan to employ these tokenization methods in classification tasks such as POS tagging, NER, and a more semantic task like sentiment analysis. Additionally, we were not able to experiment with BERT-based models using morphological tokenization due to the lack of computational resources since such an experiment would require pretraining. Given adequate resources, a future direction can be pretraining BERT models with morphological tokenizers. Finally, by making TR-News and HU-News datasets publicly available, we hope that more research can be conducted in these low-resourced languages.

5. TURKISH ABSTRACTIVE TEXT SUMMARIZATION USING PRETRAINED SEQUENCE-TO-SEQUENCE MODELS

5.1. Introduction

Lately, with the advances in deep learning, neural abstractive text summarization with sequence-to-sequence (Seq2Seq) models has gained popularity. There have been many improvements in these models such as the use of pretrained language models (e.g., GPT, BERT, and XLM) and pretrained Seq2Seq models (e.g., BART and T5). These improvements have addressed certain shortcomings in neural summarization and have improved upon challenges such as saliency, fluency, and semantics which enable generating higher quality summaries.

Unfortunately, all these research attempts have been mostly limited to the English language. Additionally, pretraining such models requires vast amount of data and computational power which are factors that limit research. However, multilingual versions of the BERT [14] model and two multilingual pretrained Seq2Seq models (mT5 [59] and mBART [60]) have been released recently. This has given rise to many possibilities in various research areas for low-resourced languages. Moreover, many monolingual BERT models in various languages have been pretrained by the community including BERTurk [88], a monolingual Turkish BERT model.

Text summarization studies in Turkish are mostly based on extractive approaches. There are very few studies that try to tackle the abstractive summarization task in Turkish [26, 90]. None of these works has made use of pretrained Seq2Seq models which have shown to reach state-of-the-art results for English. Additionally, title generation is also considered as a text summarization task since the main objective is to output a condensed summary in the form of a title [5]. However, the number of title generation studies in Turkish is very limited [64]. There are currently two large-scale

datasets, TR-News [90] and MLSum [26], which are suitable for Turkish abstractive text summarization.

Following these insights, we address the following research questions in this chapter:

- RQ1: How do pretrained sequence-to-sequence models perform on Turkish abstractive text summarization and title generation tasks?
- RQ2: Does the monolingual BERT model obtain better results than the multilingual BERT model on the BERT2BERT model architecture?
- RQ3: Does combining datasets with similar characteristics improve model performance in abstractive text summarization and title generation?
- RQ4: How do models trained on one dataset perform across other datasets that have similar characteristics?
- RQ5: How much does the input to a title generation model impact the model performance?

Inline with the research questions, the contributions in this chapter are as follows:[22]

- We show that pretrained sequence-to-sequence models reach state-of-the-art on the TR-News and MLSum datasets for summary generation and title generation tasks.
- We conduct the first study that utilizes the titles of both datasets and we provide comprehensive and strong baselines for the title generation task.
- We show that monolingual BERTurk models outperform the multilingual BERT models on BERT2BERT architecture.
- We observe that combining both datasets yields better models for both text summarization and title generation tasks.

[22] All the available code has been made publicly available at https://github.com/batubayk/enc_dec_sum

- We conduct cross-dataset evaluations for both tasks and show that the models trained on TR-News are more robust compared to those trained on MLSum.
- We measure the efficacy of providing different inputs (LEAD sentences vs abstract) to a Seq2Seq model for title generation task and demonstrate that the abstract proves to be a better option compared to the LEAD sentences.
- We show how much preprocessing affects the ROUGE calculations, which is especially important for agglutinative languages like Turkish.

The rest of the chapter is organised as follows. In Section 5.2 we introduce the models used in this chapter. Later, in Section 5.3 the datasets used in this study are presented. Section 5.4 discusses the experimental setup, an analysis of the tokenization methods used in the models, and the novelty measurements for both text summarization and title generation tasks. The quantitative and qualitative results of the experiments are presented in Section 5.5. We conclude the chapter in Section 5.6.

5.2. Models

In recent years, pretraining a sequence-to-sequence model and finetuning it on downstream tasks such as machine translation and text summarization has shown to be very effective yielding state-of-the-art results in English. Until very recently, these models were mostly limited to the English language and it was not possible to assess the performance of such models in other languages. Pretraining these models require vast amount of data, computational resources, and budget, so obtaining these models for other languages is highly challenging. These limitations have been addressed in the recent multilingual pretrained sequence-to-sequence models mBART [60] and mT5 [59]. In this work, we utilize these two pretrained multilingual sequence-to-sequence models and also warm-start sequence-to-sequence (BERT2BERT) models from pretrained BERT models.

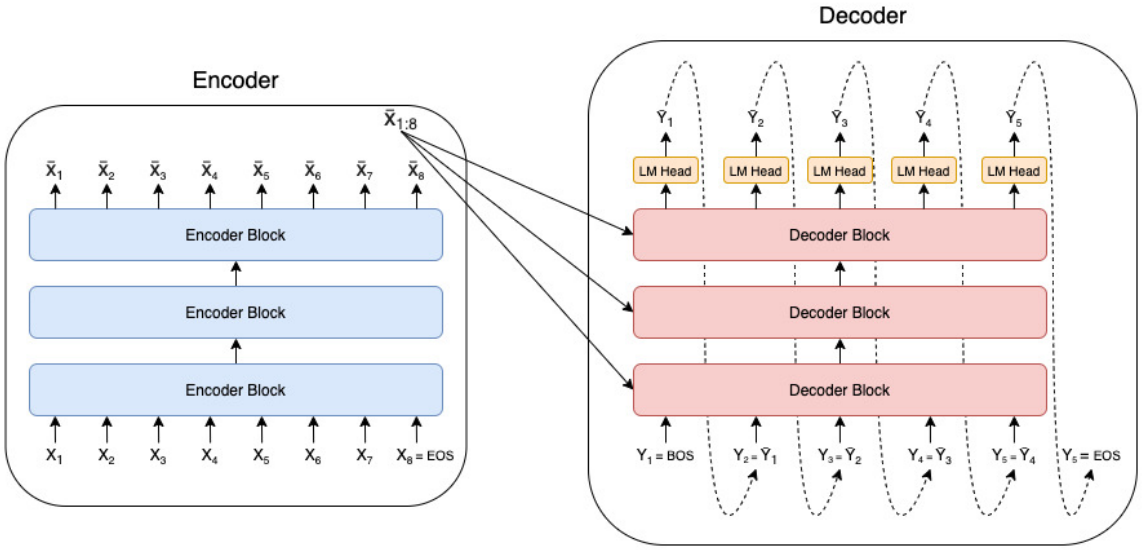

Figure 5.1. A high-level transformer-based encoder-decoder network.

5.2.1. BERT2BERT

BERT [14] is a bidirectional transformer network pretrained on a large corpus with two pretraining objectives; masked language modelling and next sentence prediction. It closely follows the original transformer network [9] with the major improvement being the bidirectional self attention mechanism. The authors have released several multilingual pretrained models that support a wide variety of languages including Turkish. In addition to the multilingual models, monolingual models have been pretrained by the community [88, 91–94]. Tokenization is an important aspect for these models since the input tokens are directly determined by the tokenization method and accordingly might impact the models' performance [7, 73]. Most of the released models follow the original BERT model and were pretrained using the WordPiece [71] tokenization method.

Unlike sequence-to-sequence models which are composed of two parts, an encoder and a decoder, BERT works as an encoder-only model. Figure 5.1 shows a high level view of a sequence-to-sequence transformer encoder-decoder model. The encoder transformer layers usually contain bidirectional connections which closely resemble the BERT model, whereas the decoder layers contain unidirectional (left to

right) connections. Although BERT is an encoder-only model, it is possible to utilize pretrained checkpoints so that a sequence-to-sequence model can be constructed by initializing both the encoder and the decoder parts with pretrained model checkpoints [18]. This procedure is known as warm-starting an encoder-decoder model. In order to achieve this objective with BERT, (1) a randomly initialized cross attention layer is added in between the self-attention layers and the feed-forward layers in the decoder layers, (2) BERT's bi-directional self-attention layers in the decoder are changed to uni-directional self-attention layers, and (3) a language model layer is added on top of the decoder component to define a conditional probability distribution while generating outputs. Consequently, the pretrained weights are directly transferred to the constructed encoder-decoder model with the only exception being the additional cross attention layers which are randomly initialized.

In this chapter, we use both the multilingual BERT model [14] and the monolingual Turkish BERT model called BERTurk [88], and their cased and uncased variations to warm-start pretrained sequence-to-sequence models. In the experiments part, the BERT2BERT models will be referred with their BERT model names (e.g. uncased multilingual BERT (mBERT-uncased) or uncased BERTurk (BERTurk-uncased)).

5.2.2. mBART

mBART (Multilingual Bidirectional and Auto-Regressive Transformers) [60] is the multilingual variation of the BART model [19]. BART is a pretrained encoder-decoder transformer network mostly suited to sequence-to-sequence tasks. The model is composed of a bidirectional encoder which closely resembles the BERT model [14] and an autoregressive decoder that takes its roots from the GPT (Generative Pretrained Transformer) model [57]. The BERT model is known to be more effective in language understanding tasks, whereas GPT-based models perform better in language generation tasks. Therefore, the BART model combines the strong aspects of both BERT and GPT-based models. Two different BART models have been released: base and large where the number of transformer layers for these models are 6 and 12, respectively.

On the other hand, only one model size for mBART has been released which has 12 transformer layers with a model dimension of 1024 on 16 heads.

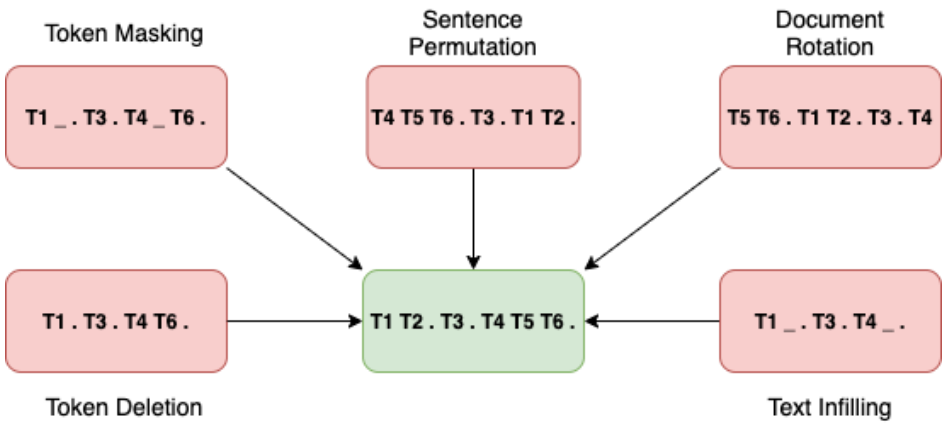

Figure 5.2. A number of noising methods experimented in the BART model. T1-T6 denote tokens. The box that the arrows point to shows the denoised text.

Similar to other pretrained models, BART makes use of several pretraining objectives and the main objective is to use denoising elements to corrupt the input and expect the model to reconstruct the original input. Hence, in principal it works as a denoising autoencoder. The noising methods on the input include token masking, token deletion, text infilling, sentence permutation, and document rotation which are displayed in Figure 5.2. The token masking operation randomly chooses tokens in the text and masks these tokens, whereas the token deletion operation deletes them. The text infilling method is similar to token masking but instead of choosing a single token, a span of tokens is chosen and masked where the span length is obtained from a Poisson distribution ($\lambda = 3$). The sentence permutation operation changes the order of the sentences and the document rotation operation shifts the entire text based on a randomly chosen token. The authors decided on a combination of text infilling and sentence permutation methods for the pretraining objective after completing extensive evaluations. The same approaches are also applied to the mBART model.

The BART model was pretrained on a combination of several resources such as books, news, web text, and stories following the work of Liu et al. [95], whereas a

subset of the Common Crawl (CC) corpus containing 25 languages [96] was used to pretrain the mBART model. Byte-pair encoding [69] method was used in the tokenization process of the BART model and SentencePiece [70] tokenization was utilized in the pretraining of the mBART model. Two mBART models have been released: mbart-large-cc25 and mbart-large-50 where the models have been trained on 25 and 50 languages, respectively. In this work we use the mbart-large-cc25 model and refer to it as the mBART model.

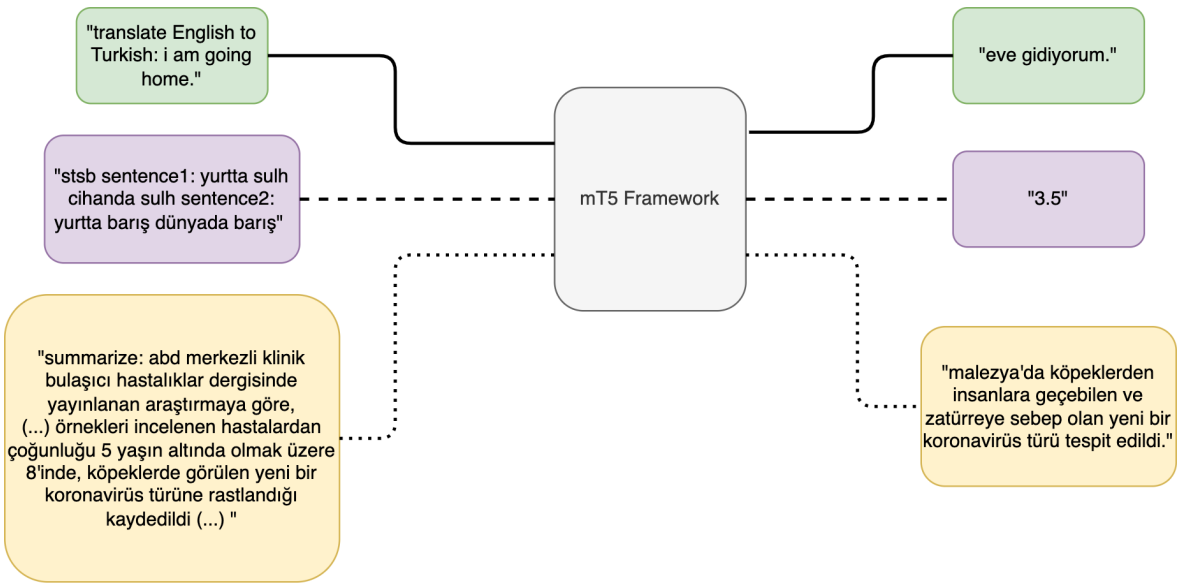

Figure 5.3. Various downstream tasks such as machine translation, semantic textual similarity, and text summarization on mT5 framework shown with examples in Turkish.

5.2.3. mT5

mT5 (Multilingual Text-to-Text Transfer Transformer) [59] is the multilingual variant of the T5 model [20] and does not incorporate any major changes in terms of the model architecture. The T5 model is a sequence-to-sequence encoder-decoder network which closely follows the originally proposed transformer architecture [9] with some minor modifications. The main idea behind the T5 model is to approach each text related task as a text-to-text problem where the system receives a text sequence

as an input and outputs another text sequence. This approach enables the system to use the same model and objective (teacher-forced maximum likelihood) for every downstream task. In that sense, T5 is an NLP framework capable of handling various tasks such as text summarization, question answering, text classification, and even tasks with continuous outputs such as semantic textual similarity under one unified framework. Figure 5.3 depicts the overall mT5/T5 models as a unified framework of various downstream tasks.

T5 makes use of several pretraining objectives to provide the model with generic capabilities which can be leveraged in downstream tasks. These include unsupervised objectives such as prefix language modeling, masked language modeling, and deshuffling along with several supervised objectives such as machine translation, text summarization, and text classification. As seen in Figure 5.3, each required task needs to be addressed with its corresponding prefix in the input sequence. For instance, the text summarization task requires the "summarize:" prefix, whereas the machine translation task requires the "translate English to Turkish:" prefix. The same approaches are also applied to mT5.

The pretraining of T5 was performed on the Colossal Clean Crawled Corpus (C4) [20] which is only suited to the English language, whereas another dataset called mC4 was derived from Common Crawl[23] for pretraining the mT5 model on 101 different languages [59]. The SentencePiece [70] algorithm is used in mT5 to cover a large multilingual vocabulary size of 250,000 which is several magnitudes higher compared to the original T5 vocabulary size of 32,000. The authors released several model sizes (small, base, large, xl, and xxl) for both T5 and mT5, and compared the model performances for the English language on the SQuAD reading comprehension benchmark [97] after finetuning the models to determine possible performance degradations. It was shown that mT5 falls behind the T5 model on all model sizes where the gap being smaller for larger models. Moreover, the mT5-xxl model reaches the state-of-the-art results in tasks such as paraphrase identification, natural language inference, and

[23]https://commoncrawl.org/

question answering on the XTREME multilingual benchmark [98] when compared to other multilingual pretrained models such as multilingual BERT [14] and XLM-R [99]. In this work, we use the mT5-base model due to computational restrictions that the larger models bring and refer to it as mT5 in our experiments.

5.3. Datasets

In this chapter, we make use of TR-News and MLSum [26] datasets. MLSum is intended as a multilingual text summarization dataset covering five languages: French, German, Spanish, Turkish, and Russian. However, we use the Turkish subset of MLSum and refer to it as MLSum (TR). The TR-News dataset was compiled from three different news websites: Cumhuriyet,[24] NTV,[25] and HaberTürk,[26] whereas the MLSum (TR) was obtained from a single website, İnternet Haber.[27] Both datasets cover news articles from a wide range of topics.

Table 5.1. Comparison of summarization datasets with respect to sizes of training, validation, and test sets, and average content, abstract, and title lengths (in terms of words and sentences)

Datasets	Num docs (train/val/test)	Content		Abstract		Title
		words	sentences	words	sentences	words
TR-News	277,573/14,610/15,379	286.18	15.72	25.05	1.48	6.53
MLSum (TR)	249,277/11,565/12,775	309.08	17.44	22.87	1.55	6.46
Combined-TR	526,850/26,175/28,154	296.97	16.53	24.02	1.51	6.50
CNN/Daily Mail	287,113/13,368/11,490	785.94	37.82	55.06	3.70	-
XSum	204,045/11,332/11,334	429.47	18.38	23.19	1.00	-

News-based datasets compiled for text summarization comprise of news articles and one or more reference summary for each article. The reference summary is normally constructed by human evaluators. However, for large scale datasets this is a very tedious work. Consequently, the reference summaries of these datasets are formed of the abstract part (highlight field) of the news articles [11, 26]. In this work, in addition

[24]https://cumhuriyet.com/
[25]https://www.ntv.com.tr/
[26]https://www.haberturk.com/
[27]https://www.internethaber.com/

to the news article and the abstract, we also leverage the titles in a separate title generation task which is considered as another type of summarization task [5, 21].

Table 5.1 shows the Turkish datasets used in this study. We also provide in the second part of the table two commonly used English summarization datasets, CNN/Daily Mail [11] and XSum [56], for comparison. As can be seen, TR-News and MLSum (TR) are similar in terms of the number of documents. Another important aspect for summarization tasks is the lengths of content, abstract, and title in number of words and sentences. These two datasets are similar to the XSum dataset with respect to the average number of sentences in the abstracts, which only contains one sentence per summary. This is partly due to the agglutinative nature of Turkish where the same information can be expressed with fewer words when compared to other languages such as English. Given the similar characteristics of TR-News and MLSum (TR), we combined these two datasets to see whether increasing the number of training samples would lead to a possible increase in model performances. We refer to the combined dataset as the Combined-TR.

Table 5.2. Comparison of summarization datasets with respect to vocabulary size and type-token ratio of content, abstract, title, and overall.

Datasets	Vocabulary size				Type-token ratio			
	content	abstract	title	overall	content	abstract	title	overall
TR-News	1,186,230	267,275	133,597	1,219,194	0.0135	0.0394	0.0665	0.0125
MLSum (TR)	1,109,917	228,511	109,628	1,143,534	0.0131	0.0365	0.0620	0.0123
Combined-TR	1,679,060	359,809	177,865	1,730,074	0.0097	0.0258	0.0471	0.0091
CNN/Daily Mail	869,792	240,663	-	893,985	0.0035	0.0140	-	0.0034
XSum	436,635	83,626	-	441,566	0.0045	0.0160	-	0.0043

The total number of distinct words (vocabulary size) and the type-token ratios for each dataset are given in Table 5.2. Type-token ratio (TTR) is calculated by dividing the vocabulary size to the total number of words. Agglutinative languages tend to have larger vocabulary sizes when compared to other languages due to the high number of suffixes the words can take. This can also be seen when the TTR values are compared; TR-News and MLSum (TR) have similar ratios, whereas CNN/Daily Mail and XSum have much lower ratios. Lastly, Combined-TR has a slightly lower TTR compared to

TR-News and MLSum (TR) since its vocabulary size is less than the sum of those of the two datasets. Importantly, higher vocabulary size of Turkish brings more complexity and causes NLP tasks to become more challenging when compared to English [39].

An example article from each dataset is given in Table 5.3. The table displays the fields URL, title, abstract, content, and date which are common to both datasets. However, TR-News also contains other valuable fields, which are topic, tags, author, and source but they are not relevant for this study and will not be used. The content field in the table has been cropped for convenience.

Table 5.3. Two news articles selected from TR-News and MLSum (TR)

	TR-News	MLSum (TR)
URL	https://www.haberturk.com/avrupa-birligi-abd-ve-cinli-teknoloji-devleriyle-mucadele-plani-hazirladi-2515715-teknoloji	https://www.internethaber.com/ise-surekli-gec-kalan-kadin-sorunun-kaynagini-bulunca-sok-oldu-2040194h.htm
Title	Avrupa Birliği ABD ve Çinli teknoloji devleriyle mücadele planı hazırladı	İşe sürekli geç kalan kadın, sorunun kaynağını bulunca şok oldu!
Abstract (Summary)	Avrupa Birliği Google, Microsoft, Apple gibi ABD'li ve Baidu, Alibaba gibi Çinli dev teknoloji şirketleriyle mücadele için bir plan hazırladı. Plan kapsamında kurulacak 100 milyar dolarlık fon Avrupalı teknoloji şirketlerine yatırım yapacak. Planda ayrıca Nokia'nın yıldızının parladığı yıllardaki stratejilerin uygulanması gerektiği belirtildi	Brezilya'da işe geç kalmaya başlayan bir kadın, alarmının her sabah kedisi tarafından kapatıldığını keşfetti.

Table 5.3 (cont.)

Content (Text)	Avrupa Birliği (AB) yetkililerinin, ABD Başkanı Donald Trump´ın ticaret savaşları politikası ve ABD merkezli teknoloji devleri Google, Apple, Amazon, Microsoft ve Facebook´a karşı alınacak önlemler hakkında 173 sayfalık bir plan hazırladığı bildirildi. Politico´nun haberine göre, plan öncelikle bir Avrupa Gelecek Fonu kurulmasını öngörüyor. Söz konusu fonun gelecek vadeden Avrupalı firmalara 100 milyar dolar yatırım yaparak ABD'li ve Çinli teknoloji şirketlerine karşı denge oluşturması hedefleniyor. (...)	Brezilya'da Sao Paolo'da yaşayan bir kadın, işe sürekli geç kalmaya başlayınca bu durumun nedenini araştırmaya başladı. Sabah saatlerine kurduğu alarmı duymamaktan şikayetçi olan kadın, yaptığı araştırma sonucunda işe geç kalma sebebinin kedisi olduğunu keşfetti. Kadın tarafından kaydedilen görüntülerde, telefon alarmı çalmaya başladıktan sonra kedisinin telefonun yanına gelerek alarmı patisiyle kapattığı görülüyor. (...)
Topic	Teknoloji	-
Tags	['microsoft', 'apple', 'google', 'baidu', 'haberler']	-
Date	23.08.2019 - 13:01	00/06/2019
Author	DHA	-
Source	haberturk	-

5.4. Experiments

In this section, we provide an analysis of the tokenization methods used in the models, briefly explain the main experiments, and present the results of the novelty analysis for the datasets used in this study. We focus on two different abstractive summarization tasks: text summarization and title generation. For both tasks we

make use of the state-of-the-art pretrained models and finetune them on the Turkish datasets.

5.4.1. Tokenization Analysis

Tokenization is one of the most important preprocessing steps in NLP problems. Tokenization approaches may vary depending on the problem. Simple methods such as whitespace tokenization can be applied if the vocabulary size is low, but in most cases the vocabulary size is immense. To solve the out-of-vocabulary problem, subword tokenization methods such as WordPiece [71], BPE [69], and SentencePiece [70] that can represent all the tokens with a vocabulary of a reasonable size have become popular in most sequence-to-sequence problems like machine translation and text summarization [9, 15]. The vocabulary and its size are critical because the input space and the output space of the pretrained models are directly determined by the tokenization method. This becomes even more important when the input is in a morphologically rich language such as Turkish or Czech and accordingly the input space has a much higher vocabulary size due to its nature.

In this work, we used BERT2BERT, mBART, and mT5 models where each model has been pretrained with a different tokenization method and has a different vocabulary. For BERT2BERT architecture two BERT-based models were used: the multilingual BERT (mBERT) and the monolingual BERTurk. Similar to the majority of published research in the summarization literature, the inputs to all the models are given in lowercase. While converting into lowercase, we took into consideration a special case in Turkish: the lowercases of characters "İ" and "I" are, respectively, "i" and "ı", unlike the "I"-"i" combination in English. Since the inputs to the models are in lowercase, we have decided to use the lowercase variations of the BERT-based models, but have recognized some encoding problems with the mBERT-uncased model. Interestingly, the encoding problem was not present in the mBERT-cased model. Therefore, we decided to additionally use the cased versions of both mBERT and BERTurk (although uncased version of BERTurk does not have any encoding problems) to further evaluate

Table 5.5. Tokenization outputs of the methods for a given Turkish sentence which translates to "If one day, my words are against science, choose science.".

Method	Output
	input: eğer bir gün benim sözlerim bilimle ters düşerse bilimi seçin.
mT5	['_', 'eğer', '_bir', '_gün', '_benim', '_söz', 'lerim', '_bilim', 'le', '_ter', 's', '_düş', 'erse', '_bilim', 'i', '_seçi', 'n', '.']
mBART	['_eğer', '_bir', '_gün', '_benim', '_sözleri', 'm', '_bilim', 'le', '_ter', 's', '_düş', 'er', 'se', '_bilim', 'i', '_seç', 'in', '.']
mBERT-uncased	['[UNK]', 'bir', '[UNK]', 'beni', '##m', '[UNK]', 'bilim', '##le', 'ter', '##s', '[UNK]', 'bilim', '##i', '[UNK]', '.']
mBERT-cased	['e', '##ğer', 'bir', 'gün', 'beni', '##m', 'söz', '##leri', '##m', 'bilim', '##le', 'ter', '##s', 'd', '##üş', '##erse', 'bilim', '##i', 'se', '##çi', '##n', '.']
BERTurk-uncased	['eğer', 'bir', 'gün', 'benim', 'sözleri', '##m', 'bilim', '##le', 'ters', 'düşer', '##se', 'bilimi', 'seçin', '.']
BERTurk-cased	['eğer', 'bir', 'gün', 'benim', 'sözleri', '##m', 'bilim', '##le', 'ters', 'düşer', '##se', 'bilimi', 'seçin', '.']

its impacts.

To show the notable differences between the tokenizers, an example input sentence and the tokenized outputs under each tokenization method are displayed in Table 5.5. As can be seen, all models' tokenizers behave uniquely and have their own format when splitting the words into subwords. The models mT5 and mBART use the SentencePiece method and place an underscore between the words and do not place any special characters between the subwords. BERT-based methods, on the other hand, make use of the WordPiece tokenization method and only place "##" between the subwords. The tokenizers are specific to the models and the outputs can differ based on the vocabulary size or the cased and uncased variation of the models. The outputs of the BERTurk models are more concise in terms of subwords, whereas the outputs of the multilingual BERT model tend to be longer and have been split from grammatically unrelated parts of the words. The tokens output for the mBERT-uncased model show the encoding problem discussed earlier where the words with some Turkish

specific characters (e.g. ğ, ü, ö, ş, ç) cannot be covered within the model's tokenizer properly. As a result, each model's output varies in terms of subwords and the number of subwords. This might reflect to the downstream tasks in terms of performance [100].

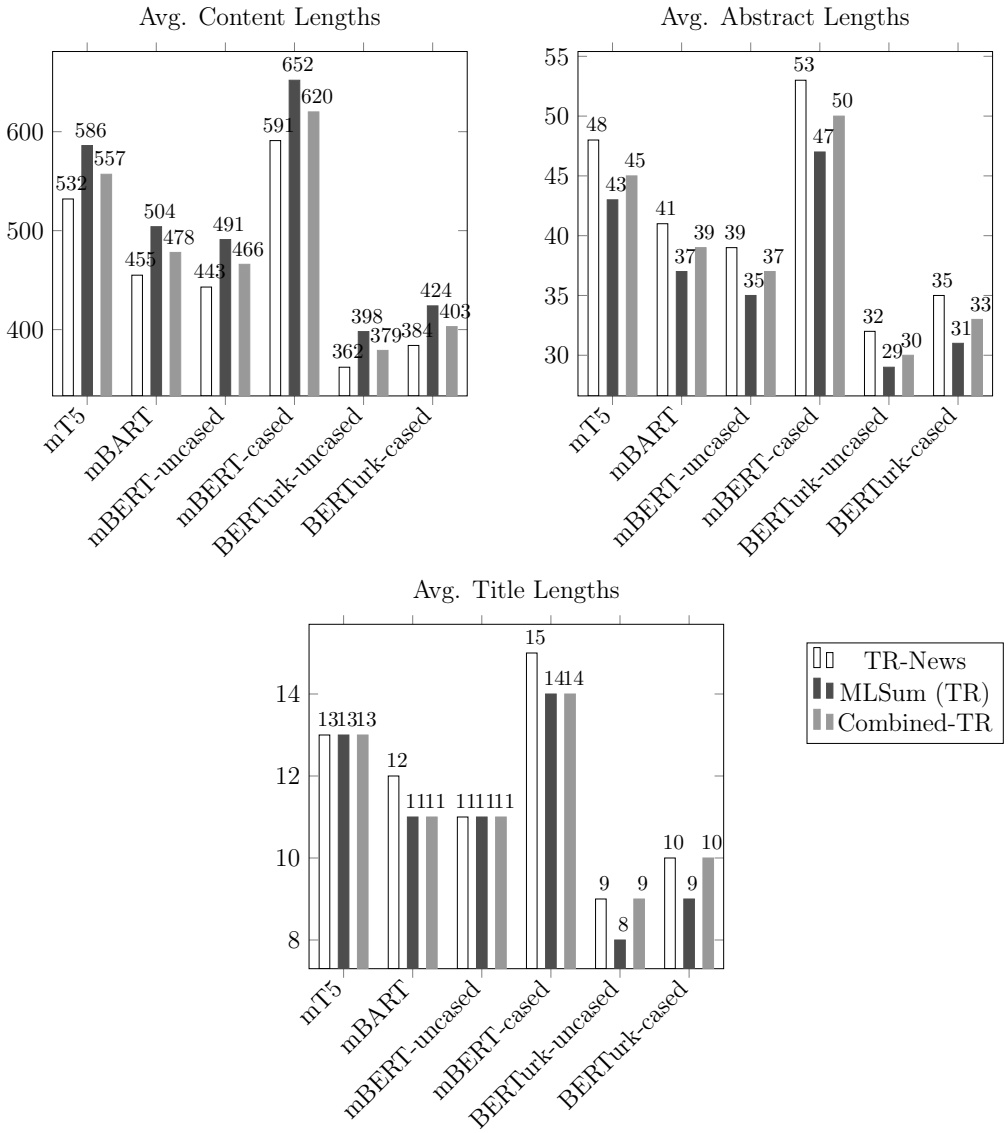

Figure 5.4. Average number of tokens generated by the tokenizers of the models for content, abstract, and title.

The average number of tokens generated by the models' tokenizers is given in Figure 5.4. We see that multilingual models generate more tokens compared to monolingual BERTurk models for all three fields as exemplified in Table 5.5. The mBERT-cased model is the one that generates the most number of tokens, whereas the BERTurk-

uncased model generates the least number of tokens. We believe the large difference between the cased and uncased versions of mBERT to be caused by the unknown tokens generated by the uncased variant. Additionally, a comparison between Table 5.1 and Figure 5.4 shows the gap between the number of words and subwords. All this information is important when constructing the models since the encoder and decoder lengths of the models need to be set based on these values. In this study, we consider the average and the maximum number of tokens generated by the tokenizers to determine an optimal size for the encoder and decoder lengths when finetuning the tasks.

5.4.2. Experiment 1 - Summary Generation

The first experiment aims to produce news article summaries in an abstractive manner using the pretrained encoder-decoder networks. For this purpose, we employ the mT5, mBART, and BERT2BERT models. For BERT-based models, both multilingual BERT models and monolingual BERTurk models are utilized to measure the effectiveness of monolingual pretrained models on the news article summarization task. As stated earlier, the uncased variant of the multilingual model cannot tokenize properly the Turkish specific characters. To assess the impact of this problem, we used both variants in the experiments. Similarly, both cased and uncased variants of BERTurk are utilized. In all the BERT2BERT models, we make use of the same BERT model in the encoder and the decoder parts. Lastly, it is known that using more data when training deep learning models usually tends to result in better performances [101]. We further investigate this notion with the same set of models on the Combined-TR dataset which we have created by merging the TR-News and MLSum (TR) datasets.

For the mT5 and mBART models we set the maximum encoder length to 768 and the maximum decoder length to 128 based on the observations given in Figure 5.4 to cover most of the contents and abstracts of the documents. For the BERT-based models, the maximum encoder length is limited to 512 due to model restrictions and the maximum decoder length is set to 128 as in the other models. The Adafactor optimizer

[102] is used for the mT5 as suggested by the authors. In our early experiments we also tried using the Adam optimizer [87] but we noticed that Adafactor converges much faster. The BERT-based models and the mBART model use the Adam optimizer. The learning rates for the mT5 and the other models are 1e-3 and 5e-5, respectively. During inference, we make use of tri-gram blocking to reduce the number of repetitions in the generated text.

Tesla V100 GPUs were used in the finetuning process of all the models with an effective batch size of 32. The models were finetuned for a maximum of 10 epochs; however, early-stopping with patience 2 was employed based on the validation loss. The number of warmup steps was set to 1000. Huggingface's transformers library was used for finetuning the models [103].

5.4.3. Experiment 2 - Title Generation

In the second experiment, we aim to generate news article titles in an abstractive manner using the same set of models and datasets as in the first experiment. Title generation task is also a summarization task in the sense that the model receives an input text that briefly describes the news article and a title that is suitable to the news is expected as an output. In this work, two types of inputs are used to generate the titles:

- **Abstract as input**: The reference summaries are considered to be concise representations of the news articles and are present for all the datasets used in this study. Therefore, we frame the title generation task by considering the abstract/reference summary as the input and the title as the output to the encoder-decoder model.
- **LEAD-3 as input**: In the literature, selecting the first three sentences of a news article (LEAD-3) is considered to be a strong baseline for the news article summarization task and is accordingly seen as a reference summary capable of reflecting the content of the news article [56]. Hence, we use the LEAD-3 as a

possible input to the title generation task as well.

For this experiment, the maximum encoder and decoder lengths have been set to 256 and 64, respectively. The remaining parameters and settings for all the models are the same as the first experiment.

Table 5.6. Novelty ratios of the datasets with respect to the summary generation and title generation tasks. N1, N2, and N3 denote uni-gram, bi-gram and tri-gram ratios, respectively.

Tasks	TR-News			MLSum (TR)			Combined-TR		
	N1	N2	N3	N1	N2	N3	N1	N2	N3
Summary	31.50	57.26	66.02	31.27	55.09	63.77	31.40	56.28	65.00
Title (Abstract)	52.61	65.11	59.56	48.22	66.21	67.88	50.62	65.61	63.33
Title (LEAD-3)	57.05	70.12	62.43	55.97	71.81	71.01	56.56	70.89	66.33

5.4.4. Novelty Analysis

In abstractive summarization, it is important to assess the degree of abstractiveness (text novelty) of the reference summaries in the datasets and of the generated summaries. High level of abstractiveness of the summaries in a dataset can be interpreted as being more challenging for the summarization task. In addition to being able to generate concise, relevant, and fluent summaries as in extractive models, abstractive models are also responsible for generating summaries that are genuine which do not contain a high amount of copied words from the source article. Novelty ratio is a commonly used metric which can provide insight to how abstractive a summary of a given article is. The novelty ratio is calculated as the percentage of the number of words in the summary that do not occur in the source document. To observe the abstractiveness of the datasets used in this study, we calculated the novelty ratios of the reference summaries and the titles. Table 5.6 shows the novelty ratios in terms of n-grams. For title generation, novelty ratios were calculated separately for the abstract and the LEAD-3 sentences as the source document. As can be seen, TR-News is slightly more abstractive than MLSum (TR) in terms of the summary generation task. For the title generation tasks, TR-News seems to have higher uni-gram ratios but

lower bi-gram and tri-gram ratios compared to MLSum (TR). The novelty analysis of the generated summaries and titles will be given in Section 5.5.1.3.

5.5. Results

In this section, we evaluate our findings both quantitatively and qualitatively for both the summary generation and the title generation tasks.

5.5.1. Quantitative Results

The models described in Section 5.2 were evaluated using the experimental settings discussed in the previous section with the ROUGE metric [35], a commonly used evaluation metric in text summarization. ROUGE-1, ROUGE-2, and ROUGE-L scores are reported. The ROUGE-n score measures the informativeness of the generated summaries by counting the number of common n-grams between the generated summary and the reference summary. ROUGE-L calculates the number of overlapping n-grams based on the longest common sub-sequences and measures the fluency of the generated summaries. In addition to the ROUGE metrics, the novelty ratios of the generated summaries and titles are also calculated in terms n-grams (n=1,2,3).

5.5.1.1. Experiment 1 - Summary Generation. Table 5.7 shows the results for the first experiment. In the first part of the table, the performance of the LEAD-2 and LEAD-3 baselines are given for all the datasets. LEAD baselines are commonly referred to in the evaluation of text summarization studies and are considered to be hard baselines to surpass [104, 105]. We also provide the results for the pointer-generator network [6] which are the state-of-the art results for both datasets. The second part of the table displays the results for the pretrained encoder-decoder models used in this study.

It is apparent that the mT5 and BERTurk-cased models perform very close to each other where the mT5 model being better on the individual datasets. Importantly, all the models except mBERT-uncased outperformed the pointer generator results with

Table 5.7. Text summarization results of pretrained encoder-decoder models on TR-News, MLSum (TR), and Combined-TR datasets. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) scores are given in F-measure. "-" denotes result is not available. Bold values show the highest scores obtained in the experiments per dataset.

Models	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
LEAD-2	31.37	17.91	26.92	36.32	23.18	31.39	33.61	20.31	28.95
LEAD-3	28.64	16.21	24.07	34.88	22.20	29.45	31.47	18.93	26.51
Pointer-generator See et al. [6]	31.61	18.55	29.57	38.04	25.01	35.70	35.23	22.03	33.04
Scialom et al. [26]	-	-	-	36.90	21.77	32.60	-	-	-
mT5	**41.13**	**25.75**	37.60	**42.26**	**27.81**	**37.96**	42.49	27.58	38.67
mBART	40.52	25.22	36.80	40.47	26.17	36.22	41.97	26.95	38.08
BERTurk-uncased	40.50	25.24	37.23	41.47	27.31	37.52	42.51	27.62	38.86
BERTurk-cased	41.06	25.60	**37.69**	41.48	27.23	37.66	**42.75**	**27.83**	**39.08**
mBERT-uncased	33.04	14.94	30.42	33.59	15.98	30.51	34.13	15.95	31.20
mBERT-cased	39.73	24.51	36.37	40.27	26.22	36.40	41.20	26.35	37.50

a large margin on both the TR-News and MLSum (TR) datasets. Hence, the results show that the pretrained encoder-decoder networks perform better than the RNN-based method (pointer-generator network) for the Turkish language (RQ1). Another finding for all the pretrained encoder-decoder models is the improvement obtained by joining the two datasets (Combined-TR). Increasing the number of training samples for the summary generation task seems to substantially increase the efficacy of all the models (RQ3). This supports the common knowledge of obtaining more training data would usually lead to performance gains in deep neural network based models. Additionally, the BERTurk-cased model slightly outperforms mT5 on the Combined-TR dataset. The multilingual BART model has performed worse than the BERTurk and mT5 models, but better than the multilingual BERT models for all the datasets.

When the BERT2BERT models are compared within themselves, it is evident that the cased models tend to perform better than the uncased models for both BERTurk and multilingual BERT. For multilingual BERT this is mostly due to the encoding problem. Moreover, the monolingual BERT models outperformed the multilingual BERT models, showing the effectiveness of pretraining on language specific data (RQ2).

Table 5.8. Title generation (abstract as input) results of pretrained encoder-decoder models on TR-News, MLSum (TR), and Combined-TR datasets. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) are given in F-measure. Bold values show the highest scores obtained in the experiments per dataset.

Models	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
mT5	**41.87**	**24.49**	40.87	**40.77**	**22.42**	**38.97**	43.04	**25.14**	41.59
mBART	37.72	20.99	36.74	34.85	18.03	33.46	39.94	22.44	38.46
BERTurk-uncased	40.93	23.67	40.05	38.04	20.16	36.37	42.48	24.51	41.07
BERTurk-cased	41.87	24.37	**40.88**	39.35	21.14	37.55	**43.06**	25.13	**41.61**
mBERT-uncased	33.88	15.39	33.20	31.18	12.68	30.04	34.48	15.46	33.50
mBERT-cased	40.83	23.50	39.89	38.98	21.07	37.30	42.14	24.32	40.70

Table 5.9. Title generation (LEAD-3 as input) results of pretrained encoder-decoder models on TR-News, MLSum (TR), and Combined-TR datasets. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) are given in F-measure. Bold values show the highest scores obtained in the experiments per dataset.

Models	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
mT5	**34.89**	**18.58**	34.01	**32.15**	**16.29**	**30.75**	35.53	**19.14**	34.36
mBART	31.81	15.96	31.03	27.06	13.06	25.92	23.27	11.00	22.44
BERTurk-uncased	33.80	17.58	33.06	30.31	15.05	29.11	34.72	18.42	33.66
BERTurk-cased	34.84	18.31	**34.08**	31.99	16.05	30.58	**35.66**	19.10	**34.52**
mBERT-uncased	27.26	11.29	26.72	24.73	9.49	23.88	28.05	11.64	27.27
mBERT-cased	33.28	17.17	32.44	30.79	15.47	29.52	34.35	18.19	33.26

5.5.1.2. Experiment 2 - Title Generation. The second experiment aims to measure the performance of the models on the title generation task. We use two different input types: abstract and LEAD sentences. Table 5.8 shows the results where abstract is used as the input and Table 5.9 shows the results where LEAD-3 is used as the input. The results are in parallel to the summary generation task in terms of model performances. When abstract is given as input to the models, mT5 and BERTurk-cased perform very close to each other in all the datasets, where the mT5 model performs slightly better on the MLSum (TR) dataset. The same is true for the LEAD-3 case in Table 5.9. In addition, combining the datasets has shown performance gains for all the models (RQ3). In terms of the BERT2BERT models, cased models have again shown to be better than their uncased variations. Moreover, monolingual BERT models outperformed their multilingual variants on the title generation task regardless of the input types (RQ2).

Interestingly, the mBART model has been unstable during training and this is reflected in the results. For instance, the model has shown an unexpectedly low performance on the Combined-TR dataset as seen in Table 5.9. If the mBERT-uncased model is set aside due to encoding problems, mBART can be considered as the model with the poorest performance amongst all the models used in this study for the title

Table 5.10. Title generation LEAD sentences ablation study results. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) are given in F-measure.

Models	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
LEAD-1	28.83	14.16	28.11	27.60	13.81	26.59	29.82	15.26	28.91
LEAD-2	33.20	17.31	32.39	30.78	15.64	29.51	33.84	17.87	32.75
LEAD-3	34.89	18.58	34.01	32.15	16.29	30.75	35.53	19.14	34.36
LEAD-4	35.41	18.95	34.54	33.06	17.06	31.56	36.06	19.48	34.86
LEAD-5	35.70	19.18	34.78	33.31	17.26	31.83	36.72	20.03	35.50

generation task. The mBART model seems to have performed worse compared to the summary generation task. This might indicate that mBART might be more suitable for tasks that require longer inputs and outputs.

Table 5.11. Novelty ratios of the summaries generated by the models per dataset. N1, N2, and N3 denote uni-gram, bi-gram, and tri-gram ratios, respectively. Bold values show the highest scores obtained in the experiments per dataset (the mBERT-uncased results are misleading and are ignored due to the high number of unknown tokens output).

Tasks	TR-News			MLSum (TR)			Combined-TR		
	N1	N2	N3	N1	N2	N3	N1	N2	N3
mT5	9.87	21.23	27.48	9.90	20..38	26.41	10.51	21.44	27.21
mBART	9.98	21.08	26.72	8.06	17.63	23.08	11.31	23.30	29.18
BERTurk-uncased	15.24	32.89	42.01	**16.23**	**32.63**	**41.27**	14.17	29.76	37.88
BERTurk-cased	**16.44**	**35.08**	**44.43**	15.93	32.21	40.72	**15.26**	**31.48**	**39.67**
mBERT-uncased	12.24	45.68	60.28	12.53	45.12	59.67	12.57	45.16	59.63
mBERT-cased	14.17	31.19	40.35	15.18	30.75	39.01	13.89	29.36	37.48

Another important finding for the title generation task is the impact of the input. When Table 5.8 and Table 5.9 are compared, providing the abstract as input to the title generation task seems to be more effective (RQ5). There can be several reasons behind this difference: (1) abstract is a more informative summary compared to LEAD-3, (2) abstract contains keywords more similar to the title, (3) abstract (around 1.5 sentences for both datasets - see Table 5.1) being shorter than LEAD-3 (3 sentences) holds more

relevant data for the title. To find out the impact of the input length related to the third claim, we conducted an ablation study where the first n LEAD sentences are given as input to the title generation model. For this and the other ablation studies in this work, the mT5 model is selected since it has shown to be one of the best models for both summary generation and title generation tasks. A total of five models were trained by feeding the first n sentences from the content as input expecting the title to be generated in the output. Table 5.10 shows the results for the ablation study. It can be seen that increasing the number of sentences in the input seems to increase the performance in the title generation task for all the datasets. This ablation study concludes that the length of the input is not a relevant factor that can explain the performance difference between providing the abstract versus LEAD-3 as the input.

Table 5.12. Novelty ratios of the titles (abstracts are given as input) generated by the models per dataset. N1, N2, and N3 denote uni-gram, bi-gram, and tri-gram ratios, respectively. Bold values show the highest scores obtained in the experiments per dataset.

Tasks	TR-News			MLSum (TR)			Combined-TR		
	N1	N2	N3	N1	N2	N3	N1	N2	N3
mT5	29.70	47.78	50.23	33.06	52.13	55.53	29.41	48.10	51.32
mBART	27.18	41.96	43.12	35.86	51.25	49.96	33.70	49.84	51.21
BERTurk-uncased	37.42	55.54	**55.45**	**47.00**	**64.29**	**64.55**	**35.63**	**54.71**	**56.38**
BERTurk-cased	**37.56**	**55.67**	55.43	44.81	62.16	62.60	34.43	53.32	55.49
mBERT-uncased	33.04	53.92	48.72	43.18	61.39	54.85	32.11	54.67	50.73
mBERT-cased	31.48	50.60	52.90	37.22	55.63	58.23	30.98	50.42	53.36

5.5.1.3. Novelty Analysis. As explained in Section 5.4.4, the novelty metric is used to assess the generated text in terms of abstractiveness. In this section, we evaluate the novelty degree of the generated summaries and titles in terms of n-grams (uni-gram, bi-gram, and tri-gram) on all the datasets. Table 5.11 and Table 5.12 show the novelty results for the summary generation and the title generation tasks, respectively. It is seen that the BERTurk models produce more novel outputs in both tasks, whereas the mT5 and the mBART models tend to produce less abstractive outputs compared to the other models for all the datasets. It is important to note that the results for

mBERT-uncased are misleading due to a high number of unknown ([UNK]) tokens generated in the outputs, which is caused by the character encoding problem of the model. Especially, the bi-gram and the tri-gram results of mBERT-uncased for the summarization task point out to irregular increases compared to its cased version. Hence, we choose to ignore the mBERT-uncased results for the novelty analysis. Lastly, the novelty ratios for the title generation task are much higher compared to summary generation. This shows that as the length of the outputs gets longer, the novelty ratio decreases.

5.5.1.4. Cross Dataset Evaluations. In Experiment 1 and Experiment 2, the models have been trained and evaluated on the same dataset. However, in real world applications of such models, one cannot make the assumption that the data will always come from the same distribution or source. Models in general tend to perform worse on sources which they were not trained on. In this experiment, we aimed to observe whether evaluating the trained models across different datasets would lead to a significant amount of performance degradation. Since the datasets we use have statistically similar attributes as described in Section 5.3, we conducted cross dataset evaluations on the two datasets and the combined dataset for both summary and title generation tasks. We made use of the mT5 model for all the evaluations.

Table 5.13. Cross-dataset evaluation results for the summary generation and the title generation (abstract as input) tasks. The values correspond to ROUGE-1 scores.

Model & Training Set	Summary			Title (Abstract)		
	TR-News	MLSum (TR)	Combined-TR	TR-News	MLSum (TR)	Combined-TR
mT5-TR-News	41.13	40.99	41.06	41.87	41.81	41.84
mT5-MLSum-TR	37.25	42.26	39.52	36.32	40.77	38.34
mT5-Combined-TR	41.23	44.01	42.49	42.46	43.79	43.04

Table 5.13 shows the results of cross dataset evaluations. (For more detailed results, please see Tables B.1 and B.2 in the Appendix B.) The rows correspond to the training set and the columns correspond to the test set. For the summary generation task, the best results were obtained with the mT5-Combined-TR model (the mT5 model trained on the Combined-TR dataset). The performance of this

model outperforms all the results obtained with the other two models (training sets) regardless of the test set used. This observation supports the findings of the previous two experiments related to RQ3.

When we consider training on individual datasets, we see that the models trained on TR-News and MLSum (TR) perform the best on their own test sets. However, the performance of mT5-TR-News is slightly affected when the test set changes, whereas the performance of mT5-MLSum-TR drops up to 5 ROUGE-1 points. The mT5-TR-News model also gives higher score than the mT5-MLSum-TR model on the combined test set. Lastly, the mT5-Combined-TR model performs better on the test set of MLSum (TR) rather than the combined test set. All these observations imply that the model trained on TR-News is a more robust model and performs well on data from other sources. This might indicate that TR-News is a more diverse dataset, providing richer information. On the other hand, the models trained on MLSum-TR and the combined training sets perform much better when the data come from the ML-Sum-TR source. This is probably a signal about the more specific nature of the MLSum (TR) dataset.

The results for the title generation task also support the cross-dataset findings of the summary generation task. In a similar manner, the mT5-Combined-TR model achieves the highest ROUGE-1 score across all the datasets. The model trained on MLSum (TR) struggles on the TR-News and Combined-TR datasets compared to the other models and also obtains the lowest score on its own test set.

5.5.1.5. Generation Parameters: Beam size and early-stopping. In the encoder-decoder models, during the inference phase the outputs are generated in an auto-regressive manner. Each token that is output is fed to the decoder as input in the next decoding step. Hence, each output token affects the tokens that will be generated in the future, which makes the decoding strategy an important variable that determines the quality of the generated text. In text summarization, the most commonly used decoding strategy at inference time is beam search. The aim of beam search is to keep track of the best

Table 5.14. Results for the summary generation and title generation (abstract as input) tasks with various beam sizes and early-stopping method. The values correspond to ROUGE-1 scores. Bold values show the highest scores obtained in the experiments per dataset.

Parameters	Summary			Title (Abstract)		
	TR-News	MLSum (TR)	Combined-TR	TR-News	MLSum (TR)	Combined-TR
mT5-beam-1	40.74	40.87	41.99	40.41	37.93	41.12
mT5-beam-2	**41.34**	42.13	**42.61**	41.58	39.95	42.49
mT5-beam-3	41.30	42.18	42.59	41.82	40.54	42.91
mT5-beam-4	41.13	**42.26**	42.49	**41.87**	**40.77**	**43.04**
mT5-beam-4 & early-stopping	41.15	41.36	42.18	41.66	40.04	42.53

n hypotheses at each step so that the sequence with the highest overall probability is not eliminated at an early stage due to a low probability token. The number n plays an important role in the performance of beam search. In this respect, we aimed to assess the effect of beam search with various beam sizes (1-4) where beam size 1 refers to greedy search. Moreover, we investigate the use of the early-stopping mechanism during the decoding phase, which allows the decoder to stop when all the hypotheses reach the special end of sentence token ([EOS]) instead of continuing until the predefined decoding length.

Table 5.14 shows the ROUGE-1 scores for the summaries and titles generated using the mT5 model on all the datasets. (For more detailed results, please see Tables B.3 and B.4 in the Appendix B.) We see that increasing the beam size mostly increases the performance. Although increasing the beam size past the size of 4 might continue increasing the scores, such an option brings more complexity and computational time. Also, we see that in some cases the ROUGE gains start to decrease after the beam size of 3. Based on these results, we consider the beam size of 4 as both yielding high ROUGE scores and allowing computationally tractable inference. Lastly, early-stopping is employed on the configuration with beam size of 4, but it reduced the performance in nearly all the evaluations.

Table 5.15. ROUGE scores calculated with different preprocessing settings. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations.

Parameters	Summary			Title (Abstract)		
	TR-News	MLSum (TR)	Combined-TR	TR-News	MLSum (TR)	Combined-TR
Punct removed Stems taken	41.13	42.26	42.49	41.87	40.77	43.04
Punct removed Stems not taken	37.60	39.03	39.12	37.91	37.22	39.24
Punct kept Stems taken	43.64	44.60	44.83	40.00	39.23	41.09
Punct kept Stems not taken	40.55	41.76	41.88	36.35	35.92	37.56

5.5.1.6. ROUGE Assessment Variations. ROUGE [35] is the most commonly used set of evaluation metrics in the literature for text summarization. The calculations are based on the overlapping tokens between the reference and the system summaries. Hence, the metric in its essence is based on exact match of the tokens. Therefore, any change to the tokens in the reference and the system summaries in terms of preprocessing operations before evaluating the ROUGE scores will affect the results. Removing the punctuations and applying stemming are commonly used as preprocessing operations in ROUGE evaluations. However, in most publications these details are not shared which makes interpreting the results difficult in some cases. Although stemming does not have a high impact on the results in English, it alters the surface form of an important number of words in agglutinative languages like Turkish, causing a significant change in the ROUGE scores.

Consequently, we aimed to show that such preferences can impact the results. We held a set of experiments which show the effect of punctuations and the stemming operation in ROUGE evaluations. During the experiments we realized that the original ROUGE script which is implemented in Perl and known as ROUGE 1.5.5[28] is

[28]https://github.com/andersjo/pyrouge/tree/master/tools/ROUGE-1.5.5

not capable of correctly processing non-English characters. Therefore, we made use of another repository which replicates the original Perl script in the Python programming language.[29] However, several modifications were needed in order to make it compatible with Turkish so that we made the necessary changes and also integrated Turkish stemming.[30]

Table 5.15 depicts the ROUGE-1 results obtained for both summary and title generation tasks. (For more detailed results, please see Tables B.5 and B.6 in the Appendix B.) As can be seen from the results, applying stemming highly increases the ROUGE-1 scores for both tasks on all the datasets. This is expected for the Turkish language since the amount of agglutination is very high. Keeping the punctuations seems to increase the score for the summary generation task as opposed to the title generation task. This implies that as the length of the evaluated texts gets longer, the amount of punctuations that get overlapped also increases, thus improving the ROUGE-1 score.

Furthermore, we held an additional set of experiments to observe whether the performance rankings of the models get affected by the preprocessing operation. Accordingly, the same experiment was conducted on all the models and datasets used in this study for the text summarization task. The results (Tables B.7, B.8, and B.9 in the Appendix B) are in parallel with the findings in Table 5.7 where mT5 and BERTurk models were again superior to the other models in most settings. However, for TR-News and Combined-TR we observe that the performance rankings in some cases change depending on the choice of the parameters. For instance, the setting where stemming is not applied and punctuations are not kept in Table B.7, BERTurk-cased slightly passes the mT5 model on the TR-News dataset. For the experiment in Table B.9, the best model becomes mT5 under the settings where punctuations are kept. These findings also support the claim that such preprocessing operations in ROUGE calculations can impact the results.

[29] https://github.com/google-research/google-research/tree/master/rouge
[30] https://github.com/otuncelli/turkish-stemmer-python

5.5.2. Qualitative Results

Apart from quantitative analysis, we provide a qualitative analysis for both the text summarization and the title generation tasks. Although quantitative analysis gives an idea about the informativeness and fluency of the models, other important aspects such as coherence and cohesion are left out. In this respect, we examined randomly chosen 50 examples from each dataset (100 examples in total) to observe on real data how well the generated summaries and titles fit to the reference summaries and titles. In this section, we provide two illustrative examples for each task from each dataset that are interesting and challenging.

5.5.2.1. Summary Generation. Tables 5.16 and 5.17 show an example from, respectively, TR-News and MLSum (TR) for the text summarization task. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models, whereas the underlined texts are for reference in the discussion below. For both examples, the content, the reference summary, and the generated summaries of the models are given. All the texts in the tables except the content fields have been translated to English. The content fields were left out due to limited space.

Table 5.16. An example from the test set of TR-News accompanied with the summaries generated by the models. The spelling and grammatical errors in the original texts are left as is. News article's content is given as the input and the reference summary is the abstract of the article. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models, whereas the underlined texts are for reference in the discussion.

Table 5.16 (cont.)

Content	fenerbahçe'nin braga'ya 4-1 yenilerek avrupa'ya veda ettiği maçı yöneten hakem ivan bebek'i hırvat basını da ağır dille eleştirdi. hırvat basınında yer alan bir analizde, ivan bebek'in sorgulanabilir kararlarla 3 fenerbahçeli oyuncuyu kırmızı kartla oyundan ihraç ettiğini, volkan şen'in ironik bir şekilde hakemi alkışladıktan sonra ittiğini belirtti. 67. dakikadaki penaltı kararının çok ağır ve haksız olduğu belirtilirken, bu dakikadan sonra ivan bebek'in fenerbahçe için kabus gibi bir yönettiğini, bu dakikadan sonra sarı lacivertlilerin kontrolü kaybettiğinin altı çizildi. ayrıca, yapılan analizde ivan bebek'in bu maçta verdiği kararların daha sonra da çok tartışılacağı kaydedildi. (sporx) (EN: the croatian press also heavily criticized referee ivan bebek, who directed the match in which fenerbahçe lost 4-1 to braga and bid farewell to europe. in an analysis in the croatian press, it was stated that ivan bebek dismissed 3 fenerbahçe players with a red card with questionable decisions, and volkan şen ironically pushed the referee after applauding. while stating that the penalty decision in the 67th minute was very harsh and unfair, it was underlined that after this minute, ivan bebek was leading like a nightmare fenerbahçe, and after this minute, the yellow blue colored team lost control. additionally, it was noted in the analysis that the decisions made by ivan bebek in this match will be discussed a lot later on. (sporx))

Table 5.16 (cont.)

Reference (Abstract)	hırvat hakem ivan bebek, fenerbahçe'nin braga'ya 4-1 yenilerek elendiği maçta verdiği ve es geçtiği kararlarla kendi ülkesinde de ağır eleştilere maruz kaldı. (EN: croatian referee ivan bebek was also subjected to heavy criticism in his own country for the decisions he made in the match where fenerbahçe was defeated 4-1 and eliminated by braga.)
mT5	**braga-fenerbahçe maçını** yöneten hakem ivan bebek'i hırvat basını da ağır dille eleştirdi. (EN: croatian press also heavily criticized the referee, ivan bebek, who directed the braga-fenerbahçe match.)
mBART	fenerbahçe'nin braga'ya 4-1 yenilerek avrupa'ya veda ettiği maçı yöneten hakem ivan bebek'i hırvat basını da ağır dille eleştirdi. (EN: croatian press also heavily criticized the referee ivan bebek, who directed the match in which fenerbahçe lost 4-1 to braga and bid farewell to europe.)
BERTurk-uncased	fenerbahçe'nin braga'ya 4 - 1 yenilerek avrupa'ya veda ettiği maçı yöneten hakem ivan bebek, hırvat basınında da ağır dille **eleştirildi**. (EN: referee ivan bebek, who directed the match where fenerbahçe lost 4 - 1 to braga and bid farewell to europe, was also heavily criticized in the croatian press.)
BERTurk-cased	**braga-fenerbahçe maçını** yöneten ivan bebek'i hırvat basını da ağır dille eleştirdi. (EN: croatian press also heavily criticized ivan bebek, who directed the braga - fenerbahce match)

Table 5.16 (cont.)

mBERT-uncased	braga'ya 4 - 1 yenilerek avrupa'ya veda hakem ivan bebek'i hırvat basını. (EN: beat braga 4 - 1, farewell to europe referee ivan bebek croatian press)
mBERT-cased	fenerbahçe'nin braga'ya 4 - 1 yenilerek avrupa'ya veda ettiği maçı yöneten hakem ivan bebek'i hırvat basını ağır dille eleştirdi. (EN: croatian press heavily criticized the referee ivan bebek, who directed the match in which fenerbahçe lost 4 - 1 to braga and bid farewell to europe.)

The first example in Table 5.16 is correctly summarized by all the models, except the mBERT-uncased model. Almost all summaries are very extractive; some of the summaries directly copy the first sentence and most of the models are not able to produce any novel unigrams. The most abstractive summaries belong to mT5 and BERTurk-cased, and they are very similar to each other. BERTurk-uncased changed the sentence from active voice to passive voice rather than directly copying, which made the summary more abstractive. The mBERT-cased model left the word "da" (also) when copying from the first sentence, but this slight change corrupted the meaning of the sentence. On the other hand, the mBERT-uncased model failed to output words with Turkish specific characters ("fenerbahçe'nin", "ettiği", etc.) which caused the summary to be incorrect both syntactically and semantically.

Table 5.17. An example from the test set of MLSum (TR) accompanied with the summaries generated by the models. News article's content is given as the input and the reference summary is the abstract of the article. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models, whereas the underlined texts are for reference in the discussion.

Table 5.17 (cont.)

Content	dün akşam saatlerinde etkili olan dolu ve sağanak nedeniyle merkeze bağlı konaklı, edikli beldeleri ile karaatlı ve çavdarlı köylerinde patates, fasulye ve mısır tarlalarını su bastı. tarım ve orman il müdürlüğü ekipleri bölgede yaptıkları incelemede yaklaşık 20 bin dekar alanın zarar gördüğünü tespit etti. tarım ve orman il müdürü asım baş, "niğde merkeze bağlı konaklı, edikli beldeleri ile karaatlı ve çavdarlı köylerinde yağmurun ardından dolu yağması bölgenin önemli geçim kaynaklarından başta patates, fasulye ve mısır olmak üzere bazı tarım ürünlerinin ekili olduğu arazileri olumsuz etkiledi. müdürlüğümüze bağlı ekiplerimizin şu anda tarım arazilerinde zarar tespit çalışmaları devam etmektedir" dedi. (EN: due to the hail and heavy rain that was effective yesterday evening, potato, bean and corn fields in the towns of konaklı, edikli and karaatlı and çavdarlı villages were flooded. the teams of the provincial directorate of agriculture and forestry determined that approximately 20 thousand decares of land had been damaged in their examination in the region. asım baş, the provincial director of agriculture and forestry, said: "the hail after the rain in the konaklı, edikli towns and karaatlı and çavdarlı villages of the center of niğde adversely affected the lands where some agricultural products are cultivated, especially potatoes, beans and corn, which are important sources of income in the region. at the moment, damage assessments of our teams are ongoing in agricultural lands.")
Reference (Abstract)	niğde'de dün etkili olan sağanak ve dolu nedeniyle 4 belde ve köyde yaklaşık 20 bin dekar tarım arazisi zarar gördü.

Table 5.17 (cont.)

	(EN: due to the heavy rain and hail that was effective yesterday in niğde, approximately 20 thousand decares of agricultural land in 4 towns and villages were damaged.)
mT5	**niğde'de** dün akşam saatlerinde etkili olan dolu ve sağanak nedeniyle patates, fasulye ve mısır tarlalarını su bastı. (EN: potato, bean, and corn fields were flooded in niğde due to the hail and heavy rain that were effective yesterday evening.)
mBART	**istan ilçesinde** etkili olan dolu ve sağanak nedeniyle 20 bin dekar alanın zarar gördüğü **bildirildi**. (EN: it has been reported that 20 thousand decares of land have been damaged due to the hail and heavy rain that were effective in the district of istan.)
BERTurk-uncased	**niğde'de** etkili olan dolu ve sağanak nedeniyle **çok sayıda ev** ve **iş yerini** su bastı. (EN: many homes and businesses were flooded due to the hail and heavy rain in niğde.)
BERTurk-cased	**niğde'de** dün akşam saatlerinde etkili olan dolu ve sağanak nedeniyle ekili tarım arazileri su bastı, 20 bin dekar alan zarar **gördü**. (EN: cultivated agricultural lands were flooded and 20 thousand decares of land were damaged in niğde due to the hail and heavy rain that were effective yesterday evening.)
mBERT-uncased	' da etkili olan dolu ve nedeniyle merkeze konaklı, edikli beldeleri ile karaatlı ve patates, fasulye ve mısır tarlalarını su bastı. (EN: to the center konaklı, edikli towns and karaatlı and potato, bean, and corn fields were flooded due to the hail and that was effective in.)

Table 5.17 (cont.)

mBERT-cased	**niğde'de** dün akşam saatlerinde etkili olan sağanak nedeniyle 20 bin dekar **alanı** su bastı. (EN: 20 thousand decares of land were flooded in niğde due to the heavy rain that was effective yesterday evening.)

Summaries generated for the second example are given in Table 5.17. None of the models included the number of towns and villages as in the reference summary or their names in the generated summaries. The mT5 model produced a correct but incomplete summary by not specifying the damaged decares of land. The mBART model generated a token referring to an unspecified location "istan" which does not exist in the news article. The BERTurk-uncased model has output unsupported information by emphasising that many homes and businesses were affected by the flood. All the models except mBART and mBERT-uncased managed to produce the word "Niğde'de" (in Niğde) which is an important novel word present in the reference summary. The best summary in terms of completeness is produced by the BERTurk-cased model although containing a small grammatical error "arazileri" (lands) (it should have been "arazilerini" by taking the accusative form of the word). Lastly, the mBERT-uncased model once again failed to output a grammatically and semantically correct summary.

5.5.2.2. Title Generation. Tables 5.18 and 5.19 show an example from, respectively, TR-News and MLSum (TR) for the title generation task. The words in bold denote novel words generated by the models, whereas the underlined texts are for reference in the discussion below. For both examples, the abstract, the reference title, and the generated titles of the models are given with their English translations.

The example in Table 5.18 is an interesting one which showcases some of the challenges in agglutinative languages. Although the titles generated by the models are syntactically correct, their meanings are different. This is caused by the novel words

Table 5.18. An example from the test set of TR-News accompanied with the titles generated by the models. News article's abstract is given as the input and the title of the article is expected as the output. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models, whereas the underlined texts are for reference in the discussion.

Abstract	çoklu organ yetmezliğinden önceki gün hayatını kaybeden istanbul âşığı tarihçi semavi eyice, hocaların hocası halil inalcık'ın fatih haziresi'ndeki kabrinin yanı başına defnedilecek (EN: historian semavi eyice, a lover of istanbul, has lost his life the earlier day due to multiple organ failure and will be buried next to the tomb of halil inalcık, the great teacher, in fatih Cemetery)
Reference (title)	tarihçi eyice, hocaların hocası inalcık'ın yanı başına defnedilecek (EN: historian eyice will be buried next to the tomb of inalcık, the great teacher)
mT5	istanbul âşığı eyice, halil inalcık'ın **kabrine** defnedilecek (EN: eyice, a lover of istanbul, will be buried to the tomb of halil inalcık)
mBART	semavi eyice **toprağa verilecek** (EN: semavi eyice will be buried)
BERTurk-uncased	semavi eyice, hocası halil inalcık ' ın kabrinin **yanında** defnedilecek (EN: semavi eyice will be buried at the side of his teacher halil inalcık's tomb)
BERTurk-cased	semavi eyice, hocası halil inalcık'ın **yanına** defnedilecek (EN: semavi eyice will be buried next to his teacher halil inalcık)
mBERT-uncased	halil inalcık'ın **hocasının yanındaki** kabrinin **yanına** defnedilecek (EN: will be buried next to the tomb that is next to the tomb of halil inalcık's teacher)
mBERT-cased	semavi eyice, halil inalcık ın yanı **başında** defnedilecek (EN: semavi eyice will be buried right beside halil inalcık)

that are introduced in the summaries. All the underlined texts in the table (except mBART) give the answer to the question *where* and result in different answers. Thus, slight changes to the morphemes of the words can alter the meaning of a whole sentence in Turkish. This is a factor that makes text generation more challenging compared to languages such as English. In the case of mBART, it has provided a much less informative title, however has managed to produce the novel phrase "toprağa verilecek" (will be buried) which has the same meaning as the word "defnedilecek" in the abstract.

Table 5.19. An example from the test set of MLSum (TR) accompanied with the titles generated by the models. News article's abstract is given as the input and the title of the article is expected as the output. The words in bold denote novel unigrams (unigrams which are not present in the input text) generated by the models.

Abstract	ingiltere 'de resmen ülkenin yeni başbakanı olan boris johnson, 31 ekim'de brexit'i gerçekleştireceklerini, ve ab'den ayrılmaya hazır olduklarını açıkladı. (EN: boris hohnson, who is officially the new prime minister of the country in england, announced that they will hold brexit on october 31st and that they are ready to leave the eu.)
Reference (Title)	boris johnson'dan brexit mesajı! ab'den ayrılmaya hazırız (EN: brexit message from boris johnson! we are ready to leave the eu)
mT5	boris johnson ab'den ayrılmaya hazır (EN: boris johnson is ready to leave the eu)
mBART	ingiltere ab'den ayrılmaya hazır (EN: england is ready to leave the eu)
BERTurk-uncased	brexit : ab'den ayrılmaya **hazırız** (EN: brexit : we are ready to leave the eu)
BERTurk-cased	ingiltere ab'den ayrılmaya hazır (EN: england is ready to leave the eu)
mBERT-uncased	ingiltere **brexit'i hazırız** (EN: england ready brexit)
mBERT-cased	ingiltere başbakanı **brexit'ten ayrılıyor** (EN: prime minister of england is leaving brexit)

Table 5.19 shows another example for the title generation task. As in the previous

example, all models except mBERT-uncased managed to produce syntactically correct titles. BERTurk-uncased, on the other hand, produced a semantically incorrect title by mistaking *Brexit* with *England*. The most abstractive output was generated by the mBERT-cased model producing novel unigrams and also generating the bigram "ingiltere başbakanı" (prime minister of England) which is not present in the abstract. However, it also failed to produce a meaningful title mistaking *Brexit* with the *EU*. Accordingly, the best titles that reflect the reference for this example belong to the mT5, mBART, and BERTurk-cased models.

5.6. Discussion

In this chapter, we analyzed in detail the performance of pretrained sequence-to-sequence models on two tasks, text summarization and title generation. The mT5 model reached the state-of-the-art results on both the TR-News and MLSum (TR) datasets in terms of the ROUGE scores for both tasks. The monolingual BERTurk-cased model also showed a performance close to the mT5 model and produced more novel summaries. We established strong baselines for both datasets for the summary generation task and also the title generation task for the Turkish language. Further analysis on the title generation task revealed that the input to the model impacts the task's outcome greatly. Providing the abstract of the news articles as input to the models showed better ROUGE scores compared to giving the LEAD sentences as input. Moreover, we created a larger dataset (Combined-TR) by combining both TR-News and MLSum (TR) since both have similar characteristics in terms of statistics and content. The models trained on Combined-TR showed performance gains for both the text summarization and title generation tasks. Lastly, the monolingual BERT models outperformed the multilingual BERT models in the BERT2BERT model architecture on both tasks.

In future works, we plan to extend this study with summarization datasets from different languages, specifically agglutinative languages. Given adequate computational resources, pretraining monolingual Seq2Seq models for low resourced languages from

scratch and comparing the results with the multilingual pretrained Seq2Seq models can be a future possibility. Moreover, the pretraining objectives can be altered to take into consideration the agglutinative nature of such languages.

6. MORPHOSYNTACTIC EVALUATION FOR MORPHOLOGICALLY RICH LANGUAGES: A CASE STUDY FOR TURKISH

6.1. Introduction

Evaluation of summarization methods is critical to assess and benchmark their performance. The main objective of evaluation is to observe how well the output summary is able to reflect the reference summaries. The commonly used evaluation methods in summarization such as ROUGE [35] and METEOR [36] are based on n-gram matching strategy. For instance, ROUGE computes the number of overlapping word n-grams between the reference and system summaries in their exact (surface) forms. While the exact matching strategy is not an issue for extractive summarization where the words are directly copied, it poses a problem for abstractive summarization where the generated summaries can contain words in different forms. In the abstractive case, this strategy is very strict especially for morphologically rich languages in which the words are subject to extensive affixation and thus carry syntactic features. It severely punishes the words that have even a slight change in their forms. Hence, taking the morphosyntactic structure of these morphologically rich languages into account is important for the evaluation of text summarization.

In this chapter, we introduce several variants of the commonly used evaluation metrics that take into account the morphosyntactic properties of the language. As a case study for Turkish, we reuse the models mT5 [59] and BERTurk-cased [88] which were trained in Chapter 5. The summaries generated by the models are evaluated with the proposed metrics using the reference summaries. In order to make comparisons between the evaluation metrics, we perform correlation analysis to see how well the score obtained with each metric correlates with the human score for each system summary-reference summary pair. It is challenging to find manually annotated data in text summarization since Turkish is a resource-scarce language. Hence, for correlation

analysis, we annotate human relevancy judgements for a randomly sampled subset of the TR-News dataset and we make this data publicly available[1]. Correlation analysis is performed using the annotated human judgements to compare the performance of the proposed morphosyntactic evaluation methods as well as other popular evaluation methods.

The rest of the chapter is organized as follows. In Section 6.2, we explain the proposed morphosyntactic evaluation methods in detail. Section 6.3 describes the dataset used in this work, the annotation process, and the models used. The results are presented and discussed in Section 6.4. Section 6.5 concludes the chapter.

6.2. Methodology

In this section, we explain the proposed methods that are based on the morphosyntactic features of Turkish. This is followed by the explanation of the evaluation metrics used to compare the system and reference summaries tokenized with the proposed methods.

6.2.1. Morphosyntactic Variations

While comparing a system summary and a reference summary, the evaluation metrics used in text summarization use either the surface forms or the lemma or stem forms of the words. As stated in Section 6.1, the former approach is too restrictive and misses matches of the inflected forms of the same words, whereas the latter approach is too flexible and allows matches of all derivations of the same root which causes semantically distant words to match. In this work, we propose and analyze several other alternatives in between these two extreme cases based on morphosyntactic properties of the language. In each proposed method, all the words in the system and reference summaries are first processed as stated in the method and then one of the evaluation metrics (ROUGE, METEOR, etc.) is applied in the usual way.

Table 6.1. Morphological analysis of an example sentence.

Input	Morphological Analysis
tutsağı	[tutsak:Noun] tutsağ:Noun+A3sg+ı:Acc
serbest	[serbest:Adj] serbest:Adj
bıraktılar	[bırakmak:Verb] bırak:Verb+tı:Past+lar:A3pl

Table 6.1 shows the disambiguated morphological analysis of the sentence *tutsağı serbest bıraktılar* (*they released the prisoner*) as an example. The square bracket shows the root and its part-of-speech, which is followed by the suffixes attached to the root and the morphological features employed during the derivation[31] . The noun *tutsak* (*prisoner*) takes the accusative suffix 'ı' and is transformed into *tutsağı*[32] (*the prisoner*). The second word *serbest* (*free*) is analysed as an adjective and does not take any suffixes. The last word *bırak* (*to release*) is the verb and takes two suffixes which are past tense and third person plural.

Table 6.2 gives the list of the methods used to process the words before applying the evaluation metrics and shows the result of each one for the example sentence depicted in Table 6.1. The Surface method leaves the words in their written forms, while the Lemma (Stem) method strips off the suffixes and takes the lemma (stem) forms of the words. The lemma and stem forms are obtained using the Zemberek library [79] which applies morphological analysis and disambiguation processes. For the Lemma and Stem methods, in addition to their bare forms, six different variations based on different usages of the suffixes are employed. The suffixes used in these variations are also obtained from the morphological parse by the Zemberek library. Only the variations of the Lemma method are shown in the table to save space; the same forms are also applied to the Stem method. The methods are explained below.

Surface: The text is only lower-cased and punctuations are removed. All the other methods also perform the same cleaning and lower-casing operations. For Turk-

[31]The morphological features used in the example are as follows: Acc=accusative, A3pl=third person plural number/person agreement, A3sg=third person singular number/person agreement, Past=past tense.

[32]The voiceless stop consonant 'k' is voiced as 'ğ' when a suffix starting with a vowel is attached.

Table 6.2. Proposed methods based on morphosyntactic variations of words.

Method	Processed Text
Surface	tutsağı serbest bıraktılar
Lemma	tutsak serbest bırak
Stem	tutsağ serbest bırak
Lemma and all suffixes	tutsak ##ı serbest bırak ##tı ##lar
Lemma and combined suffixes	tutsak ##ı serbest bırak ##tılar
Lemma and last suffix	tutsak ##ı serbest bırak ##lar
Lemma and all suffixes with Surface	tutsağı##tutsak tutsağı##ı serbest##serbest bıraktılar##bırak bıraktılar##tı bıraktılar##lar
Lemma and combined suffixes with Surface	tutsağı##tutsak tutsağı##ı serbest##serbest bıraktılar##bırak bıraktılar##tılar
Lemma and last suffix with Surface	tutsağı##tutsak tutsağı##ı serbest##serbest bıraktılar##bırak bıraktılar##lar

ish, this is the default evaluation strategy for all the metrics.

Lemma: The text is lemmatized and the lemma forms of the words are used.

Stem: The text is stemmed and the stem forms of the words are used.

Lemma and all suffixes: The text is lemmatized and the suffixes are extracted. The lemma and each suffix of a word are considered as separate tokens.

Lemma and combined suffixes: The text is lemmatized and the suffixes are extracted. The suffixes are concatenated as a single item. The lemma and the concatenated suffixes of a word are considered as separate tokens.

Lemma and last suffix: The text is lemmatized and the suffixes are extracted. The lemma and the last suffix of a word are considered as separate tokens.

The last three methods above split the lemma and the suffixes and use them as individual tokens. This may cause the same tokens obtained from different words to match mistakenly. For instance, if the system summary contains the word *tutsağı* (*the prisoner*) (the accusative form of *tutsak* (*prisoner*)) and the reference summary contains the word *gardiyanı* (*the guardian*) (the accusative form of *gardiyan* (*guardian*)), the morphological parse will output the suffix 'ı' for both of them. The evaluation metric (e.g. ROUGE-1) will match these two suffixes (tokens) although they belong to different words. To prevent such cases, we devise another variation of these three methods where the surface form of the word is prefixed to each token generated from the word as explained below.

Lemma and all suffixes with Surface: The text is lemmatized and the suffixes are extracted. The surface form of a word is added as a prefix to the lemma and each of the suffixes of the word. The lemma and each suffix of the word are then considered as separate tokens.

Lemma and combined suffixes with Surface: The text is lemmatized and the suffixes are extracted. The suffixes are concatenated as a single item. The surface form of a word is added as a prefix to the lemma and the concatenated suffixes of the word. The lemma and the concatenated suffixes of the word are then considered as separate tokens.

Lemma and last suffix with Surface: The text is lemmatized and the suffixes are extracted. The surface form of a word is added as a prefix to the lemma and the last suffix of the word. The lemma and the last suffix of the word are then considered as separate tokens.

6.2.2. Evaluation Metrics

We use four different metrics for comparing system summaries and reference summaries. We apply the morphosyntactic variations to the summaries and then score the performance using these metrics. In this way, we make a detailed analysis related to which combinations of evaluation metrics and morphosyntactic tokenizations correlate well with human judgments. We explain below each metric briefly.

ROUGE [35] is a recall-oriented metric which is commonly used in text summarization evaluation. ROUGE-N computes the number of overlapping n-grams between the system and reference summaries while ROUGE-L considers the longest common sub-sequence matches. There are also other forms of the ROUGE metric such as ROUGE-W which is the weighted version of ROUGE-L and ROUGE-S and ROUGE-SU which compute the skip-gram matches.

METEOR [36] is another commonly used metric in text summarization [26,106]. It is based on unigram matches and makes use of both unigram precision and unigram recall. Word order is also taken into account via the concept of chunk. In addition to exact matches, a back-off strategy is employed where stems and synonyms are also matched. However, this requires dependencies to external libraries.

BLEU [50] is a precision-oriented metric originally proposed for machine translation evaluation. It uses a modified version of n-gram precision. If a word in the reference summary occurs several times in the system summary, it is counted as one match rather than several matches. The metric takes into account both the common words in the summaries by the use of unigrams and the word order in the summaries by the use of higher order n-grams. The common words aspect measures the adequacy of summaries while the word order feature captures the fluency of summaries. In addition, a penalty factor called brevity penalty is applied when the system summary is shorter than the reference summary in order to prevent erroneously high precision values in such cases. Although not common as ROUGE, BLEU is also used in text summarization evaluation as an additional metric [107, 108].

BERTScore [68] is a recent metric proposed to measure the performance of text generation systems. It focuses on semantic similarity rather than syntactic similarity as in the case of n-gram-based metrics. It first extracts contextual embeddings of the words in the system and reference summaries using the BERT model and then computes pairwise cosine similarity between the words of the summaries. Optionally, inverse document frequency (IDF) values of the words extracted from a large corpus can be used to weight the importance of the words.

In this work, we make use of the Huggingface's `evaluate` library[33] for all the metrics explained above. We use the monolingual BERTurk-cased [88] model for computing the BERTScore values.

6.3. Dataset, Models, and Annotations

In this section we first explain the dataset and the models used for the text summarization experiments. We then give the details of the annotation process where the summaries output by the models are manually scored with respect to the reference summaries. The human judgment scores will be used in Section 5 to observe the

[33] https://github.com/huggingface/evaluate

goodness of the proposed morphosyntactic methods.

6.3.1. Dataset and Models

We use the TR-News dataset introduced in Chapter 4 for the experiments. For the summarization models, we use the best mT5 and BERTurk-cased models that were trained in Chapter 5.

Table 6.3. Average scores and inter-annotator agreement scores for the models. In the first row, the averages of the two annotators are separated by the / sign.

	BERTurk-cased	mT5
Avg. annotator score	5.86 / 6.22	6.00 / 5.88
Pearson correlation	0.85	0.88
Cohen's Kappa coefficient	0.44	0.25

6.3.2. Human Judgment Annotations

In order to observe which morphosyntactic tokenizations and automatic summarization metrics perform well in evaluating the performance of text summarization systems for morphologically rich languages, we need a sample dataset consisting of documents, system summaries, reference summaries, and relevancy scores between the system and reference summaries. For this purpose, we randomly sampled 50 articles from the test set of the TR-news dataset. For each article, the system summary output by the model is given a manual score indicating its relevancy with the corresponding reference summary. This is done for the mT5 model and the BERTurk-cased model separately. The relevancy scores are annotated by two native Turkish speakers. An annotator is shown the system summary and the reference summary for an article without showing the original document and is requested to give a score. We decided to keep the annotation process simple by giving a single score to each system summary-reference summary pair covering the overall semantic relevancy of the summaries instead of scoring different aspects (adequacy, fluency, style, etc.) separately. The scores range from 1 (completely irrelevant) to 10 (completely relevant).

Table 6.3 shows the average scores of the annotators and the inter-annotator agreement scores. The averages of the two annotators are close to each other for both models. The Pearson correlation values being around 0.80-0.90 indicate that there is a strong agreement in the annotators' scores. We also present the Cohen's Kappa coefficient as a measure of agreement between the annotators. The values of 0.44 and 0.25 signal, respectively, moderate agreement and fair agreement between the scores [109]. Since the Cohen's Kappa coefficient is mostly suitable for measuring agreement in categorical values rather than quantitative values as in our case, the results should be approached with caution.

6.4. Correlation Analysis

In this work, we mainly aim at observing the correlation between the human evaluations and the automatic evaluations for the system generated summaries. For each of the proposed morphosyntactic tokenization methods (Section 6.2.1), we first apply the method to the system and reference summaries of a document and obtain the tokenized forms of the words in the summaries. We then evaluate the similarity of the tokenized system and reference summaries with each of the standard metrics (Section 6.2.2). Finally, we compute the Pearson correlation between the human score (average of the two annotators) given to the reference summary-system summary pair (Section 6.3.2) and the metric score calculated based on that morphosyntactic tokenization.

In this way, we make a detailed analysis of the morphosyntactic tokenization method and text summarization metric combinations. The results are shown in Tables 6.4 and 6.5. For the ROUGE metric, we include the results for the ROUGE-1, ROUGE-2, and ROUGE-L variants that are commonly used in the literature. For the tokenization methods that include suffixes, we show only the results with the surface forms of the words prefixed to the tokens (*with Surface*). The results without the prefixed tokens are given in the Appendix C. Interestingly, the methods that do not use the prefix forms correlate better with the human judgments, although they tend to incorrect matches as shown in Section 6.2.1.

Table 6.4. Pearson correlation results of the morphosyntactic methods with prefix tokens for the BERTurk-cased summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column.

BERTurk-cased						
	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	BERTScore
Surface	0.770	0.723	0.750	0.736	0.649	**0.800**
Lemma with Surface	**0.802**	**0.730**	**0.768**	**0.807**	**0.776**	0.766
Stem with Surface	<u>0.792</u>	<u>0.728</u>	<u>0.759</u>	<u>0.802</u>	<u>0.773</u>	0.763
Lemma and all suffixes with Surface	0.773	0.712	0.743	0.796	0.765	0.760
Stem and all suffixes with Surface	0.768	0.712	0.740	0.794	0.764	0.760
Lemma and combined suffixes with Surface	0.774	0.718	0.747	0.796	0.771	<u>0.768</u>
Stem and combined suffixes with Surface	0.767	0.718	0.741	0.794	0.770	0.767
Lemma and last suffix with Surface	0.781	0.718	0.749	0.798	**0.776**	0766
Stem and last suffix with Surface	0.774	0.718	0.743	0.798	**0.776**	0.766

Table 6.5. Pearson correlation results of the morphosyntactic methods with prefix tokens for the mT5 summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column.

mT5						
	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	BERTScore
Surface	0.682	0.648	0.693	0.697	0.591	<u>0.693</u>
Lemma with Surface	**0.701**	**0.669**	**0.709**	0.753	0.719	0.682
Stem with Surface	0.688	<u>0.665</u>	<u>0.700</u>	0.742	0.714	0.674
Lemma and all suffixes with Surface	<u>0.699</u>	0.658	<u>0.700</u>	**0.771**	**0.730**	**0.694**
Stem and all suffixes with Surface	0.693	0.658	0.698	<u>0.767</u>	<u>0.728</u>	0.690
Lemma and combined suffixes with Surface	0.685	0.653	0.693	0.750	0.714	0.690
Stem and combined suffixes with Surface	0.677	0.653	0.688	0.745	0.712	0.687
Lemma and last suffix with Surface	0.692	0.653	0.699	0.749	0.712	0.674
Stem and last suffix with Surface	0.684	0.653	0.693	0.743	0.710	0.671

We observe that the Lemma method mostly yields the best results for the summaries generated by the BERTurk-cased model. The Lemma method is followed by the Stem method. These results indicate that simply taking the root of the words in the form of lemma or stem before applying the evaluation metrics is sufficient instead of more complex tokenizations. One exception is the BERTScore metric which works best with the surface forms of the words. This may be regarded as an expected behavior since BERTScore is a semantically-oriented evaluation approach while the others are mostly syntactically-oriented metrics. Hence, when fed with the surface forms, BERTScore can capture the similarities between different orthographical forms of the words.

The summaries generated by the mT5 model follow a similar pattern in ROUGE evaluations. The Lemma method and the Stem method yield high correlations with human scores. On the other hand, the other three metrics correlate better with human judgments when suffixes are also incorporated as tokens into the evaluation process in addition to the lemma or stem form. The BERTScore metric again shows a good performance when used with the Surface method.

The high correlation ratios obtained with the Lemma tokenization approach may partly be attributed to the success of the Zemberek morphological tool. Zemberek has a high performance in morphological analysis and morphological disambiguation for Turkish [79]. When the Lemma and Stem methods are compared, we see that the Lemma method outperforms the Stem method for both models and for all evaluation metrics. This is the case for both the bare forms of these two methods and their variations. The tokenization methods where the last suffixes are used follow the top-ranking Lemma and Stem methods in BERTurk-cased evaluations, whereas they fall behind the tokenization variations with all suffixes in mT5 evaluations. The motivation behind the last suffix strategy is that the last suffix is considered as one of the most informative morphemes in Turkish [110]. We see that this simple strategy is on par with those that use information of all the suffixes.

Finally, comparing the four text summarization evaluation metrics shows that METEOR yields the best correlation results for both models. Although the underlying tokenization method that yields the best performance is different in the two models (Lemma for BERTurk-cased and Lemma with all suffixes in mT5), we can conclude that the METEOR metric applied to lemmatized system and reference summaries seems as the best metric for text summarization evaluation. This is an interesting result considering that ROUGE is the most commonly used evaluation metric in text summarization.

It should be noted that the Surface method corresponds to the approach used in the evaluation tools for these metrics. That is, the ROUGE, METEOR, BLEU, and BERTScore tools used in the literature mostly follow a simple strategy and work on the surface forms of the words. However, Tables 6.4 and 6.5 show that other strategies such as using the lemma form or using the lemma form combined with the suffixes nearly always outperform this default strategy. This indicates that employing morphosyntactic tokenization processes during evaluation increases correlation with human judgments and thus contributes to the evaluation process.

6.5. Discussion

In this chapter, we introduced various morphosyntactic methods that can be used in text summarization evaluation. We trained state-of-the-art text summarization models on the TR-News dataset. The models were used to generate the system summaries of a set of documents sampled from the test set of TR-News. The relevancy of the system summaries and the reference summaries were manually scored and correlation analysis was performed between the manual scores and the scores produced by the morphosyntactic methods. The correlation analysis revealed that making use of morphosyntactic methods in evaluation metrics outperforms the default strategy of using the surface form for Turkish. We make the manually annotated evaluation dataset publicly available to alleviate the resource scarcity problem in Turkish. We believe that this study will contribute to focus on the importance of preprocessing in

evaluation in this area.

7. CONCLUSION

The focus of this thesis was on abstractive text summarization for morphologically rich languages. Following background information and related work in this topic, in Chapter 4 we addressed the resource scarcity problem by curating two large-scale datasets in Turkish (TR-News) and Hungarian (HU-News). HU-News is the first large-scale text summarization dataset in Hungarian. The datasets' main objective is abstractive text summarization, however they also contain other information that make the datasets suitable for tasks like title generation, topic classification, key phrase extraction, and author detection. Later, we introduced morphological tokenization methods for Turkish and Hungarian, and integrated them into a state-of-the-art abstractive summarization model. Accordingly, we showed that addition of morphological information into the model increases performance for Turkish and provides promising results in Hungarian. All the introduced methods are easily extendable to other morphologically rich languages.

In Chapter 5, we provided state-of-the-art models for text summarization and title generation tasks on both MLSum (Turkish subset) and TR-News datasets by utilizing multilingual pretrained Seq2Seq models. The work done in this chapter was the first study to utilize the titles in title generation task for these datasets. Comprehensive and strong baselines were set.

We studied the evaluation of text summarization for morphologically rich languages in Chapters 5 and 6. We showed the importance of preprocessing such as stemming and removal of punctuation before ROUGE evaluations and how the results can drastically be affected by these operations over a case study in Turkish. Moreover, we introduced morphosyntactic preprocessing methods to address several shortcomings of commonly used evaluation metrics in text summarization. We curated a human judgement dataset to further evaluate these methods through correlation. The correlation analysis revealed that making use of morphosyntactic methods in evaluation

metrics outperforms the default strategy of using the surface form for Turkish. This work can also be extended to other morphologically rich languages.

In addition to the research conducted in this thesis, a Turkish abstractive text summarization tool within TULAP (Turkish Language Processing Platform) was created from the state-of-the-art model obtained in Chapter 5. The tool is currently actively used. All the work in this thesis are made open source and publicly available. We believe that our work will enable more research in this field.

In future works, the morphological tokenization methods proposed in Chapter 4 can be used as a replacement for subword tokenization methods in large pretrained models and the effect of incorporating morphology can be assessed on various downstream tasks. We were initially planning to test this approach, however due to restricted time and insufficient resources, the approach could not be tested. The pretrained Seq2Seq models in Chapter 5 showed a very significant performance improvement on all Turkish datasets. Although we had curated another dataset in Hungarian, we were not able to utilize these models on the HU-News dataset. Hence, we believe that these pretrained Seq2Seq models can also improve the state-of-the-art for HU-News dataset. Lastly, the proposed morphosyntactic approaches in Chapter 6 can be extended by utilizing the root and derivational suffixes. Moreover, the amount of annotated human judgment data was limited. This dataset can be improved by adding more data points and also increasing the number of annotators to have a better generalization. The experiments can further be extended to other morphologically rich languages to better assess the effectiveness of the proposed approaches.

REFERENCES

1. Luhn, H. P., "The Automatic Creation of Literature Abstracts", *IBM Journal of Research and Development*, Vol. 2, No. 2, pp. 159–165, 1958.

2. Edmundson, H. P., "New Methods in Automatic Extracting.", *J. ACM*, Vol. 16, No. 2, pp. 264–285, 1969.

3. Mihalcea, R. and P. Tarau, "TextRank: Bringing Order into Text", *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411, Association for Computational Linguistics, Barcelona, Spain, Jul. 2004.

4. Nallapati, R., F. Zhai and B. Zhou, "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents", *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, p. 3075–3081, AAAI Press, 2017.

5. Rush, A. M., S. Chopra and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, Association for Computational Linguistics, Lisbon, Portugal, Sep. 2015.

6. See, A., P. J. Liu and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks", *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Association for Computational Linguistics, Vancouver, Canada, Jul. 2017.

7. Zhang, J., Y. Zhao, M. Saleh and P. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization", H. D. III and A. Singh (Editors), *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, pp. 11328–11339, PMLR, 13–18

Jul 2020.

8. Bahdanau, D., K. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate.", *Proceedings of the international conference on learning representations (ICLR)*, 2015.

9. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is All you Need", *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc., 2017.

10. Sutskever, I., O. Vinyals and Q. V. Le, "Sequence to Sequence Learning with Neural Networks", *Advances in Neural Information Processing Systems*, Vol. 27, Curran Associates, Inc., 2014.

11. Nallapati, R., B. Zhou, C. dos Santos, c. Gülçehre and B. Xiang, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond", *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, Association for Computational Linguistics, Berlin, Germany, Aug. 2016.

12. Gehrmann, S., Y. Deng and A. Rush, "Bottom-Up Abstractive Summarization", *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109, Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018.

13. Kryściński, W., R. Paulus, C. Xiong and R. Socher, "Improving Abstraction in Text Summarization", *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1808–1817, Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018.

14. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the*

2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Association for Computational Linguistics, Minneapolis, Minnesota, Jun. 2019.

15. Liu, Y. and M. Lapata, "Text Summarization with Pretrained Encoders", *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3730–3740, Association for Computational Linguistics, Hong Kong, China, Nov. 2019.

16. Dong, L., N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou and H.-W. Hon, "Unified Language Model Pre-training for Natural Language Understanding and Generation", H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett (Editors), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 2019.

17. Song, K., X. Tan, T. Qin, J. Lu and T.-Y. Liu, "MASS: Masked Sequence to Sequence Pre-training for Language Generation", *International Conference on Machine Learning*, pp. 5926–5936, 2019.

18. Rothe, S., S. Narayan and A. Severyn, "Leveraging Pre-trained Checkpoints for Sequence Generation Tasks", *Transactions of the Association for Computational Linguistics*, Vol. 8, pp. 264–280, 2020.

19. Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Association for Computational Linguistics, Online, Jul. 2020.

20. Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li

and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67, 2020.

21. Qi, W., Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang and M. Zhou, "Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 2401–2410, 2020.

22. Özsoy, M. G., İ. Çiçekli and F. N. Alpaslan, "Text Summarization of Turkish Texts Using Latent Semantic Analysis", *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, p. 869–876, Association for Computational Linguistics, USA, 2010.

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

24. Kartal, Y. S. and M. Kutlu, "Machine Learning Based Text Summarization for Turkish News", *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2020.

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

26. Scialom, T., P.-A. Dray, S. Lamprier, B. Piwowarski and J. Staiano, "MLSUM: The Multilingual Summarization Corpus", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8051–8067, Association for Computational Linguistics, Online, Nov. 2020.

27. Beken Fikri, F., K. Oflazer and B. Yanikoglu, "Semantic Similarity Based Evalu-

ation for Abstractive News Summarization", *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, Association for Computational Linguistics, Online, Aug. 2021.

28. Beke, A. and G. Szaszák, "Automatic Summarization of Highly Spontaneous Speech", *Speech and Computer - 18th International Conference, SPECOM 2016, Budapest, Hungary, August 23-27, 2016, Proceedings*, Vol. 9811 of *Lecture Notes in Computer Science*, pp. 140–147, Springer, 2016.

29. Tündik, M. Á., V. Kaszás and G. Szaszák, "Assessing the Semantic Space Bias Caused by ASR Error Propagation and its Effect on Spoken Document Summarization", *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 1333–1337, ISCA, 2019.

30. Güngör, O., T. Güngör and S. Üsküdarlı, "The effect of morphology in named entity recognition with sequence tagging", *Natural Language Engineering*, Vol. 25, No. 1, pp. 147–169, 2019.

31. Eşref, Y. and B. Can, "Using morpheme-level attention mechanism for Turkish sequence labelling", *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2019.

32. Döbrössy, B., M. Makrai, B. Tarján and G. Szaszák, "Investigating sub-word embedding strategies for the morphologically rich and free phrase-order Hungarian", *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 187–193, 2019.

33. Üstün, A., M. Kurfalı and B. Can, "Characters or Morphemes: How to Represent Words?", *Proceedings of The Third Workshop on Representation Learning for NLP*, pp. 144–153, Association for Computational Linguistics, Melbourne, Australia, Jul. 2018.

34. Pan, Y., X. Li, Y. Yang and R. Dong, "Morphological Word Segmentation on Agglutinative Languages for Neural Machine Translation.", *CoRR*, Vol. abs/2001.01589, 2020.

35. Lin, C.-Y., "ROUGE: A Package for Automatic Evaluation of Summaries", *Text Summarization Branches Out*, pp. 74–81, Association for Computational Linguistics, Barcelona, Spain, Jul. 2004.

36. Banerjee, S. and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments", *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Association for Computational Linguistics, Ann Arbor, Michigan, Jun. 2005.

37. Nuzumlalı, M. Y. and A. Özgür, "Analyzing Stemming Approaches for Turkish Multi-Document Summarization", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 702–706, Association for Computational Linguistics, Doha, Qatar, Oct. 2014.

38. Körtvélyessy, L., "Essentials of Language Typology", *Košice: UPJŠ. Available online*, 2017.

39. Oflazer, K., "Turkish and its challenges for language processing", *Language Resources and Evaluation*, Vol. 48, No. 4, pp. 639–653, 2014.

40. Erguvanlı, E. E. and E. E. Taylan, *The function of word order in Turkish grammar*, Vol. 106, Univ of California Press, 1984.

41. Kiefer, F., *On emphasis and word order in Hungarian*, Vol. 76, Psychology Press, 1997.

42. Hochreiter, S. and J. Schmidhuber, "Long short-term memory", *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

43. Grosse, R., "Lecture 15: Exploding and vanishing gradients", *University of Toronto Computer Science*, 2017.

44. Schuster, M. and K. K. Paliwal, "Bidirectional recurrent neural networks.", *IEEE Trans. Signal Process.*, Vol. 45, No. 11, pp. 2673–2681, 1997.

45. Cho, K., B. van Merriënboer, c. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Association for Computational Linguistics, Doha, Qatar, Oct. 2014.

46. Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 of *Proceedings of Machine Learning Research*, pp. 2048–2057, PMLR, Lille, France, 07–09 Jul 2015.

47. Luong, T., H. Pham and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Association for Computational Linguistics, Lisbon, Portugal, Sep. 2015.

48. Cheng, J., L. Dong and M. Lapata, "Long Short-Term Memory-Networks for Machine Reading", *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 551–561, Association for Computational Linguistics, Austin, Texas, Nov. 2016.

49. Paulus, R., C. Xiong and R. Socher, "A Deep Reinforced Model for Abstractive Summarization", *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

50. Papineni, K., S. Roukos, T. Ward and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation", *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, Jul. 2002.

51. Porter, M. F., "An algorithm for suffix stripping.", *Program*, Vol. 40, No. 3, pp. 211–218, 2006.

52. Miller, G. A., "WordNet: A Lexical Database for English", *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994, `https://aclanthology.org/H94-1111`.

53. Chopra, S., M. Auli and A. M. Rush, "Abstractive Sentence Summarization with Attentive Recurrent Neural Networks", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 93–98, Association for Computational Linguistics, San Diego, California, Jun. 2016.

54. Çelikyılmaz, A., A. Bosselut, X. He and Y. Choi, "Deep Communicating Agents for Abstractive Summarization", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1662–1675, Association for Computational Linguistics, New Orleans, Louisiana, Jun. 2018.

55. Gehrmann, S., Z. Ziegler and A. Rush, "Generating Abstractive Summaries with Finetuned Language Models", *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 516–522, Association for Computational Linguistics, Tokyo, Japan, Oct.–Nov. 2019.

56. Narayan, S., S. B. Cohen and M. Lapata, "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization", *Proceedings of the 2018 Conference on Empirical Methods in Natural*

Language Processing, pp. 1797–1807, Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018.

57. Radford, A., K. Narasimhan, T. Salimans and I. Sutskever, "Improving language understanding by generative pre-training", , 2018.

58. Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding", H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett (Editors), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 2019.

59. Xue, L., N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua and C. Raffel, "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer", *ArXiv*, Vol. abs/2010.11934, 2021.

60. Liu, Y., J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis and L. Zettlemoyer, "Multilingual Denoising Pre-training for Neural Machine Translation", *Transactions of the Association for Computational Linguistics*, Vol. 8, pp. 726–742, 2020.

61. Altan, Z., "A Turkish automatic text summarization system", *IASTED International Conference on*, 2004.

62. Pembe, F. C. and T. Güngör, "Towards a new summarization approach for search engine results: An application for Turkish", *2008 23rd International Symposium on Computer and Information Sciences*, pp. 1–6, IEEE, 2008.

63. Güran, A., N. G. Bayazit and M. Z. Gürbüz, "Efficient feature integration with Wikipedia-based semantic feature extraction for Turkish text summarization", *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 21, No. 5, pp. 1411–1425, 2013.

64. Karakoç, E. and B. Yılmaz, "Deep Learning Based Abstractive Turkish News Summarization", *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2019.

65. Kusner, M., Y. Sun, N. Kolkin and K. Weinberger, "From Word Embeddings To Document Distances", F. Bach and D. Blei (Editors), *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 of *Proceedings of Machine Learning Research*, pp. 957–966, PMLR, Lille, France, 07–09 Jul 2015.

66. Chow, J., L. Specia and P. Madhyastha, "WMDO: Fluency-based Word Mover's Distance for Machine Translation Evaluation", *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 494–500, Association for Computational Linguistics, Florence, Italy, Aug. 2019.

67. Mikolov, T., K. Chen, G. S. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space", , 2013, `http://arxiv.org/abs/1301.3781`.

68. Zhang, T., V. Kishore, F. Wu, K. Q. Weinberger and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT", , 2019.

69. Sennrich, R., B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Association for Computational Linguistics, Berlin, Germany, Aug. 2016.

70. Kudo, T., "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates", *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, Association for Computational Linguistics, Melbourne, Australia, Jul. 2018.

71. Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz

Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", *CoRR*, Vol. abs/1609.08144, 2016.

72. Creutz, M. and K. Lagus, "Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0", *Helsinki University of Technology*, 2005.

73. Bostrom, K. and G. Durrett, "Byte Pair Encoding is Suboptimal for Language Model Pretraining", *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4617–4624, Association for Computational Linguistics, Online, Nov. 2020.

74. Huck, M., S. Riess and A. Fraser, "Target-side word segmentation strategies for neural machine translation", *Proceedings of the Second Conference on Machine Translation*, pp. 56–67, 2017.

75. Tawfik, A., M. Emam, K. Essam, R. Nabil and H. Hassan, "Morphology-aware Word-Segmentation in Dialectal Arabic Adaptation of Neural Machine Translation", *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 11–17, Association for Computational Linguistics, Florence, Italy, Aug. 2019.

76. Nemeskey, D. M., "`emMorph` a Hungarian Language Modeling baseline", *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017)*, pp. 91–102, Szeged, 2017.

77. Sandhaus, E., "The New York Times annotated corpus", *Linguistic Data Consortium, Philadelphia*, Vol. 6, No. 12, 2008.

78. Kettunen, K., "Can Type-Token Ratio be Used to Show Morphological Complex-

ity of Languages?", *Journal of Quantitative Linguistics*, Vol. 21, No. 3, p. 223–245, 2014.

79. Akın, A. A. and M. D. Akın, "Zemberek, an open source NLP framework for Turkic languages", *Structure*, Vol. 10, pp. 1–5, 2007.

80. Simon, E., B. Indig, A. Kalivoda, I. Mittelholcz, B. Sass and N. Vadasz, "Újabb fejlemények az e-magyar háza táján", *XVI. Magyar Számítógépes Nyelvészeti Konferencia*, pp. 29–42, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged, 2020.

81. Indig, B., B. Sass, E. Simon, I. Mittelholcz, N. Vadász and M. Makrai, "One format to rule them all – The `emtsv` pipeline for Hungarian", *Proceedings of the 13th Linguistic Annotation Workshop*, pp. 155–165, Association for Computational Linguistics, Florence, Italy, aug 2019.

82. Indig, B., B. Sass, E. Simon, I. Mittelholcz, P. Kundráth and N. Vadász, "`emtsv` — Egy formátum mind felett", *XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019)*, pp. 235–247, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged, 2019.

83. Váradi, T., E. Simon, B. Sass, I. Mittelholcz, A. Novák, B. Indig, R. Farkas and V. Vincze, "E-magyar – A Digital Language Processing System", *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Miyazaki, Japan, May 7-12, 2018 2018.

84. Váradi, T., E. Simon, B. Sass, M. Gerőcs, I. Mittelholtz, A. Novák, B. Indig, G. Prószéky and V. Vincze, "Az e-magyar digitális nyelvfeldolgozó rendszer", *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017)*, pp. 49–60, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged, 2017.

85. Nemeskey, D. M., "Egy `emBERT` próbáló feladat", *XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2020)*, pp. 409–418, Szeged, 2020.

86. Duchi, J. C., E. Hazan and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.", *J. Mach. Learn. Res.*, Vol. 12, pp. 2121–2159, 2011.

87. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

88. Schweter, S., *BERTurk - BERT models for Turkish*, Apr. 2020, `https://doi.org/10.5281/zenodo.3770924`.

89. Nemeskey, D. M., *Natural Language Processing Methods for Language Modeling*, Ph.D. Thesis, Eötvös Loránd University, 2020.

90. Baykara, B. and T. Güngör, "Abstractive text summarization and new large-scale datasets for agglutinative languages Turkish and Hungarian", *Language Resources and Evaluation*, pp. 1–35, 2022.

91. Virtanen, A., J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter and S. Pyysalo, "Multilingual is not enough: BERT for Finnish", *CoRR*, Vol. abs/1912.07076, 2019.

92. Polignano, M., P. Basile, M. de Gemmis, G. Semeraro and V. Basile, "AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets", *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, Vol. 2481, CEUR, 2019.

93. Kuratov, Y. and M. Arkhipov, "Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language", *CoRR*, Vol. abs/1905.07213, 2019.

94. Chan, B., S. Schweter and T. Möller, "German's Next Language Model", *CoRR*, Vol. abs/2010.10906, 2020.

95. Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", *CoRR*, Vol. abs/1907.11692, 2019.

96. Wenzek, G., M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin and E. Grave, "CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data", *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 4003–4012, European Language Resources Association, Marseille, France, May 2020.

97. Rajpurkar, P., J. Zhang, K. Lopyrev and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text", *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Association for Computational Linguistics, Austin, Texas, Nov. 2016.

98. Hu, J., S. Ruder, A. Siddhant, G. Neubig, O. Firat and M. Johnson, "XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation", H. D. III and A. Singh (Editors), *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, pp. 4411–4421, PMLR, 13–18 Jul 2020.

99. Conneau, A., K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Association for Computational Linguistics, Online, Jul. 2020.

100. Rust, P., J. Pfeiffer, I. Vulic, S. Ruder and I. Gurevych, "How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models",

CoRR, Vol. abs/2012.15613, 2020.

101. Ng, A., J. Ngiam, C. Y. Foo and Y. Mai, "Deep learning", *CS229 Lecture Notes*, pp. 1–30, 2014.

102. Shazeer, N. and M. Stern, "Adafactor: Adaptive Learning Rates with Sublinear Memory Cost", J. Dy and A. Krause (Editors), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 4596–4604, PMLR, 10–15 Jul 2018.

103. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest and A. Rush, "Transformers: State-of-the-Art Natural Language Processing", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Association for Computational Linguistics, Online, Oct. 2020.

104. Torres-Moreno, J.-M., *Automatic text summarization*, John Wiley & Sons, 2014.

105. Narayan, S., S. B. Cohen and M. Lapata, "Ranking Sentences for Extractive Summarization with Reinforcement Learning", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1747–1759, Association for Computational Linguistics, New Orleans, Louisiana, Jun. 2018.

106. Koupaee, M. and W. Y. Wang, "WikiHow: A Large Scale Text Summarization Dataset", , 2018.

107. Graham, Y., "Re-evaluating Automatic Summarization with BLEU and 192 Shades of ROUGE", *EMNLP*, 2015.

108. Parida, S. and P. Motlícek, "Abstract Text Summarization: A Low Resource

Challenge", *EMNLP*, 2019.

109. Landis, J. R. and G. G. Koch, "The measurement of observer agreement for categorical data.", *Biometrics*, Vol. 33 1, pp. 159–74, 1977.

110. Oflazer, K., B. Say, D. Z. Hakkani-Tür and G. Tür, *Building a Turkish Treebank*, pp. 261–277, Springer Netherlands, Dordrecht, 2003.

APPENDIX A: ADDITIONAL TABLES REGARDING CHAPTER 4

Table A.1. A detailed example for Hungarian morphological parsing and disambiguation.

Word	Possible Analysis	Disambiguated Analysis
a	a[/Det|Pro|(Post)] a[/Det|art.Def] a[/N|Ltr] + [Nom] a[/N|Pro|(Post)] + [Nom]	a [/Det|art.Def] Det
tanulók	tanul[/V] + ó[_ImpfPtcp/Adj] + k[Pl] + [Nom] tanuló[/N] + k[Pl] + [Nom]	tanuló [/N][Pl][Nom] N
igényeihez	igény[/N] + ei[Pl.Poss.3Sg] + hez[All]	igény [/N][Pl.Poss.3Sg][All] N
kell	kell[/V] + [Prs.NDef.3Sg]	kell [/V][Prs.NDef.3Sg] V
igazodniuk	igaz[/Adj] + odik[_AdjVbz_Ntr/V]=od + niuk[Inf.3Pl]	igazodik [/V][Inf.3Pl] V

Table A.1 (cont.)

a	a[/Det|Pro|(Post)] a[/Det|art.Def] a[/N|Ltr] + [Nom] a[/N|Pro|(Post)] + [Nom]	a [/Det|art.Def] Det
nyelvvizsga	nyelv[/N] + vizsga[/N] + [Nom] nyelvvizsga[/N] + [Nom]	nyelvvizsga [/N][Nom] N
követelményeinek	követelmény[/N] + ei[Pl.Poss.3Sg] + nek[Dat]	követelmény [/N][Pl.Poss.3Sg][Dat] N
is	is[/Adv]	is [/Adv] Adv

Table A.2. A detailed example for Turkish morphological parsing and disambiguation.

Word	Morphological Parses	Disambiguated Analysis
şampiyon	[şampiyon:Noun] şampiyon:Noun+A3sg	[şampiyon:Noun] şampiyon:Noun+A3sg
yüzücünün	[yüzmek:Verb] yüz:Verb|ücü:Agt→Noun+A3sg+nün:Gen [yüzmek:Verb] yüz:Verb|ücü:Agt→Noun+A3sg+n:P2sg+ün:Gen	[yüzmek:Verb] yüz:Verb|ücü:Agt→Noun+A3sg+nün:Gen
abd	[ABD:Noun,Abbrv] abd:Noun+A3sg [abd:Noun] abd:Noun+A3sg]	[ABD:Noun,Abbrv] abd:Noun+A3sg
kongre	[kongre:Noun] kongre:Noun+A3sg	[kongre:Noun] kongre:Noun+A3sg
baskınındaki	[baskın:Noun] baskın:Noun+A3sg+ın:P2sg+da:Loc|ki:Rel→Adj [baskın:Noun] baskın:Noun+A3sg+ı:P3sg+nda:Loc|ki:Rel→Adj [baskın:Adj] baskın:Adj|Zero→Noun+A3sg+ın:P2sg+da:Loc|ki:Rel→Adj [baskın:Adj] baskın:Adj|Zero→Noun+A3sg+ı:P3sg+nda:Loc|ki:Rel→Adj	[baskın:Noun] baskın:Noun+A3sg+ı:P3sg+nda:Loc|ki:Rel→Adj
görüntüleri	[görüntü:Noun] görüntü:Noun+A3sg+leri:P3pl [görüntü:Noun] görüntü:Noun+ler:A3pl+i:Acc [görüntü:Noun] görüntü:Noun+ler:A3pl+i:P3sg [görüntü:Noun] görüntü:Noun+ler:A3pl+i:P3pl	[görüntü:Noun] görüntü:Noun+ler:A3pl+i:Acc

Table A.2 (cont.)

ortaya	[orta:Noun] orta:Noun+A3sg+ya:Dat [Orta:Noun,Prop] orta:Noun+A3sg+ya:Dat [orta:Adj] orta:Adj|Zero→Noun+A3sg+ya:Dat [ortay:Adj] ortay:Adj|Zero→Noun+A3sg+a:Dat	[orta:Noun] orta:Noun+A3sg+ya:Dat
çıktı	[çıkmak:Verb] çık:Verb+tı:Past+A3sg [çıktı:Noun] çıktı:Noun+A3sg	[çıkmak:Verb] çık:Verb+tı:Past+A3sg
.	[.:Punc] .:Punc	[.:Punc] .:Punc

APPENDIX B: ADDITIONAL TABLES REGARDING CHAPTER 5

Table B.1. Cross-dataset evaluation results for the summary generation task.

Model & Training Set	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
mT5-TR-News	41.13	25.75	37.6	40.99	26.24	36.77	41.06	25.97	37.22
mT5-MLSum-TR	37.25	22.1	33.66	42.26	27.81	37.96	39.52	24.69	35.61
mT5-Combined-TR	41.23	25.98	37.73	44.01	29.49	39.79	42.49	27.58	38.67

Table B.2. Cross-dataset evaluation results for the title generation (abstract as input) task.

Model & Training Set	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
mT5-TR-News	41.87	24.49	40.87	41.81	23.08	39.87	41.84	23.87	40.41
mT5-MLSum-TR	36.32	19.05	35.3	40.77	22.42	38.97	38.34	20.59	36.97
mT5-Combined-TR	42.46	24.96	41.41	43.79	25.32	41.81	43.04	25.14	41.59

Table B.3. The analysis results for the summary generation task given various beam sizes and early-stopping method.

Parameters	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
mT5-beam-1	40.74	24.98	37.3	40.87	25.83	36.65	41.99	26.51	38.26
mT5-beam-2	41.34	25.81	37.9	42.13	27.44	37.85	42.61	27.48	38.84
mT5-beam-3	41.3	25.87	37.8	42.18	27.66	37.92	42.59	27.62	38.82
mT5-beam-4	41.13	25.75	37.6	42.26	27.81	37.96	42.49	27.58	38.67
mT5-beam-4 & early-stopping	41.15	25.74	37.86	41.36	26.92	37.32	42.18	27.22	38.61

Table B.4. The analysis results for the title generation (abstract as input) task given various beam sizes and early-stopping method.

Parameters	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
mT5-beam-1	40.41	22.76	39.41	37.93	19.81	36.25	41.12	23.11	39.75
mT5-beam-2	41.58	24.1	40.56	39.95	21.65	39.19	42.49	24.51	41.08
mT5-beam-3	41.82	24.39	40.81	40.54	22.22	38.76	42.91	25	41.49
mT5-beam-4	41.87	24.49	40.87	40.77	22.42	38.97	43.04	25.14	41.59
mT5-beam-4 & early-stopping	41.66	24.17	40.73	40.04	21.7	38.3	42.53	24.57	41.18

Table B.5. ROUGE scores with different preprocessing settings for the summary generation task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations.

Parameters	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
Punct removed Stems taken	41.13	25.75	37.60	42.26	27.81	37.96	42.49	27.58	38.67
Punct removed Stems not taken	37.60	23.93	34.89	39.03	26.22	35.57	39.12	25.85	36.12
Punct kept Stems taken	43.64	25.75	39.66	44.60	27.67	39.90	44.83	27.46	40.59
Punct kept Stems not taken	40.55	24.17	37.34	41.76	26.29	37.86	41.88	25.94	38.41

Table B.6. ROUGE scores with different preprocessing settings for the title generation (abstract as input) task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations.

Parameters	TR-News			MLSum (TR)			Combined-TR		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
Punct removed Stems taken	41.87	24.49	40.87	40.77	22.42	38.97	43.04	25.14	41.59
Punct removed Stems not taken	37.91	22.30	37.15	37.22	20.65	35.79	39.24	23.05	38.10
Punct kept Stems taken	40.00	23.02	39.02	39.23	20.79	37.37	41.09	23.44	39.70
Punct kept Stems not taken	36.35	21.07	35.60	35.92	19.20	34.42	37.56	21.57	36.47

Table B.7. ROUGE-1 scores of all the models calculated under different preprocessing settings on the TR-News dataset for the text summarization task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations.

Parameters	TR-News Text Summarization Task					
	mT5	BERTurk-uncased	BERTurk-cased	mBART	mBERT-uncased	mBERT-cased
Punct removed Stems taken	**41.13**	40.50	41.06	40.52	33.04	39.73
Punct removed Stems not taken	37.60	37.13	**37.63**	36.97	30.38	36.22
Punct kept Stems taken	**43.64**	42.34	42.85	43.05	35.78	41.37
Punct kept Stems not taken	**40.55**	39.88	39.43	39.95	33.52	38.30

Table B.8. ROUGE-1 scores of all the models calculated under different preprocessing settings on the MLSum (TR) dataset for the text summarization task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations.

Parameters	MLSum (TR) Text Summarization Task					
	mT5	BERTurk-uncased	BERTurk-cased	mBART	mBERT-uncased	mBERT-cased
Punct removed Stems taken	**42.26**	41.47	41.48	40.47	33.59	40.27
Punct removed Stems not taken	**39.03**	38.35	38.40	37.27	31.27	37.16
Punct kept Stems taken	**44.60**	43.33	43.28	42.95	41.89	36.28
Punct kept Stems not taken	**41.76**	40.62	40.59	40.14	34.28	39.15

Table B.9. ROUGE-1 scores of all the models calculated under different preprocessing settings on the Combined-TR dataset for the text summarization task. ”Punct removed” refers to removing the punctuations, whereas “Punct kept” refers to keeping the punctuations before the ROUGE calculations. "Stems taken" refers to applying stemming operation on the words, whereas "Stems not taken" refers to leaving the words in their surface form before the ROUGE calculations.

Parameters	Combined-TR Text Summarization Task					
	mT5	BERTurk-uncased	BERTurk-cased	mBART	mBERT-uncased	mBERT-cased
Punct removed Stems taken	42.49	42.51	**42.75**	41.97	34.13	41.20
Punct removed Stems not taken	39.12	39.20	**39.47**	38.56	31.60	37.82
Punct kept Stems taken	**44.83**	44.14	44.33	44.32	36.70	42.69
Punct kept Stems not taken	**41.88**	41.28	41.48	41.34	34.55	39.73

APPENDIX C: ADDITIONAL TABLES REGARDING CHAPTER 6

Table C.1. Pearson correlation results of the morphosyntactic methods without prefix tokens for the BERTurk-cased summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column.

BERTurk-cased						
	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	BERTScore
Surface	0.770	0.723	0.750	0.736	0.649	**0.800**
Lemma	**0.831**	**0.744**	**0.795**	**0.809**	0.671	<u>0.775</u>
Stem	<u>0.815</u>	<u>0.738</u>	<u>0.777</u>	<u>0.799</u>	0.668	0.768
Lemma and all suffixes	0.796	0.737	0.762	0.783	<u>0.768</u>	0.746
Stem and all suffixes	0.789	0.736	0.757	0.779	0.766	0.745
Lemma and combined suffixes	0.798	0.727	0.769	0.793	0.763	0.752
Stem and combined suffixes	0.789	0.725	0.758	0.789	0.759	0.753
Lemma and last suffix	0.807	0.733	0.769	0.789	**0.773**	0.756
Stem and last suffix	0.795	0.732	0.757	0.784	<u>0.768</u>	0.757

Table C.2. Pearson correlation results of the morphosyntactic methods without prefix tokens for the mT5 summarization model. Bold and underline denote, respectively, the best score and the second-best score for a column.

mT5	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	BERTScore
Surface	0.682	0.648	0.693	0.697	0.591	**0.693**
Lemma	**0.713**	**0.677**	**0.708**	0.737	0.602	0.682
Stem	0.696	<u>0.659</u>	0.693	0.716	0.594	0.675
Lemma and all suffixes	<u>0.702</u>	0.648	0.691	0.730	**0.701**	0.671
Stem and all suffixes	0.693	0.642	0.688	0.721	<u>0.695</u>	0.666
Lemma and combined suffixes	0.691	0.652	0.690	**0.748**	0.678	0.687
Stem and combined suffixes	0.680	0.643	0.679	0.737	0.669	<u>0.690</u>
Lemma and last suffix	0.700	0.656	<u>0.702</u>	<u>0.741</u>	0.678	0.656
Stem and last suffix	0.688	0.647	0.690	0.730	0.669	0.652