T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A FAITHFULNESS-AWARE PRETRAINING STRATEGY FOR ABSTRACTIVE TEXT SUMMARIZATION

MSc THESIS

Mohanad ALREFAAI

Computer and Information Engineering Department

Computer Engineering Program

DECEMBER 2023

T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A FAITHFULNESS-AWARE PRETRAINING STRATEGY FOR ABSTRACTIVE TEXT SUMMARIZATION

MSc THESIS

Mohanad ALREFAAI

Computer and Information Engineering Department

Computer Engineering Program

Thesis Advisor: Prof. Dr. Devrim AKGÜN

DECEMBER 2023

The thesis work titled "A Faithfulness-Aware Pretraining Strategy for Abstractive Text Summarization" prepared by Mohanad Alrefaai was accepted by the following jury on 28/12/2023 by unanimously of votes as a MSc THESIS in Sakarya University Graduate School of Natural and Applied Sciences, Computer and Information Engineering department, Computer Engineering program.

Thesis Jury

Head of Jury :	**Doç. Dr. Zehra KARAPINAR ŞENTÜRK** Duzce University
Jury Member :	**Prof. Dr. Devrim AKGÜN (Advisor)** Sakarya University
Jury Member :	**Doç. Dr. Ünal ÇAVUŞOĞLU** Sakarya University

STATEMENT OF COMPLIANCE WITH THE ETHICAL PRINCIPLES AND RULES

I declare that the thesis work titled "A FAITHFULNESS-AWARE PRETRAINING STRATEGY FOR ABSTRACTIVE TEXT SUMMARIZATION", which I have prepared in accordance with Sakarya University Graduate School of Natural and Applied Sciences regulations and Higher Education Institutions Scientific Research and Publication Ethics Directive, belongs to me, is an original work, I have acted in accordance with the regulations and directives mentioned above at all stages of my study, I did not get the innovations and results contained in the thesis from anywhere else, I duly cited the references for the works I used in my thesis, I did not submit this thesis to another scientific committee for academic purposes and to obtain a title, in accordance with the articles 9/2 and 22/2 of the Sakarya University Graduate Education and Training Regulation published in the Official Gazette dated 20.04.2016, a report was received in accordance with the criteria determined by the graduate school using the plagiarism software program to which Sakarya University is a subscriber, I accept all kinds of legal responsibility that may arise in case of a situation contrary to this statement.

(28/12/2023)

Mohanad ALREFAAİ

To my beloved family, the foundation of my dreams and aspirations

ACKNOWLEDGMENTS

The completion of this master's thesis marks a significant milestone in my academic journey, and I am deeply grateful to the numerous individuals who played a crucial role in making it possible.

I owe a profound debt of gratitude to my advisor, Professor Devrim AKGÜN, whose unwavering guidance, insightful feedback, and constant encouragement were the cornerstones of my research journey. Their passion for the field and dedication to student success fueled my own ambition and allowed me to explore the subject's depth.

The exceptional environment and resources provided by Sakarya University were essential to my research endeavors. I deeply appreciate the opportunity to learn from renowned professors and access cutting-edge facilities, which undeniably enriched my academic experience.

Words cannot express my heartfelt gratitude to my family, particularly my parents and my wife, for their unwavering love, unwavering support, and endless patience throughout this challenging yet fulfilling process. Your understanding, encouragement, and shared sacrifices were the bedrock upon which I built my success.

Finally, I would like to acknowledge the invaluable friendship and support of my friends. You provided me with much-needed breaks from the academic rigors and reminded me of the importance of balance and personal growth. Your camaraderie, laughter, and shared experiences have made this journey truly enjoyable and enriching.

Thank you all for being a part of this incredible journey. I am forever indebted to your support and guidance, and I carry your love and encouragement with me as I embark on the next chapter of my academic career.

Mohanad ALREFAAİ

TABLE OF CONTENTS

Page

ABBREVIATIONS

BART : Bidirectional and Autoregressive Transformer
BERT : Bidirectional Encoder Representations from Transformers
BoW : Bag of Words
BS-Fact : BERT Score between summaries and source documents
FNN : Feedforward Neural Network
GloVe : Global vectors for word representation
GRU : Gated Recurrent Units
LCS : Longest Common Subsequence
LLM : Large Language Model
LR : Learning Rate
LSTM : Long Short-Term Memory
MLM : Masked Language Modeling
MLP : Multilayer Perceptron
NER : Named Entity Recognition
NLP : Natural Language Processing
NSP : Next Sentence Prediction
QA : Question Answering
RNN : Recurrent Neural Network
RoBERTa : Robustly Optimized BERT Pretraining Approach
ROUGE : Recall Oriented Understudy for Gisting Evaluation
Seq2seq : Sequence to sequence model
TF-IDF : Term Frequency-Inverse Document Frequency
Word2vec : Word to Vector

LIST OF TABLES

Page

LIST OF FIGURES

Page

A FAITHFULNESS-AWARE PRETRAINING STRATEGY FOR ABSTRACTIVE TEXT SUMMARIZATION

SUMMARY

One of the main challenges in abstractive text summarizing is maintaining the faithfulness of the generated summaries compared to the source documents. In abstractive text summarizing, the term "faithfulness" refers to the degree to which a summary accurately and completely captures the essential information from the source text while maintaining the overall meaning and context.

Recent works have made remarkable progress in addressing the issue of faithfulness in abstractive text summarization from several perspectives. For instance, some works suggested a post-process method to refine faithfulness. Others focused on the relationship between the decoding generation phase of the generative model and faithfulness. Furthermore, many studies put efforts into customizing the training phase in order to improve faithfulness. Nevertheless, these researches fail to adequately explore a central aspect, which is how pretraining strategies can impact and enhance the accuracy and reliability of faithfulness in abstractive text summarization.

To address this problem, we have introduced an innovative pretraining strategy that stimulates the BART large language model to attend more to tokens and contexts correlated with faithfulness of the source text. To assess our approach, we conducted a thorough examination of its effects on both faithfulness and summarization. Our research revealed that the proposed technique improves the model's attention to the critical contexts that are strongly connected to the faithfulness of the original text. Furthermore, our experiments and analysis demonstrated that the introduced method outperforms the baseline model, which is pretrained using the traditional MLM techniques, in terms of different faithfulness metrics, such as QuestEval and BS-Fact metrics, in two downstream abstractive text summarization datasets.

In addition, we investigated the possibility that the pretraining processes that were provided could improve the quality of the summaries that were created. This was determined by using summarization metrics such as ROUGE-N and BERT-Score.

SOYUTLAYICI METİN ÖZETLEME İÇİN SADAKAT-FARKINDA BİR ÖN EĞİTİM STRATEJİSİ

ÖZET

Metin özetlemesi, bir metinden anahtar noktaları çıkarmak ve metnin özünü yakalayan kesin bir temsil oluşturmakla ilgilidir. Bu süreç, bilgi zengini bir dünyada bilgi korumayı ve anlamayı kolaylaştırır. Soyutlama, temel kavramları kısa ve tutarlı bir şekilde iletmek için orijinal materyalin sıkıştırılmasını, yeniden ifade edilmesini ve kaynak metin sözcüklerinden farklı sözcükler kullanılarak yeni cümleler oluşturulmasını içerir. Derin Öğrenme son yıllarda soyutlayıcı metin özetlemede önemli ilerlemelere yol açmıştır. Soyutlama oluşturmanın geleneksel yolu, LSTM ve GRU gibi tekrarlayan yapıdaki sinir ağlarından (RNN) oluşan diziden diziye (seq2seq) modellerini kullanmaktır. Ancak RNN modelleri, giriş dizilerindeki kelimeler arasındaki anlamsal ve bağlamsal ilişkilerin anlaşılması ve yavaş hesaplama sorunu yaşamaktadır. Transformer mimarisi, özellikle metnin yeniden ifade edilmesi, makine çevirisi ve metin özetleme gibi metin oluşturma faaliyetlerinde doğal dil işleme teknolojisi (NLP) alanını önemli ölçüde etkilemiştir. Büyük dil modelleri (LLM) modelleri son yıllarda yapılan çalışmalarda giderek daha fazla kullanılmakta ve soyutlayıcı metin özetlemede önemli başarılar elde etmektedir. Bu gelişmelere rağmen son dönemde yapılan çalışmalar hazırlanan özetlerde "halüsinasyon" olarak adlandırılan bir durumunun ön plana çıktığını göstermektedir. Halüsinasyon, kaynak metindeki bazı önemli ifadelerin ve öğelerin özetten çıkarıldığı, konu dışı bilgilerin yanlışlıkla dahil edildiği anlamına gelir. Bu durum özetin kaynak materyale ne ölçüde sadık kaldığını vurgulama ihtiyacını ortaya çıkarmaktadır.

Çok sayıda araştırma soyutlayıcı metin özetlemenin doğruluğu üzerine araştırmalar gerçekleştirmiştirr. Bu çalışmalar üç ana kategoriye ayrılabilir: süreç sonrası yaklaşımlar, sadakat bilinci oluşturma yöntemleri ve özel eğitim yöntemleri. Bazı süreç sonrası yaklaşımlar, özeti oluşturduktan sonra halüsinasyonlu varlıklar sorununu çözmeyi amaçlamaktadır. Bu süreç, halüsinasyonlu varlıkların tanımlanmasını ve daha doğru adlandırılmış varlıklarla değiştirilmesini içerir. Ayrıca, diğer çalışmalar alternatif bir süreç sonrası strateji uygulamak için karşılaştırmalı bir öğrenme yaklaşımı kullanmıştır. Öte yandan, sadakat bilincine sahip üretim stratejileri, kod çözme aşamasında sadakati önceliklendirmek için ışın aramayı (beam search) kullanarak özet sürecinin üretim aşamasında sadakati artırmaya odaklanır. Bunun yanında bazı çalışmalar sadakati geliştirmek için özelleştirilmiş eğitim yöntemleri önermektedir.

Birçok çalışma metin oluşturma görevleri için özelleştirilmiş ön eğitim hedeflerini kullanılmıştır. Örneğin, soyutlayıcı metin özetlemede gerçekçiliği geliştirmek için özel bir sadakat-farkındalığı ön eğitim stratejisi tanıtılmıştır. Ek olarak, adlandırılmış varlıkların soyutlayıcı özetlemeye dahil edilmesini geliştirmek için başka bir özelleştirilmiş ön eğitim yöntemi önerilmiştir. Ancak bu çalışmalar öncelikle model

düzeltme ve son işlemlere odaklanmakta ve ön eğitimin kritik rolünü ihmal etmektedir.

Bu araştırmanın amacı ön eğitim yöntemlerinin sadakat üzerindeki etkisini araştırmak ve bunu geliştirmeye yönelik yeni bir yaklaşım sunmaktır. Önerilen ön eğitim stratejisi, BART büyük dil modelini sadakatle güçlü bir şekilde ilişkili olan belirteçlere ve varlıklara öncelik verme konusunda yönlendirir. Model bu belirteçlerin bağlamsal temsillerine yönelirse daha aslına uygun özetler üretme olasılığı daha yüksektir. Ön eğitim sürecini üç ayırt edilebilir adım oluşturur. İlk adım, kaynak belgelerde yer alan her cümle için bir derecelendirme sağlamak amacıyla eğitim öncesi veri kümesini ön işlemek olacaktır. Bunu takiben her cümleye verilen önem doğrultusunda belirteçleri (token) seçici olarak maskelenmiştir. Daha yüksek önceliğe sahip cümleler, daha düşük önceliğe sahip cümlelere göre daha önemli sayıda maskelenmiş belirtece sahiptir.

Çalışmada LLM modellerini sadakat üzerinde olumlu etkisi olan varlıklara ve belirteçlere daha fazla öncelik vererek yönlendirmek amaçlanmıştır. Maskeli dil modelleme (MLM), BART dil modelinin temel eğitim yaklaşımıdır. Bu yöntemde, rastgele bir kelime alt kümesi maskelenir ve eğitimin amacı, doğru bir şekilde maskelenen belirteçleri oluşturmaktır.

BART görevin belirli amacına göre kullanılabilecek MLM'e yönelik diğer yaklaşımlar için bir temel oluşturabilir. BART, belirli bir maskeli token grubunu tahmin etmek için önceden eğitildiğinde, bu belirteçlere daha fazla dikkat eder ve onlara daha duyarlı hale gelir. Bundan dolayı,maskeleme için en uygun belirteçleri belirlemek amacıyla yeni bir yöntem tasarlamak önemlidir. Bu amaçla, maskelemenin doğru bir şekilde tahmin edilmesine ve dolayısıyla her bir belirtecin maskelenip maskelenmeyeceğine karar verilmesine olanak tanıyan bir strateji geliştirilmiştir. Bu yöntem, varlıklar olarak adlandırılan veya cümlelerde diğerlerinden daha büyük anlam taşıyan belirteçlere daha yüksek puanlar atar.

Tez çalışmasında, her bir ifadenin önem düzeyini belirlemek için iki temel metrik kullanılmıştır. Bunlardan ilki ROUGE-1 puanı ilk ölçümüdür ve bir ifadenin ve tüm metnin kelimeler arasındaki örtüşme derecesini belirlemek için kullanılır. İkinci metrik ise bir cümlede bulunan adlandırılmış varlıkların sayısıdır. ROUGE-1'in seçimi, dikkatin bir kısmının özetten oluşan göreve ayrılması ve aynı zamanda belirtilen varlıkların sayısı da dikkate alınarak yapılmıştır. Listelenen öğelerin benzersiz bağlamına daha fazla vurgu yapılarak ve doğrudan sadakatle ilişkilendirilerek metriklerin kullanılması, süreci iyileştirme potansiyeline sahiptir. BART'ın adlandırılmış varlık belirteçleri ile diğer maskelenmiş belirteçler arasında tanımlama yapabilmesini sağlamak için iki tür maske oluşturduk. Maskelerin ilk kategorisi adlandırılmamış varlık belirteçlerini <mask1> içerirken, ikinci kategori adlandırılmış varlık belirteçlerine <mask2> ayrılmıştır.

Ön eğitim aşaması öncelikle maskelenmiş belirteçlerin tahmin edilmesiyle ilgili olduğundan ve alt hedefin özet yapı olması amaçlandığından, ön eğitim ve ince ayar aşamaları arasında hala bir boşluk bulunur. Bundan dolayı, bağlayıcı belirtecini belgenin başına dahil ederek transfer öğrenme sürecinin etkinliğini artırmak için her iki aşamada da giriş belgesine bir bağlayıcı belirteci eklemeyi içeren bağlayıcı stratejisini kullanılmıştır. Geliştirilen yöntem iki soyutlayıcı özetleme veri seti olan XSUM ve ARXIV üzerinde değerlendirilmiş ve iki veri seti üzerinde BART'ın ön eğitimi ile elde edilen modeller BART-XFA ve Bart-AFA olarak adlandırılmıştır.

Deneysel sonuçlar ince ayarlı BART-XFA'nın, QuestEval metriği ile ölçülen, BART-MLM'ye kıyasla tüm deneylerde daha yüksek doğruluk puanları elde ettiğini göstermiştir. Bunun yanında, aslına uygunluktaki bu iyileşmenin özetleme puanı üzerinde olumsuz bir etkisi gözlenmemiştir. BART-XFA ve BART-AFA modelleri, neredeyse tüm özet metriklerde tipik BART-MLM sonuçlarına göre daha olumlu sonuçlar elde edilmiştir.

Adlandırılmış varlık tespit yöntemlerinin güvenilirliğinin, özelleştirilmiş ön eğitim yaklaşımımızla elde edilen sonuçların doğruluğu üzerindeki etkisi incelenmiştir. Soyutlayıcı metin özetlemedeki doğruluk derecesini doğru bir şekilde ölçmek için özelleştirilmiş bir QuestEval metriğini uygulanmıştır. Ayrıca özel maskeleme işlevinin, diğer sadakat ölçütlerini de dahil ederek daha da özelleştirilebileceğini gösterilmiştir. Bu, zaman alıcı yapısı ve metriklerin ölçümlerinin gerektirdiği yoğun hesaplamalar nedeniyle genellikle daha fazla işlem kaynağı gerektirir. Aslına sadık kalma ve özetleme sağlama arasında bir uzlaşma sağlamak için maskeleme tekniği özel fonksiyonumuzda tanımlanan skalerler kullanılarak yapılmıştır. Elde edilen sonuçlara göre, ön eğitim tekniklerinin soyutlayıcı metin özetlemenin doğruluğu üzerindeki önemli etkisini açıkça görülmektedir. Elde edilen sonuçlar, doğal dilin inceliklerinin ve karmaşıklıklarının daha derinlemesine anlaşılmasına yardımcı olarak gelecek çalışmalarda daha güvenilir ve kesin özetleme sistemleri oluşturmasına faydalı olacaktır.

1. INTRODUCTION

Text summarization is the procedure of extracting the fundamental content from a text and creating a compact and cohesive representation that captures the essence of the original while eliminating extraneous elements. Text summarization can boost cognition in an information-rich world. We can better understand complicated issues, retain knowledge, and navigate information-rich surroundings using it.

In contrast to extractive text summarization, which just selects some statements from the original text, abstractive text summarization takes it a step further. Abstractive summarization involves compressing the original material, rephrasing, and generating new sentences using different and distinct words from the source text words in order to convey the core concepts in a concise and coherent way.

Deep learning has led to significant advancements in abstractive text summarization in the recent years. The sequence-to-sequence (seq2seq) models using recurrent neural networks (RNNs) such as LSTMs and GRUs were the traditional way to generate abstractive summaries (Gu et al., 2016; Nallapati et al., 2016; Rush et al., 2015; Xu et al., 2020). However, RNNs still suffer from the problem of grasping the semantic and contextual relationships between words in the input sequences (Rush et al., 2015), as well as their sequential and slow computations.

The advent of Transformer architecture (Vaswani et al., 2017) was a game changer in the Natural Language Processing (NLP) field, especially in text-generative tasks, such as text rephrasing, machine translation, and text summarization. The Transformer architecture lays the groundwork for the emergence of Large Language Models (LLMs) as its dynamic structure allows the model to capture long-range relationships in texts. Recent studies have increasingly relied on LLMs and have achieved significant success in abstractive text summarization, such as in the works, (Durmus et al., 2020; Fischer et al., 2022; Ladhak et al., 2021; Maynez et al., 2020; Scialom et al., 2021).

Nevertheless, recent studies (Anil et al., 2023; Devlin et al., 2018; Lewis et al., 2019; Liu et al., 2019, 2022; Liu & Liu, 2021; Ravaut et al., 2022; J. Zhang et al., 2020)

have conducted both human and automatic evaluations on the generated summaries. These evaluations have revealed that the summaries still suffer from the issue of hallucinations. This means that some crucial statements and entities from the source text are omitted from the summary, while irrelevant and extraneous information is mistakenly included. This highlights the need to emphasize the extent to which the summary remains faithful to the source material.

Prior studies have examined the issue of faithfulness in abstractive text summarization and put forward approaches to improve faithfulness. The primary objective of manifold efforts is to refine the generated summaries and correct any instances of hallucinations that may have occurred. For example, in (S. Chen et al., 2021), a technique was suggested to substitute the hallucinated words and entities in the summary with comparable alternatives. In addition, (Cao & Wang, 2021) employed a contrastive learning approach. Separate studies (Falke et al., 2019; Wan et al., 2023) examined the impact of decoding techniques on faithfulness. Additional studies, such as (X. Chen et al., 2022; Goyal & Durrett, 2021; Xiao & Carenini, 2022; H. Zhang et al., 2022) proposed custom training strategies to improve faithfulness. However, these studies neglect the consideration of the pretraining phase since they mostly rely on pretraining large language models, then fine-tuning the models, or simply post-processing the early fine-tuned versions.

This study focuses on examining the impact of pretraining methods on faithfulness and introduces an innovative pretraining approach to enhance faithfulness. The objective of our pretraining strategy is to incentivize the Bidirectional Autoregressive Transformer (BART) (Lewis et al., 2019) to prioritize tokens and entities that have a stronger correlation with faithfulness. If the model is inclined toward the contextual representations of those tokens, it is more probable to produce summaries that are more faithful. Considering this, we carried out our tests on the BART large language model utilizing two abstractive datasets, namely XSUM (Narayan et al., 2018) and ARXIV (Clement et al., 2019).

1.1. Contributions of the Thesis

This thesis presents its contribution to the research on enhancing faithfulness in text summarization as follows:

- Investigate the impact of pretraining methodologies for BART on faithfulness and summarization.
- Introduce an innovative method for pretraining the BART language model to enhance its faithfulness.
- Conducting experiments to demonstrate that our approach overcomes the traditional MLM in terms of faithfulness metrics. Our proposed BART-XFA model scores 37.09 and 39.00, while the baseline model, BART-MLM scores 35.99 and 38.62 in terms of QuestEval on the ARXIV and XSUM datasets, respectively.
- Study the effect of the quality of Named Entities Recognition tools on summarization and faithfulness.
- Show that our proposed method can even enhance generated summaries in terms of summarization metrics.

1.2. Structure of the Thesis

The aim of this thesis is to comprehensively analyze the notion of faithfulness in abstractive summarization by employing a novel pretraining method. Chapter 2, "Background and Related Works", presents essential ideas, encompassing a comprehensive analysis of deep learning algorithms, text summarization, and evaluation approaches for summarization. Furthermore, we analyze the previous studies regarding abstractive summarization and faithfulness. Chapter 3, titled Methodology, introduces our proposed pretraining technique designed to improve faithfulness in the summary process. The 4th chapter, Experiments, provides a detailed description of our empirical research and practical implementation on ARXIV and XSUM. Subsequently, we engage in a comprehensive examination and careful consideration of the discoveries and outcomes in the "Results" chapter. Ultimately, we wrap up our work and propose future endeavors in the Conclusion and Future Works chapter.

2. BACKGROUND AND RELATED WORKS

2.1. Natural Language Processing

Natural language processing (NLP) is a multidisciplinary domain that integrates linguistics, computer science, and artificial intelligence. The field concentrates on the examination of interactions between computers and human language as well as the creation of computer algorithms that can effectively handle and analyze large amounts of natural language data. The goal is to create a computer that can understand the contents of papers, including the complex contextual nuances of the language used in them. The system is capable of accurately extracting information and insights from the papers, as well as categorizing and organizing the documents.

2.2. Text Summarization

Text summarizing, also known as automatic text summarization, refers to the procedure of generating a concise and cohesive rendition of a longer document. Text summarizing involves extracting crucial information from one or more sources to create a condensed version tailored to a specific user and purpose (M. Zhang et al., 2022).

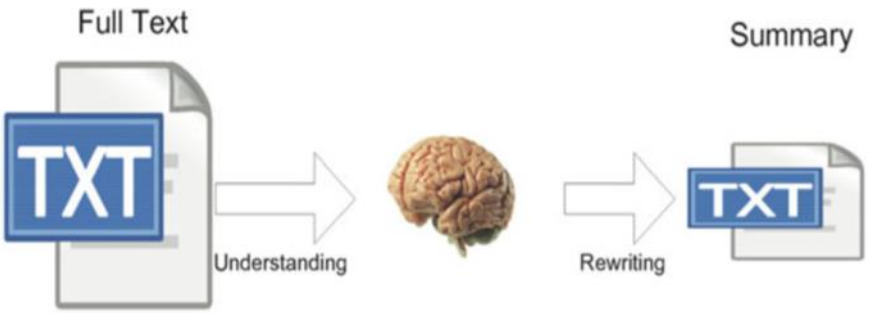

Figure 2.1. Text summarization process.

Humans excel at this particular activity because of their ability to comprehend the significance of the original content and subsequently extract the core meaning while capturing important details in the new description. The goal of automatically creating

text summaries is to create summaries that are on par with human-authored ones in terms of quality.

2.2.1. The need of automatic text summarization

From a broad perspective, it can be stated that text-generated summaries have the potential to enhance the effectiveness of information retrieval and other text mining activities. The need for text summarization is growing rapidly as the volume of text documents on the web continues to expand. There are numerous benefits that come from using automatic text summarizing.

Here are some instances of the advantages:

- Text summarizing decreases the amount of time spent on reading.
- Create a news headline.
- Access significant content from blogs.
- Enhance the process of making decisions.
- Automatic summarizing algorithms exhibit lower bias compared to human summarizers.

2.2.2. Extractive vs. abstractive

Nowadays, the most important classification is extractive and abstractive text summarization. The reason behind that is that extractive text summarization redefines text summarization as the process of selecting phrases and words from the original text and copying them directly into the summary. As a result, a generated summary only includes sentences from the original text. However, abstractive text summarization uses the genuine definition of text summarization, generating a new summary with different words and phrases from the original one. Extractive text summarization seems to be easier to implement than abstractive summarization, but it is also less effective. On the other hand, abstractive text summarization is more complicated but achieves better results.

2.3. Text Representation

The core of NLP revolves around the representation of text, which acts as a connection between unprocessed textual data and machine learning algorithms.

Converting text into a format that is comprehensible to machines is essential for a wide range of natural language processing (NLP) operations.

2.3.1. Bag-of-words (BoW)

This approach displays a document as a compilation of words, disregarding the sequence of words and grammatical connections. Every word is considered a distinct attribute, and the document is depicted as a vector in which each element represents the frequency of a word.

2.3.2. Term frequency-inverse document frequency (TF-IDF)

This approach enhances the Bag of Words (BoW) method by assigning a weight to each word depending on its frequency within the document (Term Frequency, TF) and its scarcity across the full collection of documents (Inverse Document Frequency, IDF). This feature serves to emphasize words that are particularly pertinent to the given document.

2.3.3. N-grams and skip-n-grams

N-grams and skip-n-grams are types of text modulation that extend the text to include not only a list of the words represented by the sequence but also a list of different combinations based on the n factor.

To be more detailed, N-grams are defined as a set of co-occurring words within a given window (N). For example, if we take the next example: “The cow jumps over the moon”. 2-grams representation is represented as follows: [The cow, cow jumps, jumps over, over the, the moon]. While Skip-n-gram is similar to N-grams but with counting the missing words in a given window (n). Returning to previous example, Skip-1-grams is represented as follows: [The jumps, cow over, jumps the, over moon]. The idea of those methods of representation is to use different combinations of words within the sequence and not be restricted only to the given sequence. It is worth mentioning here that many more complicated structures are built based on those simple representations. In Section (2.9.3), we see how ROUGE-N evaluation metrics are defined using those basic structures.

2.3.4. Word embedding

Word embeddings are computational depictions of specific words that encode both their semantic significance and connections to other words within a provided lexicon. These models overcome the limitations of bag-of-words models by considering the semantic and contextual connections between words. Typical word embedding techniques consist of the following:

Word2Vec (Mikolov et al., 2013) is a technique that uses neural networks to obtain vector representations of words. It does this by examining how words are commonly found together in a large collection of written works.

GloVe (Pennington et al., 2014) is a method that employs a statistical approach by examining patterns of word co-occurrence and producing word vectors using information from a global word-word co-occurrence matrix.

Contextual embeddings, such as ELMo (Peters et al., 2017), BERT (Devlin et al., 2018), and RoBERTa (Liu et al., 2019), acquire word representations that are responsive to the surrounding context of the word. This enables them to apprehend the subtleties of significance and emotion that may fluctuate based on the context of the words.

2.3.5. Sentence embedding

Sentence embeddings expand the notion of word embeddings to encompass full sentences. Their objective is to encapsulate the complete significance and emotional tone of a sentence within a singular vector representation. This enables the efficient comparison and analysis of sentences, which is essential for jobs such as:

- Text summaries involve the use of sentence embeddings to discern crucial sentences and provide succinct summaries of longer texts.
- Paraphrase detection: Sentence embeddings have the ability to ascertain if two phrases express the same meaning, irrespective of any variations in language.
- Sentiment analysis: Sentence embeddings provide the capability to assess the sentiment of complete sentences, offering a more intricate comprehension compared to sentiment analysis at the word level.

2.4. Text Preprocessing

One of the most important stages in NLP is called text preprocessing. This process involves cleaning up and transforming unprocessed textual input into a structure that later NLP activities can interpret and assess. Some examples of these tasks include machine learning, information extraction, and text summarization. Getting rid of noise and irregularities in the text, such as punctuation, HTML tags, and special characters, is one of the most important objectives of text preprocessing. Other processes, such as text normalization, dimensionality reduction, tokenization, and so on, may be included in the process of text preparation.

2.4.1. Tokenization

Tokenization, a crucial step in NLP, is the segmentation of a continuous text stream into distinct units referred to as tokens. These tokens have the ability to represent many language components, including words, punctuation marks, and even subword units. Tokenization is an essential step for following NLP operations since it forms the basis for further analysis and processing of textual input.

Tokenization methods can be classified according to the level of detail in the tokens generated:

- Word-Level Tokenization: The prevailing method is dividing the text into separate words using whitespace or punctuation marks as delimiters.
- Character-Level Tokenization: Every individual character in the text is treated as a distinct token, resulting in a detailed representation of the text.
- Sentence Segmentation: The text is segmented into discrete sentences, facilitating the examination of sentence composition and connections.
- N-gram Tokenization: N-grams are obtained by extracting sequences of n consecutive tokens from the text, which allows for the identification of local word order patterns.
- Subword Tokenization: Methods such as byte pair encoding (BPE) or word segmentation algorithms are utilized to divide words into smaller, understandable subword units.

2.4.2. Named entity recognition (NER)

In the domain of NLP, named entity recognition (NER) is a critical element of information extraction. The process entails the identification and categorization of identified entities referenced in unstructured textual data into predetermined categories, such as individuals, corporations, geographical locations, dates, and other pertinent entities. NER is crucial in multiple NLP applications as it allows machines to extract useful information from unstructured text and improve their comprehension of human language.

NER is primarily employed for the purpose of identifying named entities within a text corpus. This involves locating and annotating instances of named entities, which are then classified into preset categories such as "PERSON", "ORGANIZATION", "LOCATION", and so on. NER enables the retrieval of organized data from unorganized language, allowing machines to efficiently process and analyze the retrieved data.

2.5. Neural Networks

Neural networks are a potent category of algorithms that draw inspiration from the intricate structure and functioning of the human brain. They have transformed numerous domains, such as natural language processing (NLP), by empowering machines to acquire intricate connections within data and produce text of high human-like quality.

2.5.1. Multilayer perceptron (MLP)

Multilayer perceptron (MLP), commonly referred to as Feedforward Neural Networks (FNNs), are a fundamental form of artificial neural network design. They serve as the foundation for numerous sophisticated learning algorithms and have had a substantial impact on diverse machine learning applications, such as NLP, pattern recognition, and computer vision.

A FNN consists of interconnected layers of artificial neurons organized in a sequential information flow. Below is an analysis of the essential elements:

- The input layer of a neural network is responsible for receiving the unprocessed data that will be fed into the network for processing. Every neuron in this layer corresponds to a distinct characteristic of the input data.
- Hidden layers: These layers are responsible for the majority of the computational tasks. Every neuron in the network gets inputs from the preceding layer, conducts weighted computations, and produces an output. The complexity and learning capability of the network are determined by the number of hidden layers and neurons in each layer.
- The output layer is responsible for producing the ultimate output of the network, which is determined by the specific task being performed. For NLP tasks, the result can take the form of a classification label (such as the sentiment of a text), a created sentence, or a representation of the input text.

Data is transmitted from the input layer, traverses through the hidden levels, and ultimately reaches the output layer in a single forward propagation. Throughout the training phase, the network modifies the weights of the connections between neurons by taking into account the training data and a selected error function. This approach enhances the network's capacity to precisely forecast or generate desired outcomes. units.

2.5.2. Recurrent neural networks (RNNs)

RNNs are a class of neural networks that can utilize previous outputs as inputs while maintaining internal memory. This allows the network to retain information about the past and use it to make predictions about the future. RNNs are used in a wide variety of NLP tasks as well as in other areas of machine learning, such as time series forecasting, music generation, anomaly detection, etc. Figure 2.2 shows the basic architecture of RNN. The outputs and hidden states can then be computed as follows:

$$a_t = g_a(W_a \,.\, a_{t-1} + W_x . x_t + b_a) \tag{2.1}$$

$$y_t = g_y\left(W_y \,.\, a_t + b_y\right) \tag{2.2}$$

where g_a, g_y are activation functions, W_a, W_x, b_a, b_y are the weights of the RNN

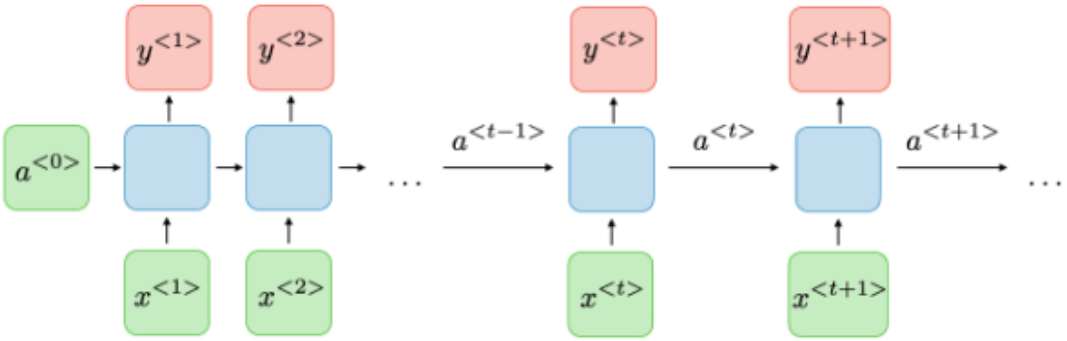

Figure 2.2. Recurrent Neural Network.

In general, RNNs provide the ability to feed inputs of different lengths. The model size is fixed, regardless of the input length. However, computation is sequential. Consequently, it takes more time to compute the results. Furthermore, information from earlier steps seems to be forgotten in later steps. This phenomenon is known as gradient vanishing. In order to solve this problem, more complex RNN architectures were presented, like LSTMs and GRUs.

2.5.3. Long short-term memory (LSTM)

Long short-term memory (LSTM) networks are a specific type of RNN structure that is specifically developed to address the issue of the vanishing gradient problem. This problem is a limitation that regular RNNs have while trying to learn and capture long-range relationships or dependencies in data. LSTM networks are capable of efficiently storing and retrieving information over extended sequences due to their distinctive gating mechanism. This makes them highly suitable for tasks, such as NLP, audio recognition, and time series forecasting. The fundamental architecture of LSTM necessitates the utilization of three primary categories of gates:

- Forget Gate: determines which portion of information from the preceding phases should be discarded.
- Input Gate: is used to assess the significance of fresh information received from the inputs.
- Output Gate: is responsible for determining which components of the state should be included in the output.

2.5.4. Gated recurrent unit (GRU)

While LSTMs effectively address the issue of gradient vanishing, they introduce additional computational complications that result in slower training and prediction times. Gated recurrent units (GRUs) are proposed as a less complex alternative to LSTMs, as demonstrated in Figure 2.3. Thanks to their efficient design, the training and operation of these systems require fewer computational resources, reducing the computational burden and making them better suited for resource-limited situations or real-time applications.

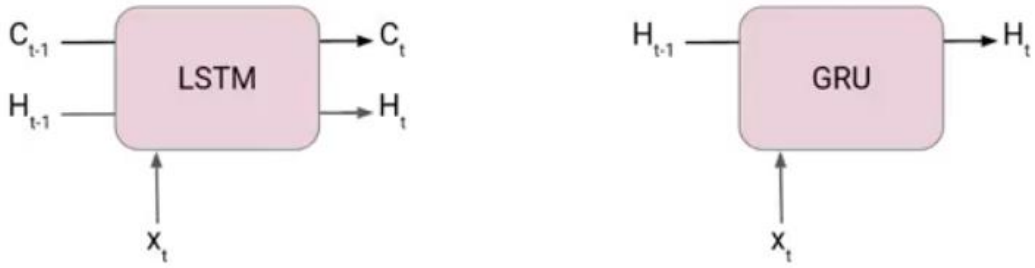

Figure 2.3. LSTM and GRU basic architectures.

Basically, GRU architecture consists of two gates:

- The reset gate is responsible for evaluating the relevance of information from the previous state in order to decide which information should be used to update the current state.
- The update gate is utilized to determine which fresh information from the input will be incorporated into the current state.

2.6. The Traditional Sequence to Sequence Model (Seq2seq)

The traditional sequence-to-sequence model (seq2seq) relies on RNNs, such as LSTMs and GRUs. Seq2seq models primarily comprise three components: word embedding, encoder, and decoder. The architecture is a generative structure composed of neural networks. This implies that the output is not limited to a single class but rather consists of a series of texts that represent the created summary. The Seq2Seq architecture, depicted in Figure 2.4, demonstrates that the input sequence is initially transformed into a semantic representation through the use of an embedding layer. Next, the input is processed by the encoder, which functions as a compressor,

transforming the input sequence into a vector known as the encoder vector. Subsequently, the encoder vector undergoes additional processing by the decoder to generate a token at each time step. This model can be trained using several neural network training algorithms and approaches, such as TensorFlow, Keras, PyTorch, or other compatible tools, utilizing a labeled dataset.

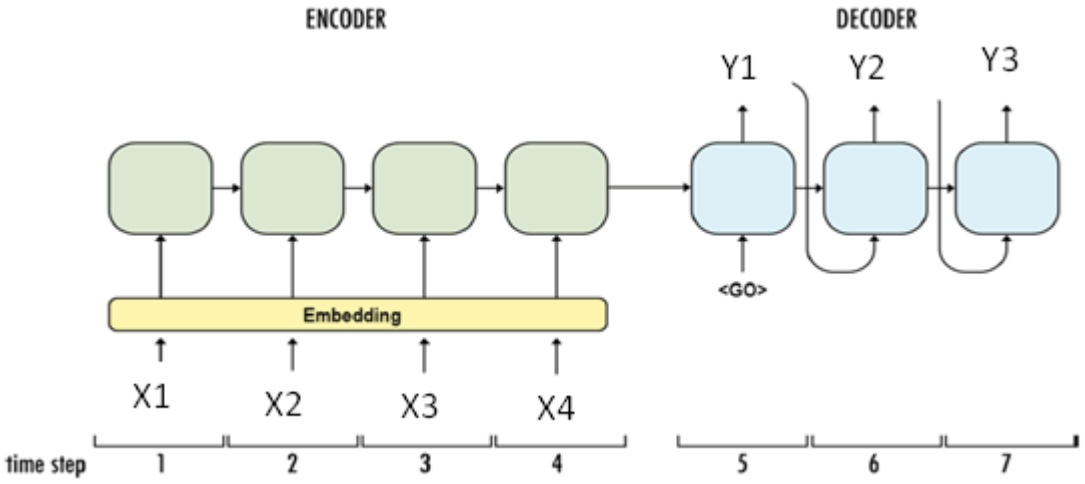

Figure 2.4. The encoder-decoder architecure.

2.6.1. Word embedding layer

Within the domain of NLP, it is a frequent undertaking to handle words and sequences of text. In order for a text to be utilized by deep learning models, it is necessary to translate it into numerical representations. An effective approach to achieving this is by implementing a straightforward tokenizer that assigns a one-hot vector to each word. However, this approach treats two words such as "play" and "plays" as distinct tokens with no inherent relationship between them. Word embedding approaches, such as Word2Vec and Glove, address this issue by discovering a semantic depiction of each word that accurately reflects its meaning. During the word embedding step, the word is encoded as a vector of numerical values that encapsulate a greater amount of semantic information pertaining to the word. For instance, the words "play" and "plays" have approximately comparable meanings.

2.6.2. Encoder

The encoder in a sequence-to-sequence architecture comprises a stack of recurrent neural units, often LSTMs and GRUs. Each unit is provided with a single input from the embedding layer and undergoes processing, resulting in two values: a hidden

state (in the case of LSTM) and an output value (in the case of LSTM and GRU). The output of each unit is transmitted to the adjacent unit together with the subsequent token. Consequently, each token undergoes processing based on its own value and the value of the preceding tokens' processing. The final units produce an encoder vector, which is a compressed form of the input sequence. This vector is then available for processing by the decoder.

2.6.3. Decoder

The decoder consists of a sequence of RNN units, such as LSTM and GRU models. The primary intention of each decoder unit is to utilize the encoder vector and the previously generated token as inputs in order to determine the subsequent token to be generated. To be more precise, the task of the decoder unit is a multi-class classification task where it determines the most suitable class (token) to be chosen for the next output.

2.6.4. Beam search decoding

Beam search decoding is an exploratory approach employed in generative neural networks for NLP tasks. The probabilistic technique seeks to identify the optimal sequence of words or tokens that maximizes the likelihood with respect to specified outputs.

Throughout the decoding process, a beam consisting of the most favorable candidate sequences is retained to aid in beam search decoding. The beam size determines the number of candidate sequences that are kept for consideration at each step. Increasing the size of the beam results in higher computational complexity, but it also allows for more extensive exploration of different paths, potentially leading to improved results.

The algorithm extends the beam iteratively by analyzing alternative extensions for each sequence in the current beam and selecting the top k most probable sequences to build the new beam for the next phase. This procedure continues until a termination condition is met, such as reaching a maximum length or encountering a stop symbol.

2.6.5. The problem of seq2seq models

The encoder vector generated during the encoding phase exhibits a bias towards the latter segments of the sequence, particularly in the case of longer sequences where the first sections of the sequence have a lesser influence on the semantic representation of the encoder vector. This outcome is a consequence of the structure of the encoding step. For a lengthy sequence, each encoder unit receives information at each step and passes it on to the next unit, resulting in a diminishing impact of earlier inputs on the final outputs. Consequently, the latter sections of the sequence are perceived as more influential.

Another issue with the encoder in seq2seq models is the underutilization of the hidden states of each encoder unit in the encoder vector. An attention mechanism was introduced to address these concerns.

2.6.6. Attention mechanism

(Rush et al., 2015) suggested the attention mechanism as a solution to the difficulty of the encoder-decoder model discussed in Section 2.6.2. The attention mechanism in a deep learning model translates the hidden states and output of the encoder units to the decoder, producing a context vector and a collection of attention weights. These weights can be learned throughout the model's learning process. In conjunction with the encoder vector, the decoder receives the context vector for the purpose of generating the output sequence.

From a deep learning standpoint, the attention mechanism serves to prioritize and concentrate on particular segments of the input data throughout the learning process. This behavior appears to be particularly effective in deep learning generating tasks and enhances the accuracy of the predictions.

Another form of attention is self-attention, where the input sequence is partitioned into segments (h1, h2, h3,..., hn). When the word "wi" is found within the segment "hj", the attention graph exclusively emphasizes the "j" portion.

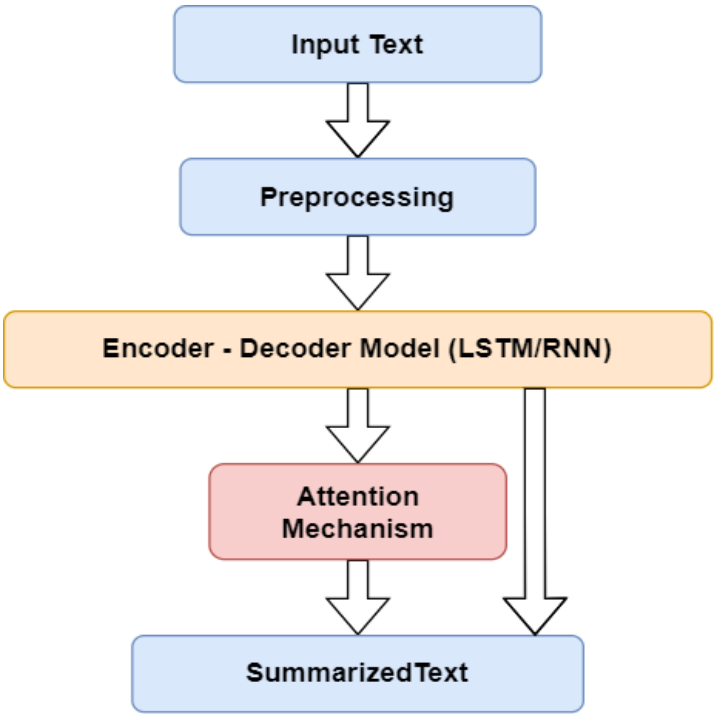

Figure 2.5. Text summarization architeture using the encoder-decoder model with the attention mechanism from (Rush et al., 2015).

2.6.7. The copy mechanism

The copy mechanism was initially introduced in 2016 by (Gu et al., 2016) to address the issue of out-of-vocabulary (OOV) tokens. OOV stands for out of vocabulary words, which refers to a set of words that do not exist in the vocabulary dictionary during the tokenizing process. The copy method is responsible for directly transferring certain terms from the source document to the generated summary without any changes. Consequently, OOV words can now be effectively reflected in the generated summary.

In 2020, the use of the copy technique was further enhanced by implementing an attention graph structure to evaluate the process of word prioritization. The self-attention graph (Xu et al., 2020) provided is constructed based on the semantic relationships among words. Centrality metrics can be used to compute the priority of a single word. The centralities are incorporated into the copy procedure to further refine the accuracy of the final model.

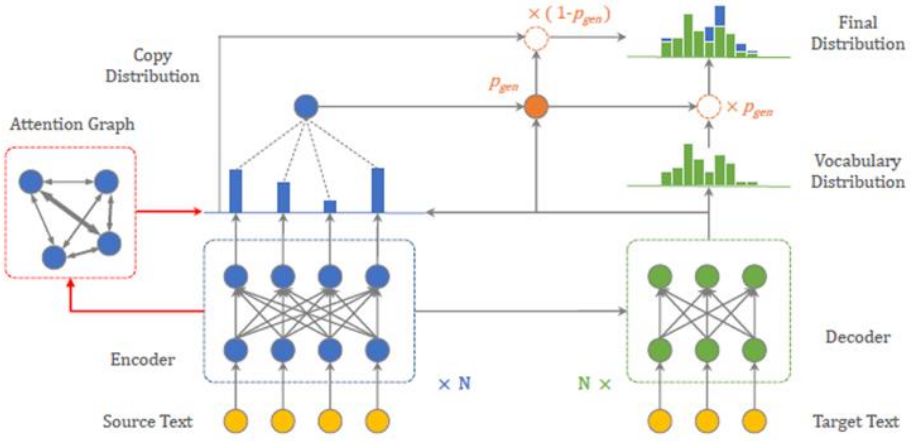

Figure 2.6. The copy mechanism flow from (Xu et al., 2020).

2.7. The Trasformer Architecture

The discipline of NLP has experienced a significant transformation due to the revolutionary architecture called the Transformer model. The Transformer model, initially proposed in the work "Attention is All You Need" by (Vaswani et al., 2017), revolutionized NLP by exclusively relying on an attention mechanism. This eliminated the necessity of RNNs and the associated challenges they entail.

The attention mechanism is the main breakthrough of the Transformer model. By utilizing the attention mechanism, the model is capable of concurrently evaluating every element of the input sequence, effectively capturing long-range associations better than RNNs, which study sequences in a sequential manner.

2.7.1. Scaled dot-product attention

The scaled dot-product attention function is a mathematical process that takes a query and a set of key-value pairs as input and produces an output. The mapping is depicted using vectors to represent the query, keys, values, and output. The input consists of queries and keys, both with a specified dimension, as well as values, also with a specified dimension. The weights on the values are determined by computing the dot products of the query with all keys, dividing them by the square root of each, and applying a softmax function as shown in equation 2.3:

$$\text{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \text{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{V} \qquad (2.3)$$

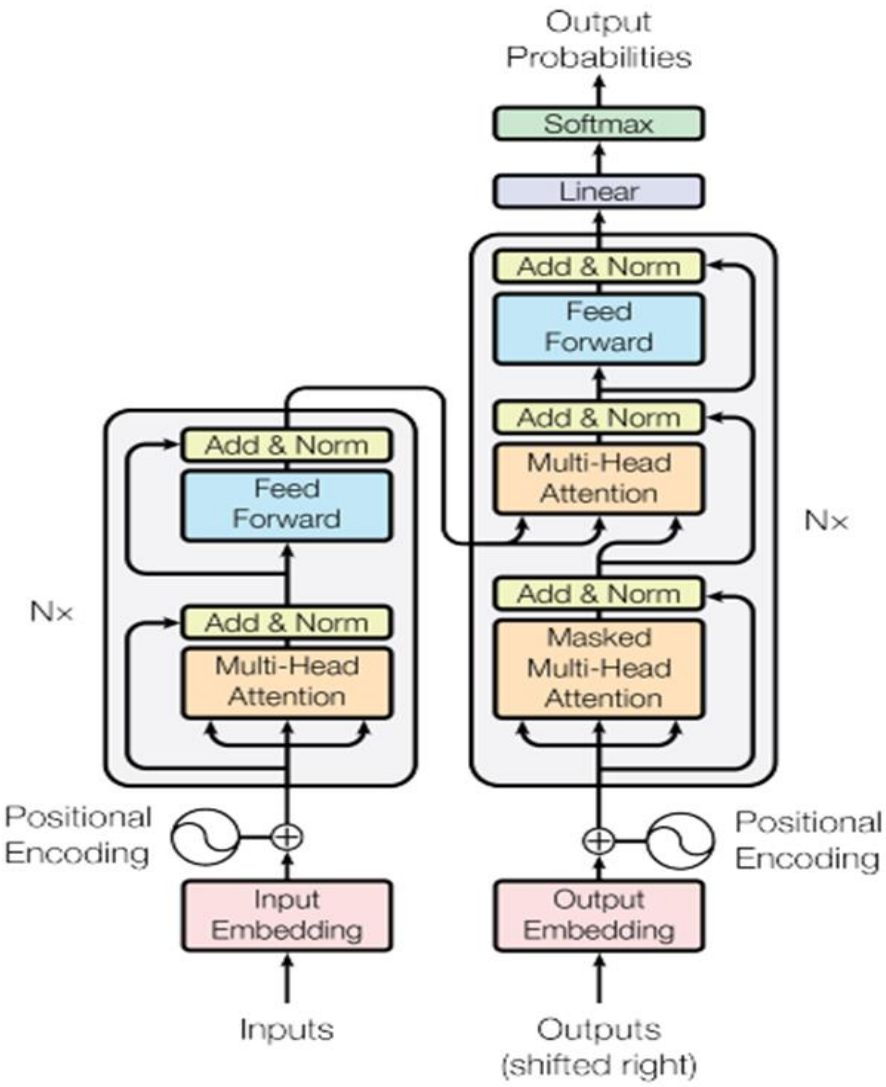

Figure 2.7. The transformer structure from (Vaswani et al., 2017).

2.7.2. Multi-head attention

The fundamental attention mechanism, also known as single-head attention, seeks to capture all aspects of connections among input tokens. However, in the context of genuine language, several types of connections, such as grammatical links or semantic relationships, may require separate representations.

Multi-head attention solves this limitation by utilizing many attention mechanisms concurrently, each focusing on a specific type of interaction. This enables the model to understand the input sequence with increased complexity and nuance. Figure 2.8 depicts the primary framework of multi-head and scaled dot-product flow.

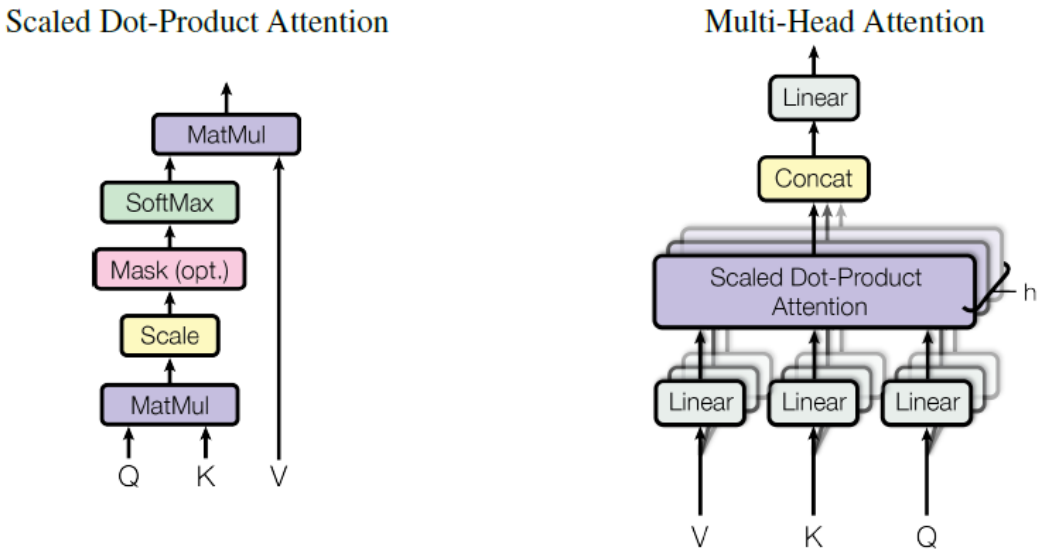

Figure 2.8. Scaled dot-product attention and multi-head attention from (Vaswani et al., 2017).

2.7.3. Positional encoding

While RNNs inherently preserve positional information by successively analyzing sequences, the Transformer model relies on an attention mechanism that simultaneously considers all elements of the input sequence. Without positional encoding, the Transformer is unable to distinguish between different positions in the sequence, leading to inaccurate representations and inferior performance.

Sinusoidal functions are commonly employed in positional encoding to accurately depict the periodic nature of language. By allocating unique vectors to different positions in the sequence, these functions offer the model a better understanding of the relative arrangement of the tokens.

The model incorporates positional information into its token representation by augmenting the input embeddings with positional encoding vectors. By doing this, the Transformer's ability to understand the context of the sequence and the relationships between its words and tokens is guaranteed.

2.8. Large Language Models (LLMs)

Large language models (LLMs), which are classified as a kind of artificial intelligence (AI) methods, are developed through rigorous training using vast quantities of textual data in order to obtain intricate language representations and

patterns (Zhao et al., 2023). These impressive models possess the ability to generate writing that is comparable to human quality, carry out language translation, create various types of creative content, and offer useful solutions to your queries. LLMs have emerged as a groundbreaking force in the field of NLP, leading to significant advancements in many tasks and introducing a novel method for handling and generating text.

By nature of their training on enormous quantities of textual data, LLMs are capable of acquiring knowledge of intricate linguistic structures and patterns. The Transformer architecture is the fundamental framework used for the processing and creation of text in LLMs. The model's ability to handle long sequences and its efficient attention mechanism make it well-suited for the demanding training and inference requirements of LLMs.

2.8.1. Masked language modeling (MLM)

Within language modeling, MLM is a widely used technique in the pre-training of LLMs for NLP applications (Zhao et al., 2023). It involves training the models to predict missing words in a given text. MLM has become an essential component of LLM pre-training, enabling these models to gain strong language representations and achieve impressive performance on various NLP tasks.

MLM involves randomly obscuring a specific percentage of tokens inside an input phrase and then teaching the LLM to anticipate the obscured words by taking into account the context of the remaining words. This strategy stimulates the model to reflect on the relationships between words and cultivate a deep understanding of linguistic patterns.

2.8.2. Transfer learning

Transfer learning is a technique in machine learning that utilizes the knowledge and skills of a pre-trained model for a certain task and applies them to a different task or domain. This approach is particularly advantageous when the new task lacks sufficient training data or when it shares similarities with the original task.

LLMs might employ transfer learning to leverage the vast knowledge and skills acquired by training on a huge dataset. This enables them to address novel problems or improve performance on tasks that require a smaller amount of data.

Transfer learning involves the practice of fine-tuning, which entails training a pre-trained LLM using a smaller dataset that is specifically relevant to the current task. Through this additional training, the model can focus on the nuanced intricacies and repetitive patterns of the specific action, thereby improving its skill in performing the intended task.

2.8.3. Bidirectional encoder representations from transformers (BERT)

In 2018, Google unveiled BERT, a sophisticated language model referred to as bidirectional encoder representations from transformers. (Devlin et al., 2018). BERT, in contrast to traditional LLMs, uses the Transformer architecture to examine text bidirectionally, facilitating a more holistic comprehension of the contextual relationships among words. The BERT model is built using a sequence of transformer layers, with 12 layers for the base-BERT variant and 24 layers for the large-BERT variant.

BERT undergoes pre-training using a comprehensive dataset that encompasses a substantial volume of text, incorporating both BooksCorpus and Wikipedia. This exposes the model to a diverse array of language patterns and prepares it for subsequent tasks. Subsequently, the model undergoes finetuning on a downstream dataset, as illustrated in Figure 2.9.

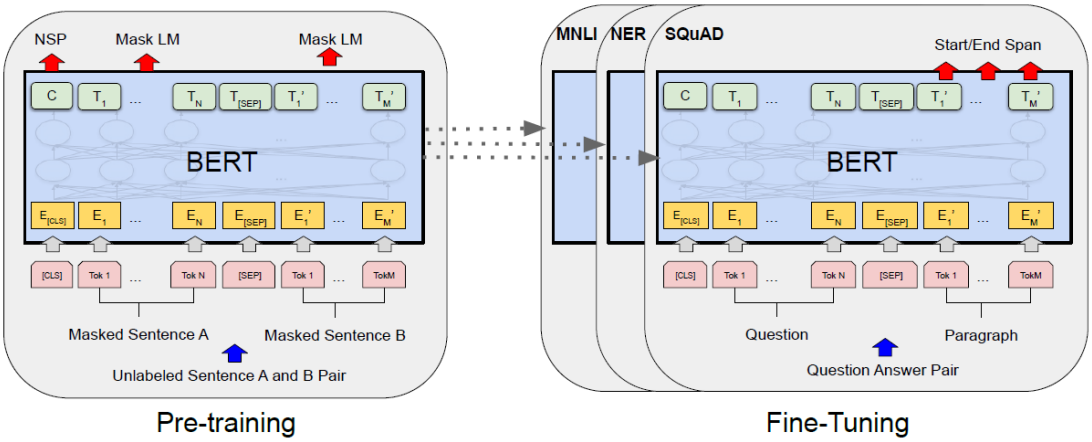

Figure 2.9. BERT pretraining and finetuning phases from (Devlin et al., 2018).

BERT was subjected to pretraining utilizing the MLM and Next Sentence Prediction (NSP) approaches, wherein the model was trained to determine whether two phrases appear consecutively within a document. This aids the model in understanding the flow and coherence of phrases, which is crucial for tasks such as question answering.

2.8.4. Robustly optimized BERT pretraining approach (RoBERTa)

RoBERTa (Liu et al., 2019) is a highly powerful and extensive language model that is constructed based on the BERT architecture. It is essentially an improved version of BERT, designed to surpass certain limitations and achieve higher performance. The subsequent elements delineate the enhanced training methodologies:

- RoBERTa has longer training sessions than BERT, allowing it to capture a higher degree of subtle language nuances.
- Utilizing greater quantities: By augmenting the data batch size employed for training, the model is exposed to a broader spectrum of scenarios, resulting in the creation of more resilient representations.
- RoBERTa utilizes dynamic masking, a technique that involves modifying the masking pattern rather than employing a fixed proportion of words in each sequence. This methodology hinders the model's ability to acquire basic patterns.
- The RoBERTa model eliminates the NSP component seen in BERT because it is not necessary for downstream tasks.
- RoBERTa employs byte pair encoding, a technique that utilizes bytes as the basic unit for subword tokenization instead of characters. RoBERTa is able to efficiently handle a wider range of languages and unusual characters.
- RoBERTa possesses a more expansive lexicon compared to BERT, allowing it to encompass a broader spectrum of diverse and exact terminology.

RoBERTa consistently outperforms BERT in several benchmarks, particularly in question answering, summarization, and natural language inference. Furthermore, it provides a stronger and more adaptable portrayal of language, hence enhancing its versatility in enhancing performance on various NLP tasks.

2.8.5. Bidirectional and autoregressive transformer (BART)

The Bidirectional and Autoregressive Transformer (BART) (Lewis et al., 2019) is a LLM that combines the strengths of bidirectional and autoregressive transformers, resulting in a highly efficient model suitable for various applications.

As described in figure 2.10, the model is composed of a total of 24 layers, divided evenly between the encoder and the decoder.

Similar to BERT, the bidirectional component allows BART to process text in both directions, capturing deeper contextual relationships between words, while the autoregressive component allows BART to generate text one word at a time, taking into account the words that have already been generated. This makes BART well-suited for tasks such as machine translation and summarization, where it is important to generate fluent and coherent text.

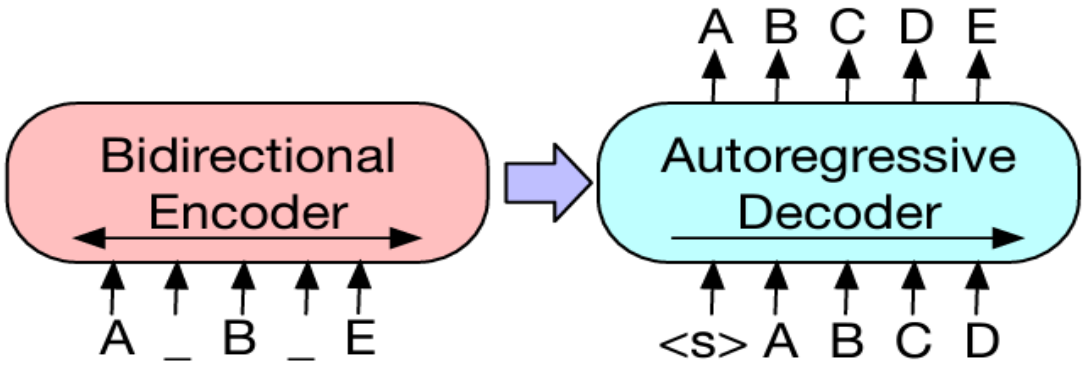

Figure 2.10. BART bidirectional encoder and autoregressive decoder architecture from (Lewis et al., 2019).

BART is trained on a giant dataset of text by deliberately introducing errors into texts and then optimizing an error loss, which is measured by the cross-entropy between the decoder's output and the original document. Unlike other denoising autoencoders that are designed for certain types of noise, BART enables us to apply any kind of document degradation. If all information about the source is completely lost, BART might be seen as being identical to a language model. BART depends on various noising strategies, including token masking and text infilling. For finetuning, BART can be easily finetuned on generative downstream tasks as the architecture includes the autoregressive decoder.

2.9. Evaluation Metrics

2.9.1. Precision and recall

Precision and recall are accuracy metrics that evaluate the similarity between a created instance and a reference instance. Precision, as described by equation 2.4, is the proportion of pertinent instances to the total number of instances retrieved. Recall, as denoted by equation 2.5, is the proportion of relevant instances among the

reference ones. The F1-measure combines the two precisions using the formula introduced in equation 2.6.

$$\mathrm{P} = \frac{\text{overlab between reference and system summary}}{\text{Total number of words ofsystem summary}} \tag{2.4}$$

$$\mathrm{R} = \frac{\text{overlab between reference and system summary}}{\text{Total number of words of reference summary}} \tag{2.5}$$

$$\mathrm{F1} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.6}$$

2.9.2. Recall oriented understudy for gisting evaluation (ROUGE)

ROUGE (Lin, 2004) is a software program and a collection of metrics utilized in NLP to assess the effectiveness of automated summarization and machine translation technologies. ROUGE metrics are employed to compare a generated summary or translation with a reference or a set of references, which are summaries or translations authored by humans.

There are several ROUGE measures, with the frequently employed ones being ROUGE-N, ROUGE-L, and ROUGE-S.

The degree of closeness between the predicted summary and the source summary is assessed by ROUGE-N through the evaluation of the intersection of n-grams. A higher value of N corresponds to a more accurate match. As an illustration, ROUGE-1 quantifies the degree of similarity among unigrams (individual words), ROUGE-2 quantifies the degree of similarity between bigrams (pairings of words), and so forth.

The formula for calculating ROUGE-N is as follows:

$$\mathrm{ROUGE_N} = \frac{\sum_{s \,\epsilon\, \{\text{reference summaries}\}} \sum_{\mathrm{gram_n} \,\epsilon\, s} \mathrm{Count_{match}(gram_n)}}{\sum_{s \,\epsilon\, \{\text{reference summaries}\}} \sum_{\mathrm{gram_n} \,\epsilon\, s} \mathrm{Count(gram_n)}} \tag{2.7}$$

The length of the longest common subsequence (LCS) of terms that appear in both the generated summary and the reference summary is quantified by ROUGE-L. LCS refers to the word sequence in both documents that is the longest and occurs in the same order.

ROUGE-L can be computed by the following formula:

$$\mathrm{R_{LCS}} = \frac{\mathrm{LCS}(\text{system summary},\ \text{reference summary})}{\text{Total number of words of reference summary}} \tag{2.8}$$

ROUGE-S measures the level of overlap between the predicted summary and the ground truth summary using skip-gram co-occurrence. Skip-gram concurrence measures the frequency at which words that appear together in the reference summary also appear together in the produced summary, without considering their order.

Equation 2.9 is used to find the ROUGE-S value:

$$R_{SkipN} = \frac{\text{SkipN(system summary, reference summary)}}{\text{Total number of skipN grams of reference summary}} \quad (2.9)$$

Commonly, ROUGE metrics are computed as F-scores, which are the product of accuracy and recall. Recall is the metric by which the percentage of words in the reference summary that are also shown in the generated summary is quantified, whereas precision quantifies the percentage of phrases in the generated summary that are shown as well in the reference summary. In order to determine the F-score, the harmonic mean of accuracy and recall is utilized.

ROUGE measures are highly effective for evaluating the efficiency of automated summarization and machine translation systems. Researchers and developers utilize them to perform comparative studies of different systems and track the advancement of systems over a specific period of time.

2.9.3. BERTScore

BERTScore (T. Zhang et al., 2019) is an automated evaluation metric used to quantify the resemblance between a created text and a reference text in tasks related to text production. It primarily depends on the BERT model. While both BERT and RoBERTa are language models that can be used for BERTScore evaluation, RoBERTa is generally considered to be the superior model for this task.

BERTScore uses the model to compute a similarity score for each token (word or subword) in the generated text relative to each token in the reference text. The similarity evaluations are subsequently averaged to obtain a definitive BERTScore.

BERTScore utilizes the final layer of the pre-trained language model as its default setting, regardless of whether the model is BERT or RoBERTa. This is because the final layer is perceived as the one that captures the most complex and meaningful representations of the input text.

BERTScore has numerous advantages in comparison to other automated evaluation metrics, such as ROUGE. Here are other examples that illustrate these advantages:

- BERTScore demonstrates a stronger association with human evaluation of text quality in comparison to other metrics.
- BERTScore has heightened sensitivity towards fluency, allowing it to precisely evaluate the fluency of a given text. This property is essential for evaluating activities associated with text generation.
- BERTScore demonstrates more robustness against noise: it is less susceptible to disruptions in the reference text, such as typographical errors and grammatical faults.

BERTScore is an indispensable tool for evaluating the quality of text-generation algorithms. Researchers and engineers utilize it to compare different systems and track system performance over time.

2.9.4. QuestEval

The QuestEval (Scialom et al., 2021) methodology assesses summarization systems by considering the factual coherence and relevancy of the output text without the need for any human reference. QuestEval comprises a question-generation module and a question-answering module, as seen in Figure 2.11.

The input document undergoes preprocessing in the question-creation component to extract the named entities, followed by the development of a collection of potential questions. The purpose of these questions is to encompass different facets of the information, encompassing factual specifics (questions pertaining to who, what, where, when, and how), connections between entities such as cause-and-effect, comparison, and association questions, and fundamental concepts and reasoning (questions requiring explanation, justification, and analysis of how and why).

Subsequently, the component identifies the most informative questions and rephrases them using an LLM like the T5 model (Kale & Rastogi, 2020).

In the question-answering step, the questions formed in the preceding phase are inputted into the QA model together with the source text and the output text. Subsequently, the QA model independently analyzes each question and candidate text combination. The system examines the intention and significance of the inquiry

and retrieves pertinent information from the given text. The QA model utilizes sophisticated methods such as semantic comprehension, context-aware reasoning, and machine reading comprehension to accurately anticipate the most suitable response to a given question, drawing from the presented text. Subsequently, the answer may be taken directly from the text or formed as a natural language response, contingent upon the QA model used. Certain QA models additionally include a confidence score that indicates the level of certainty associated with the anticipated response. Supplementary authentication methods may be used to guarantee that the obtained response precisely corresponds to the information contained in the text.

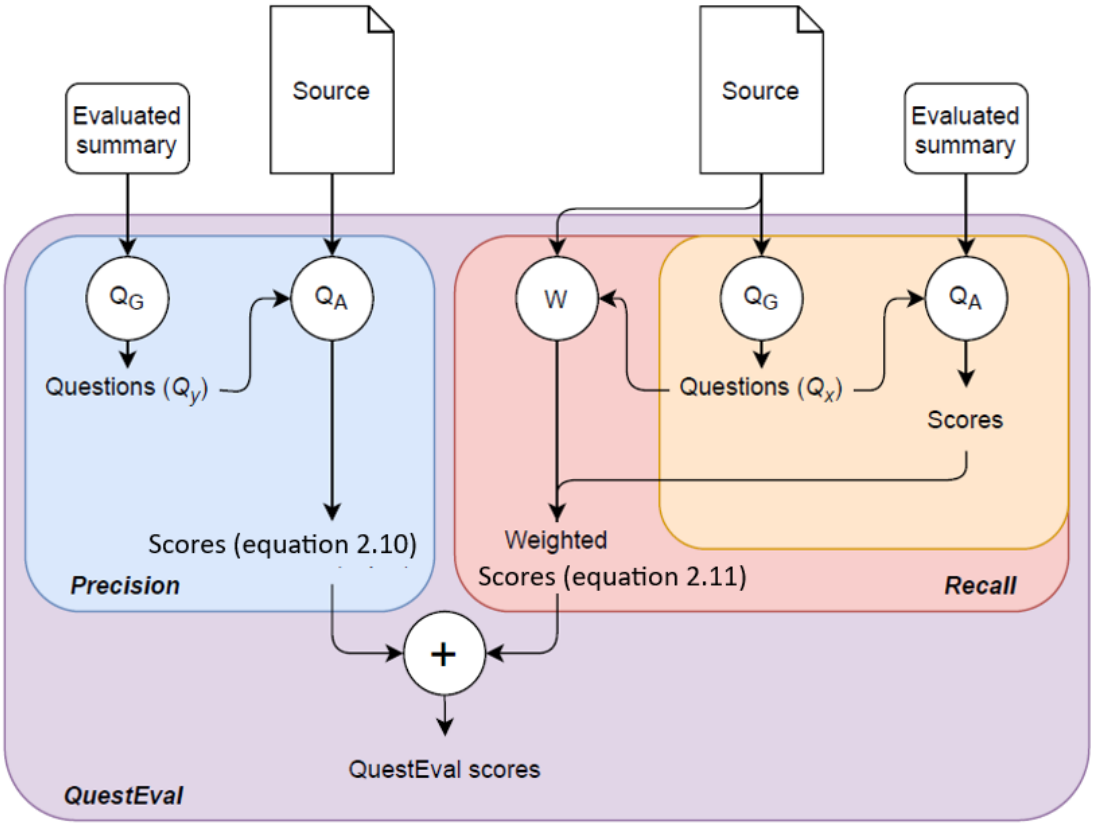

Figure 2.11. QuestEval framework from (Scialom et al., 2021).

Then, as illustrated in Figure 2.11, we compute the precision and recall using equation 2.10 and equation 2.11, respectively.

$$\text{Precision}(\mathrm{D},\mathrm{S}) = \frac{1}{|\mathrm{Q_G}(\mathrm{S})|} \sum_{(\mathrm{q},\mathrm{r})\in \mathrm{Q_G}(\mathrm{S})} \mathrm{F1}(\mathrm{Q_A}(\mathrm{D},\mathrm{q}),\mathrm{r}) \qquad (2.10)$$

where D is a source document and S is the related summary, $\mathrm{Q_G}(\mathrm{S})$ is the set of all questions and answers pairs of the summary, r is the ground truth answer, $\mathrm{Q_A}(D,q)$

is the generated answer of q from D, and F1 is the F1 score to measure the overlap between the predicted answer and the corresponding ground truth (Scialom et al., 2021).

$$\text{Recall}(\text{D}, \text{S}) = \frac{\sum_{(\text{q,r})\in \text{Q}_\text{G}(\text{D})} \text{W}(\text{q}, \text{D})(1 - \text{Q}_\text{A}(\epsilon|\text{S}, \text{q}))}{\sum_{(\text{q,r})\in \text{Q}_\text{G}(\text{D})} \text{W}(\text{q}, \text{D})} \tag{2.11}$$

where D is the document, S is the summary, $\text{Q}_\text{G}(\text{D})$ is the set of all generated question-answer pairs of the document. W(q, D) is the weight of q in D, and $\text{Q}_\text{A}(D, q)$ is the generated answer of q from D.

Upon creating and responding to the questions, the anticipated replies are compared for similarity, taking into account both the original text and the created text. QuestEval employs several similarity measures, such as ROUGE-L, F1 score, and word-level matching, to measure the extent of overlap between the two replies. The scores acquired for each produced question are combined to get a total QuestEval score. One may do this by using techniques such as averaging or weighted averaging, which rely on the given significance of certain issues. The final score evaluates the comprehensiveness and precision of the produced text in relation to the original material.

2.10. Related Works

Several studies (Fischer et al., 2022; Maynez et al., 2020) have shown that summaries produced by LLMs exhibit issues of hallucination and lack of faithfulness when compared to the source documents. Several investigations have been conducted to explore the issue of faithfulness in abstractive text summarization. Those works can be classified into three primary categories: post-process approaches, faithfulness-aware generating methods, and custom training methods. Two examples of post-processing methods include the study by (S. Chen et al., 2021) and the work by (Cao & Wang, 2021). The proposed approach, as recommended by (S. Chen et al., 2021), intends to rectify the hallucinated entities that arise during summary generation. This is achieved by identifying those entities and subsequently replacing them with more accurate named entities. Nevertheless, this approach resulted in incoherent texts. (Cao & Wang, 2021) utilized a contrastive learning approach to implement an alternative post-process strategy. On the other hand, faithfulness-aware generation

strategies concentrate on enhancing faithfulness during the generation stage of the summary process. (Falke et al., 2019) utilized beam search to create many summarization candidates and subsequently choose the most faithful summary based on a single metric. Alternatively, (Wan et al., 2023) propose a faithfulness-aware decoding strategy that incorporates a specific lookahead technique based on multiple faithfulness metrics. Other studies proposed customized training approaches to enhance faithfulness. For example, (X. Chen et al., 2022) enhance the training process by introducing a multi-task framework that incorporates a customized loss function. This framework relies on a question-answering-aware decoder and a multi-task encoder. Furthermore, (H. Zhang et al., 2022) propose the utilization of entity coverage encoding during the training phase by incorporating customized control codes into the training dataset samples. While the previously mentioned studies investigate faithfulness through various methods, they neglect opportunities to refine faithfulness during the crucial pre-training stage.

From another point of view, many studies utilized custom pretraining objectives for generation tasks. For instance, (Wan & Bansal, 2022) introduced a custom factuality-aware pretraining strategy in order to enhance factuality in abstractive text summarization. While (Berezin & Batura, 2023) suggested a customized pretraining method to enhance named entity inclusion in abstractive summaries.

Table 2.1. Literature review summary.

Model	Objective	Category	Strategy
BART (Lewis et al., 2019)	MLM	Pretraining	BART-MLM
BART+correct (S. Chen et al., 2021)	Faithfulness	Post-process	Contrast candidate selection
CLIFF (Cao & Wang, 2021)	Faithfulness	Post-process	Contrastive learning
NLI (Falke et al., 2019)	Faithfulness	Decoding	Candidates ranking

Table 2.1. (Continued) Literature review summary.

Model	Objectuve	Category	Strategy
Lookahead +Ranking (Wan et al., 2023)	Faithfulness	Decoding	A custom beam search strategy that can lookahead and rank based on faithfulness metrics
FES (X. Chen et al., 2022)	Faithfulness	Training	Multi-task framework that depends on QA
ECC (H. Zhang et al., 2022)	Faithfulness	Training	Entity coverage encoding during the training phase
FactPEGASUS (Wan & Bansal, 2022)	Factuality	Pretraining	Pretraining strategy to enhance factuality
MNELM (Berezin & Batura, 2023)	NER Inclusion	Pretraining	Pretraining strategy to enhance NER inclusion

3. METHODOLOGY

There are three distinguishable steps that make up the pretraining process. As shown in Figure 3.1, the first step that we take is to preprocess the pretraining dataset in order to provide a rating to each sentence that is contained within the source documents. Following this, we selectively masked the tokens in line with the importance that has been allocated to each sentence. Sentences that have a higher priority have a greater number of masked tokens in comparison to sentences that have a lower priority. Furthermore, in order to enhance the process of fine-tuning, we are reliant on the technology that facilitates connections.

3.1. Faithfulness-Aware MLM

Through the implementation of our plan, we intend to direct LLM models to give more attention to entities and tokens that have a favorable impact on faithfulness. MLM is the foundation of the fundamental training approach for the BART language model. In this method, a random subset of words is masked off, and the objective of training is to accurately generate the tokens that have been masked out. The original work done by BART can serve as a foundation for a variety of other approaches to masked language modeling (MLM), which can be utilized based on the particular objective of the task. When BART is trained in advance to predict a particular group of masked tokens, it is anticipated that it will pay more attention to those tokens and become more responsive to them (Berezin & Batura, 2023; Lewis et al., 2019). With this in mind, our challenge may be clearly described as an MLM that is customized for faithfulness. As a result, it is essential to devise a heuristic in order to ascertain the tokens that are the most appropriate for masking.

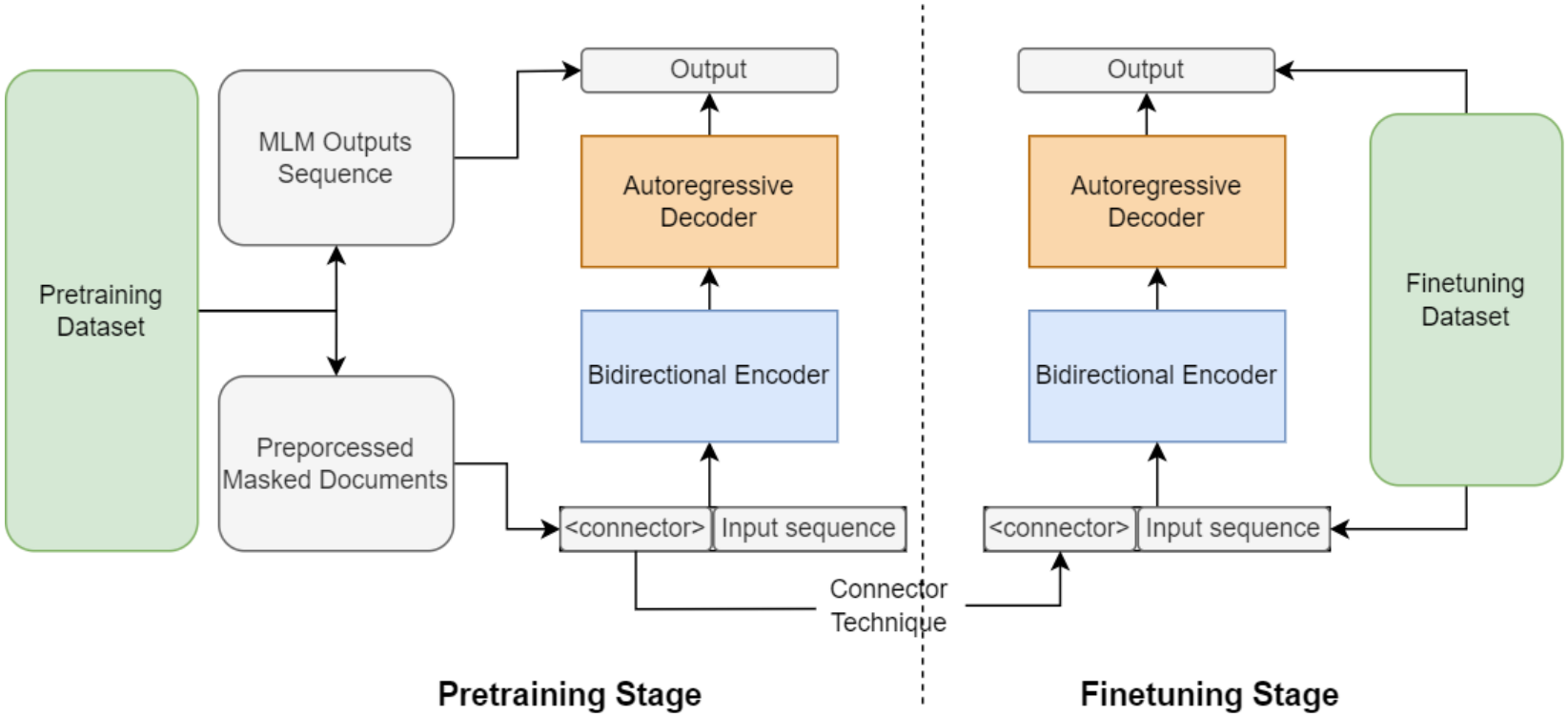

Figure 3.1. Our faithfulness-aware pretraining strategy. In the pretraining stage, the dataset is preprocessed to extract occurred named entities and rank the sentences using our custom ranking, depending on ROUGE-1 and |NEs|. Then our custom MLM pretraining is conducted using the BART, that consists of a bidirectional encoder and an autoregressive decoder, by masking our tokens based on our custom masking probabilities. Then we use the connector strategy to link between pretraining and finetuning stages.

3.2. Sentence Ranking

For the purpose of determining whether or not each token ought to be masked, we devise a heuristic that enables us to make an accurate prediction regarding the opportunity for masking. The heuristic gives greater marks to tokens that are named entities or tokens that appear in sentences that are more meaningful than others.

Two fundamental criteria are what we rely on in order to determine the level of significance that each statement possesses. The ROUGE-1 score is the first metric, and it is used to determine the degree to which a phrase and the entire text overlap between words, while the second measured metric is the count of named entities that are present in a sentence. The selection of ROUGE-1 was made with the intention of devoting a portion of the attention to the task consisting of the summary while also taking into consideration the number of specified entities.

By focusing more emphasis on the unique context of the elements that are listed and are directly related to faithfulness, the utilization of metrics has the potential to improve the process. We have built two kinds of masks in order to enable BART to identify between named entity tokens and other masked tokens. The first category of masks involves non-named entity tokens <mask1>, while the second category is dedicated to named entity tokens <mask2>.

Let D be a document comprising n sentences, denoted as D = {x1 , x2 , ..., xn}. Let α and β be scalars, and |NEi| represent the count of named entities in the sentence xi. The equation 3.1 is used to determine the significance of a sentence in a formal way:

$$S_i = \alpha \times \mathrm{ROUGE}(x_i, D|x_i) + \beta \times |NE_i| \quad (3.1)$$

The masking probability of a non-named entity token tj in a phrase xi is formally defined as:

$$P_i = \lambda_1 \times S_i \quad (3.2)$$

where λ_1 is a scalar.

While the masking probability of a named entity token is given by equation 3.3.

$$P_{NE} = \lambda_2 \quad (3.3)$$

where λ_2 is a scalar and it is explicitly specified as a value between 0 and 1 for named entity tokens, regardless of the significance of the sentence in question.

3.3. Connector

As a result of the fact that the pretraining stage is primarily concerned with predicting masked tokens and the downstream objective is intended to be the construction of the summary, there is still a gap between the pretraining and fine-tuning stages (Wan & Bansal, 2022). We use the connector strategy, which involves inserting a connector token into the input document at both stages, in order to bolster the effectiveness of the transfer learning process. By incorporating the connector token at the beginning of the document, we followed the methodology described in (Wan & Bansal, 2022).

4. EXPERIMENTS

4.1. Datasets

Our experiments are carried out on two different datasets for the purpose of this study. Specifically, the ARXIV (Clement et al., 2019) and XSUM (Narayan et al., 2018) databases are the ones in question.

4.1.1. The XSUM dataset

The XSUM dataset (Narayan et al., 2018) is a significant example of text summarization, demonstrating the impact of extensive data and abstractive methods. XSUM compels models to surpass the inclination of just repeating facts by requiring brief yet thorough descriptions. It forces readers to confront the fundamental nature of the text, reducing the author's purpose, emotional subtleties, and complex network of implicit relationships into a powerful, concise summary in one sentence.

The dataset comprises a vast compilation of 226,711 news articles, each meticulously paired with a succinct summary created by humans. Specifically, it consists of 204,000 training samples, 11,300 validation samples, and 11,300 test samples.

XSUM promotes the growth of models that understand the implicit, going beyond mere identification of crucial things and events. Users engage in a thorough analysis, deducing underlying significance, recognizing the subtle connections between different pieces of information, and perceiving the author's unstated purpose and emotional nuances. Their capacity to decipher implicit meaning enables kids to grasp the fundamental nature of the story rather than merely the basic facts. XSUM also promotes the reimagining of information, allowing models to change viewpoints, simplify intricate arguments into easily understandable fragments, and add a hint of imaginative style while staying true to the fundamental message. This creates opportunities for summarizing technologies that can overcome language barriers and make knowledge available to a broader audience, irrespective of their mother tongue.

However, its single-sentence limitation may result in the oversimplification of complex subjects and the neglect of delicate emotional subtleties inherent in human

communication. These constraints present prospects for additional enhancement and advancement, compelled by the need to explore more deeply the complex interplay between factual precision, emotional impact, and significant portrayal.

4.1.2. The ARXIV dataset

ARXIV (Clement et al., 2019) is a valuable resource that provides insights into the complex realm of scientific communication. It focuses on scientific articles, offering a comprehensive collection of research papers from various disciplines, enabling models to understand domain-specific vocabulary and specialized language. ARXIV also emphasizes the intricate nature of research papers, allowing models to extract important discoveries and comprehend the fundamental logic of reasoning presented in the document.

The Arxiv dataset comprises a total of 215K samples. The training dataset consists of 203K items. However, for the purpose of validation and testing, it includes a total of 6.4K samples for each category.

The platform also transcends the limitations of a singular writing style, incorporating the polished writing style of experienced researchers, the passionate expression of emerging scientists, and the cooperative approach of publications written by multiple authors. This diversity challenges models to adjust their summarizing algorithms to accurately capture the author's voice and intention. ARXIV's significance extends to the wider domain of summarization, addressing the knowledge gap by producing succinct summaries that render intricate discoveries into easily comprehensible English. This enables citizens to actively participate in scientific progress, promoting a well-informed society. ARXIV-trained models can streamline the process of literature reviews, enabling scientists to focus on analysis and invention rather than information retrieval. Moreover, ARXIV's ability to spark scientific curiosity and fuel creativity is crucial. However, it also faces difficulties in accurately depicting specialized vocabulary and domain-specific terminology in summaries. Additionally, ARXIV models must be updated and adjusted to stay current and effective in the ever-evolving scientific research landscape.

Despite these challenges, ARXIV's impact on summarization is undeniable, as it functions as a crucible for evaluating and improving summary techniques within the context of intricate scientific communication.

4.2. Evaluation Metrics

4.2.1. Summarization metrics

In order to evaluate the quality of summaries in terms of summarization performance, we find the F1 score of ROUGE-1, ROUGE-2, and ROUGE-L. Although they are not ideal, ROUGE metrics continue to be essential for evaluating text-summarizing models. This is because they have the capacity to quantify the overlap that exists between generated summaries and references that have been authored by humans. Furthermore, they provide a consistent measurement of factual accuracy and fluency.

In addition to that, we added the BERTScore of the created summary in comparison to the summaries of the ground truth. Because of its capacity to combine semantic similarity and fluency, BERTScore is able to perform better than ROUGE in some cases, as shown by many studies (Dreyer et al., 2023; Fischer et al., 2022; Ladhak et al., 2021; Louis & Nenkova, 2013; Pagnoni et al., 2021) . Because of this, it is able to recognize the subtle differences in meaning that go beyond simple word matching.

4.2.2. Faithfulness metrics

To evaluate faithfulness, we depend on the QuestEval metric that was briefly explained in Section 2.9.4. That is because QuestEval differs from traditional metrics by examining the core aspects of abstractive summarization (Scialom et al., 2021). It evaluates models based on their ability to accurately convey facts, maintain fluency, and effectively address user-generated questions. This approach expands the scope of evaluation beyond simple text comparison, allowing for a more comprehensive assessment of the comprehension and usefulness of generated summaries. Ultimately, QuestEval aims to enable users to actively interact with information rather than simply be informed by it.

Furthermore, we report the BERTScore of the generated summary against the source document as a faithfulness metric called BS-Fact since BS-Fact correlates better with human evaluation (Fischer et al., 2022). For a detailed explanation of the used metrics, refer to Section 2.9.

4.3. Faithfulness-Aware Pretraining Setup

As outlined in Section 3.1, our initial approach involves ranking sentences within each text according to their faithfulness and importance in summarizing. Initially, we performed an NER operation to extract named entities from the pretraining datasets. We employed the spacy pipeline for NER as our chosen method for identifying and classifying named entities. We calculated the ROUGE-1 score for each sentence in comparison to the entire document. Subsequently, we evaluated each sentence using the scoring method (equation 3.1) with α set to 1.0 and β set to 0.25. We conducted preprocessing on the pretraining datasets, masking out tokens in each phrase according to equation 3.2. Non-named-entity tokens were masked with a $\lambda 1$ value of 0.5, while named-entity tokens were masked with a $\lambda 2$ value of 0.6.

In accordance with the information provided in Section 3.3, we further attach the connection token at the beginning of each document.

We employ the BART-Large architecture, a Facebook implementation provided in Huggingface, for pretraining. This design has been pretrained on the BART-MLM challenge, as outlined in the original work.

We report this model as the baseline model of the pretraining stage. Subsequently, we conducted additional pretraining of BART-Large using the preprocessed XSUM and ARXIV datasets, resulting in the models BART-XFA and BART-AFA, respectively, according to Table 4.1 which displays the pretraining procedure and hyperparameters of the two models.

Table 4.1. Hyperparameters for the pretraining stage.

	BART-XFA	BART-AFA
Learning rate	2×10^{-5}	2×10^{-5}
Batch size	8	1
Max Length	512	1024
Label smoothing	0	0.1
Training steps	34500	80000

4.4. Finetuning Setup

Following the pretraining phase, we proceeded to refine the models specifically for the purpose of summarization, which was the subsequent task. In order to ensure an accurate evaluation of the results, we consistently employ identical hyperparameters for BART, BART-XFA, and BART-AFA in all experiments. We employed a batch size of 8 and an initial learning rate of 2×10^{-5}. The models were fine-tuned on a single Tesla M40 24GB GPU for one epoch, as our computing resources were constrained. Subsequently, we assessed the results using the same hardware. The scoring process for QuestEval required 13 hours per experiment on the ARXIV dataset and 25 hours on the XSUM dataset. The longer time on XSUM is due to the larger number of test samples. On the other hand, the computation of BERTScore and BS Fact took less than one hour. Given ARXIV's exclusive focus on scientific papers, the retrieved named entities are not relevant for a wide-range dataset such as XSUM. Consequently, our findings indicate that it is unsuitable to do fine-tuning on the XSUM dataset using BART-AFA. Nevertheless, we conducted fine-tuning on BART-XFA using both the XSUM and ARXIV datasets, with a preference for the XSUM dataset because of its wider domain coverage. The resources and tools utilized in the experiments are illustrated in Table 4.2.

Table 4.2. The utilized tools and resources.

Tool / Resources	Version
System	Windows 11
GPU	Tesla-M40-24GB
Python	3.10.11
Torch	2.0.0
Transformers	4.30.2
Pandas	1.5.3
Scipy	1.11.1

5. RESULTS AND DISCUSSION

5.1. Experimental Results

As described in Section 4.3, we reported our results on our proposed models, BART-XFA and BART-AFA, which are pretrained and finetuned on the XSUM dataset and ARXIV dataset, respectively. Then we compare the results against BART-MLM. The reader can refer to Table 5.1 as well as Section 4.3 for more information about the description of the used models.

Table 5.1. A brief description of the used models.

Model	Description	Dataset	Pretraining Strategy
BART-MLM (Lewis et al., 2019)	The baseline model. It is the pretrained BART-Large implementation by Facebook, and finetuned on the abstractive summarization downstream datasets with the hyperparameters illustrated in Section 4.	-	BART-MLM
BART-XFA	The BART-Large architecture after pretrained on the XSUM dataset as described in Section 4.	XSUM	Our Faithfulness aware strategy
BART-AFA	The BART-Large architecture after pretrained on the ARXIV dataset as described in Section 4	ARXIV	Our Faithfulness aware strategy

According to the data presented in Table 5.2 and Figure 5.1 regarding the ARXIV dataset, our suggested model, BART-XFA, outperforms the classic BART-MLM model in terms of the QuestEval metric, achieving a score of 37.1. Additionally, it outperforms BART-MLM in terms of BS-Fact score. In addition, BART-XFA achieved the highest score of 38.4 in ROUGE-1, which is a statistic used for evaluating summarization. BART-XFA exhibits no decline in other summarization metrics since it achieves comparable outcomes to the BART-MLM model.

Table 5.2. Testing scores on ARXIV dataset where target length=128.

			BART-MLM	BART-XFA	BART-AFA
Faithfulness	QuestEval		35.99	**37.09**	33.32
	BS-Fact	F1	86.59	86.87	85.62
		Precision	91.53	91.67	90.23
		Recall	82.17	82.57	81.47
Summarization	ROUGE-1	F1	38.20	**38.40**	38.34
		Precision	42.25	40.89	44.56
		Recall	36.27	37.27	35.05
	ROUGE-2	F1	11.77	11.67	**12.16**
		Precision	13.20	12.55	14.31
		Recall	11.07	11.27	11.03
	ROUGE-L	F1	33.43	33.79	**33.85**
		Precision	35.70	34.71	38.09
		Recall	30.54	31.59	29.88
	BERTScore	F1	85.09	84.97	85.31
		Precision	85.41	85.13	**85.90**
		Recall	84.79	84.82	84.76

A number of summarization measures, such as ROUGE-2, ROUGE-L, and BERTScore, show that BART-AFA performs well, but when it comes to faithfulness, it can be considered lacking. When we applied the named entity identification tool to the ARXIV dataset, we found that it performed poorly, which

led us to conclude that this phenomenon was its cause. For more information, please refer to Section 5.2.

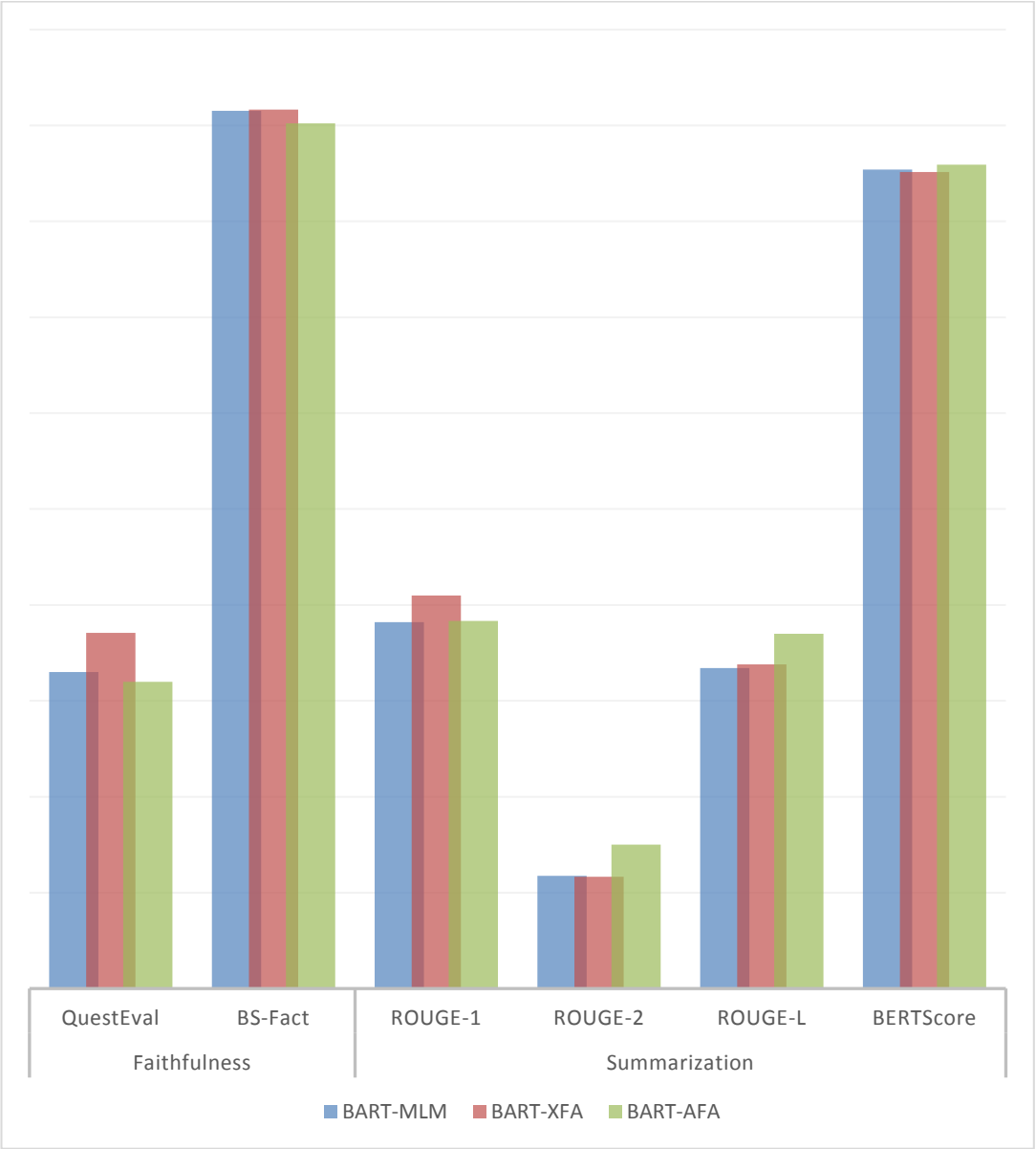

Figure 5.1. Testing scores on ARXIV dataset where target length=128.

Table 5.3 and Figure 5.2 illustrate the outcomes of the experiment carried out on the ARXIV dataset using identical hyperparameters as the Table 5.2 experiment, with the exception of the target length, since the target length used in Table 5.2 experiment is 128.

The results of Table 5.3 indicate that BART-XFA continues to exhibit similar patterns as the prior experiment, with the highest levels of faithfulness metrics in terms of QuestEval without any decrease in summarization metrics. However, it

achieves favorable outcomes in the majority of summary metrics. Similarly, BART-AFA demonstrates strong performance in summarization but performs poorly in terms of faithfulness.

Table 5.3. Testing scores on ARXIV dataset where target length=64.

			BART-MLM	BART-XFA	BART-AFA
Faithfulness	QuestEval		35.79	**36.21**	33.02
	BS-Fact	F1	85.08	85.03	84.49
		Precision	90.98	90.89	89.79
		Recall	79.92	79.89	79.80
Summarization	ROUGE-1	F1	35.50	35.19	**36.35**
		Precision	36.24	36.30	37.60
		Recall	35.09	34.45	35.57
	ROUGE-2	F1	11.28	11.18	**11.90**
		Precision	11.54	11.56	12.37
		Recall	11.16	10.94	11.65
	ROUGE-L	F1	29.96	29.83	**30.92**
		Precision	29.73	29.94	31.14
		Recall	28.80	28.41	29.45
	BERTScore	F1	85.36	85.30	85.57
		Precision	85.46	85.41	**85.90**
		Recall	85.28	85.19	85.45

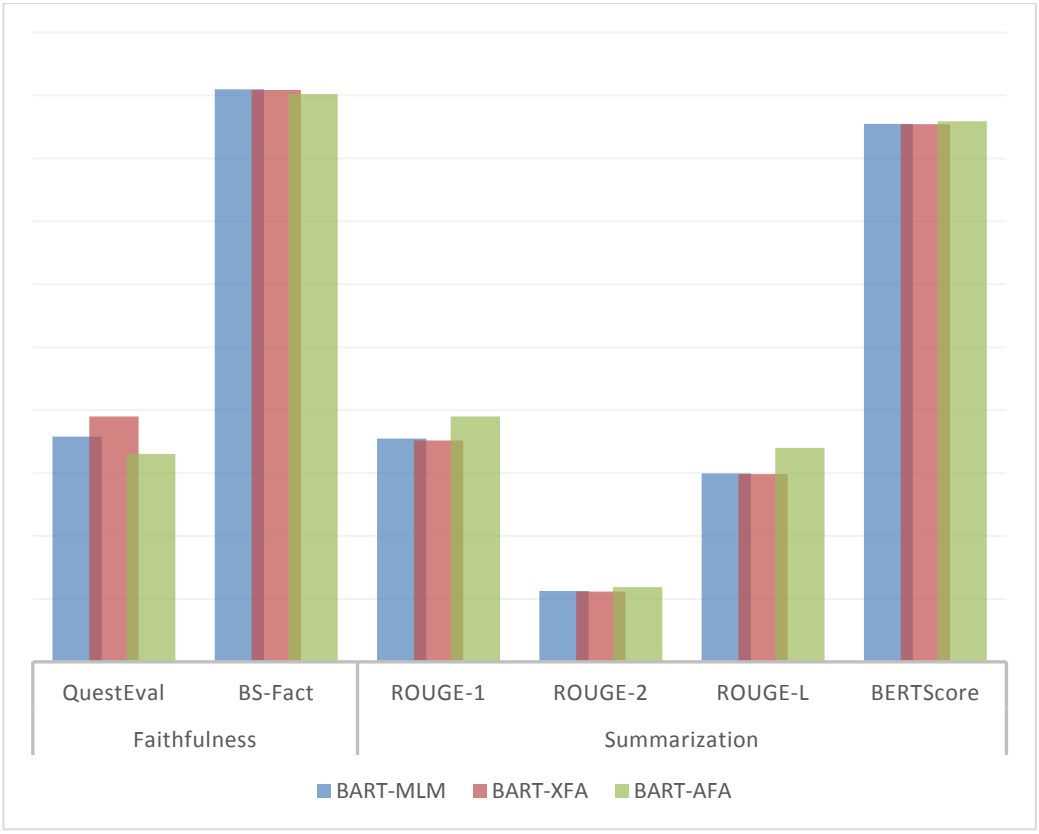

Figure 5.2. Testing results reported on ARXIV dataset where target length=64.

Table 5.4 and Figure 5.3 reveal that BART-XFA achieved the top results in terms of faithfulness metrics on the XSUM dataset. Its QuestEval score was 39.00, which indicates that it was the most faithful. BS-Fact scores do not differ significantly from one another in any significant way. In a number of summarization measures, BART-XFA performs better than the traditional BART-MLM. It achieves a ROUGE-1 score of 40.12 and a ROUGE-2 score of 17.45.

Table 5.4. Experimental results reported on XSUM dataset.

			BART-MLM	BART-XFA
Faithfulness	QuestEval		38.62	**39.00**
	BS-Fact	F1	84.58	84.78
		Precision	89.44	89.42
		Recall	80.25	80.61

Table 5.4. (Continued) Experimental results reported on XSUM dataset.

			BART-MLM	BART-XFA
Summarization	ROUGE-1	F1	39.85	**40.12**
		Precision	42.84	39.44
		Recall	38.84	42.65
	ROUGE-2	F1	17.24	**17.45**
		Precision	18.60	17.21
		Recall	16.74	18.52
	ROUGE-L	F1	**31.77**	31.47
		Precision	34.16	30.98
		Recall	30.94	33.43
	BERTScore	F1	90.49	90.31
		Precision	**90.91**	90.17
		Recall	90.08	.90.48

Although our proposed method, BART-XFA, is mostly comparable to BART-MLM as both of them are pretraining strategies and they are finetuned using the same hyperparameters in terms of beam window size, target length, and number of epochs, we compare the results with the state-of-the-art results on the XSUM dataset. Table 5.5 demonstrates that BART-XFA overcomes most of the related works in terms of QuestEval. Although the lookahead and ranking method (Wan et al., 2023) overcomes our method, a hybrid method, BART-XFA with lookahead and ranking functionality, is supposed to produce better faithfulness results. Our proposed method shows lower summarization metrics since we finetuned the model with few-shot settings as described in Section 4.

Table 5.5. Performance comparison against related works on the XSUM dataset. The results are reported in terms of F1 score for ROUGE metrics and QuestEval, and precision for BS and BS-Fact.

Model	Summarization				Faithfulness	
	R1	R2	RL	BS	BS-Fact	QuestEval
BART-MLM (Lewis et al., 2019)	38.85	17.24	31.77	90.91	89.44	38.62
CLIFF (Cao & Wang, 2021)	-	-	35.86	-	-	33.35
BART+correct(S. Chen et al., 2021)	-	-	36.62	91.10	-	-
Greedy+Lookahead (Wan et al., 2023)	-	-	36.25	92.11	89.71	37.17
Beam +Lookahead (Wan et al., 2023)	-	-	35.27	91.94	90.78	39.24
Beam +Lookahead +Ranking (Wan et al., 2023)	-	-	34.71	91.10	90.78	41.94
BART-XFA	40.12	17.45	31.47	90.17	89.42	39.00

Because of the lack of faithfulness works on the ARXIV dataset, our comparison against the majority of the related works on the ARXIV dataset is mostly reported in terms of summarization metrics. In addition to the MNELM model (Berezin & Batura, 2023), we compare the results against PEGASUS (J. Zhang et al., 2020), which is a transformer-based LLM, and to the PTGEN+Cov model (See et al., 2017), which is a pointer-generator baseline with coverage mechanism. As well, we used DiscAttn (Cohan et al., 2018) as a baseline model that depends on the traditional seq2seq architecture. As shown in Table 5.6, it is notable that PEGASUS has better scores than our proposed models. This is due to the fact that the objective of our

proposed models is to enhance faithfulness. In addition, our models are finetuned for a limited number of epochs, as mentioned in Section 4. Our method is more comparable to the BART-MLM and the MNELM since the hyperparameters are approximately similar to those for the BART-XFA and BART-AFA experiments. However, our proposed models overcome the majority of works in summarization metrics such as ROUGE-1, ROUGE-2, and ROUGE-L.

Table 5.6. Performance comparison against related works on the ARXIV dataset. The results are reported in terms of F1 score for ROUGE metrics and QuestEval, and precision for BS and BS-Fact.

Model	Summarization				Faithfulness	
	R1	R2	RL	BS	BS-Fact	QuestEval
BART-MLM (Lewis et al., 2019)	38.20	11.77	33.34	85.41	91.53	35.99
PEGASUS (J. Zhang et al., 2020)	43.82	16.74	39.09	-	-	-
PTGEN+Cov (See et al., 2017)	32.06	9.04	25.16	-	-	-
DiscAttn (Cohan et al., 2018)	35.80	11.05	31.80	-	-	-
MNELM (Berezin & Batura, 2023)	36.00	13.00	32.00	-	-	-
BART-AFA	38.43	12.16	33.85	85.13	90.23	33.32
BART-XFA	38.40	11.67	33.79	85.13	91.67	37.09

5.2. The Effect of NER Quality on Faithfulness

Figure.5.1 and Figure.5.2 indicate that BART-AFA exhibits poor faithfulness scores. This can be reasonably explained as the employed named entity recognition technology was trained to identify general named entities in wide range datasets. The

ARXIV dataset is a domain-specific dataset that contains research papers. Consequently, the named entity recognition tool was unsuccessful in accurately identifying the entities. Our custom masking tokens rely on two metrics: the faithfulness metric, which is determined by the number of named entities, and the summarization metric, which is represented by the ROUGE-1 score.

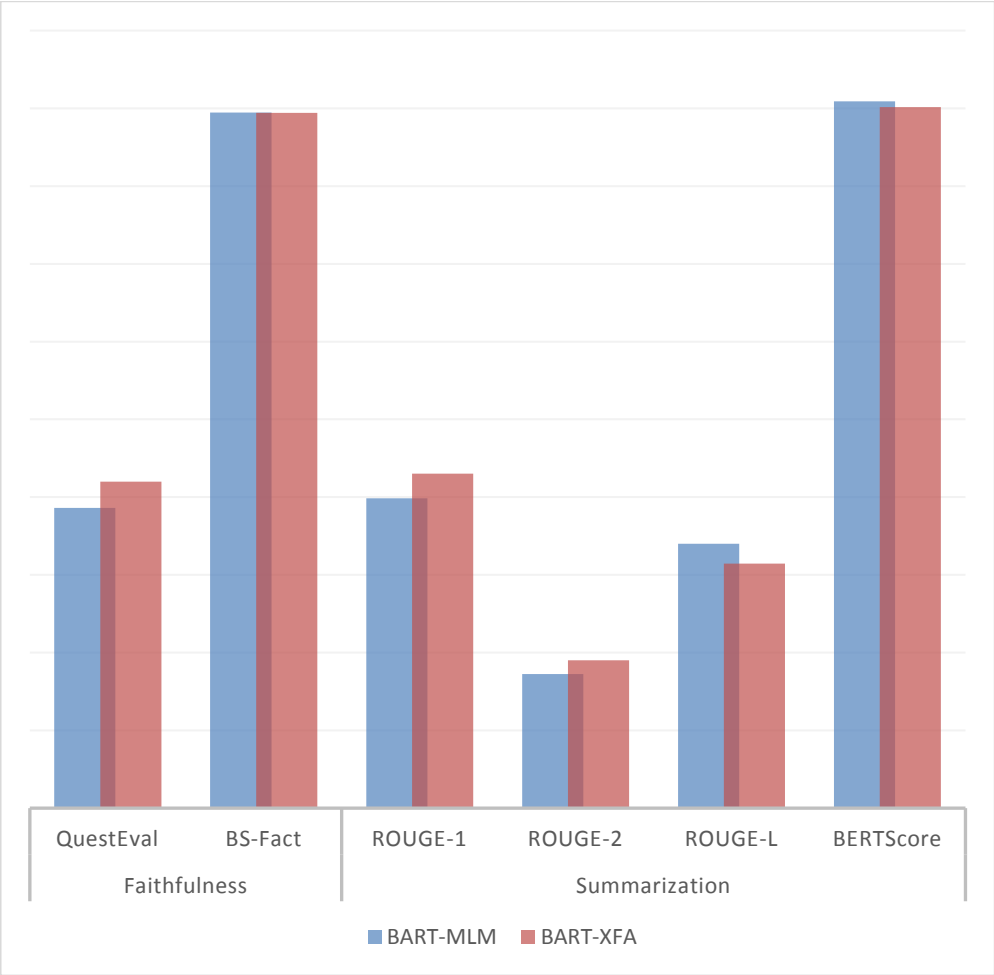

Figure 5.3. Experimental results reported on XSUM dataset.

Consequently, the BART-AFA score indicates lower levels of faithfulness. Undoubtedly, BART-AFA achieves superior summarization outcomes due to the fact that the domain of the datasets does not impact the ROUGE-1 metric.

These results are highly significant and suggest the need for implementing a custom QuestEval metric. The reason for this is because we employed the identical named entity identification tool utilized in the official implementation of QuestEval, namely the spacy pipeline of NER. However, it demonstrates an inability to identify the listed items within the ARXIV dataset.

5.3. Customized Faithfulness Masking Probability

On the basis of the examination of the reported results, it is obvious that BS-Fact remains consistent across all of the studies, with no significant changes being found. Given that the masking function does not take into account either the BERTScore of the tokens or the BERTScore of the detected entity tokens, this is a reasonable observation to make. As a helpful idea, the utilization of various heuristics in the computation of the specific likelihood of concealing allegiance can be implemented. The computer resources that were available to us during our experiment were so limited that we were unable to add BERTScore to the scoring procedure. On the other hand, this can be performed in an effective manner in subsequent research. In addition, the scoring algorithm for masking can also make use of other metrics of faithfulness, especially those that require less computational power.

6. CONCLUSION AND FUTURE WORKS

6.1. Thesis Conclusion

Within the scope of this thesis, we investigate the impact of the pretraining method on faithfulness in the process of abstractive text summarization. Additionally, we present a novel pretraining strategy that considers faithfulness. Our methodology involves prioritizing sentences in datasets utilizing the ROUGE scoring and NER extractor, and then stimulating the BART large language model to attend more to faithfulness-related context by trying to predict faithfulness-related masked-out tokens. We demonstrate that our approach surpasses the outcomes of traditional MLM on two subsequent abstractive text summarization datasets.

The results of our study showed that the finetuned BART-XFA achieved greater faithfulness scores in all experiments, as measured by the QuestEval metric, compared to BART-MLM. Importantly, this improvement in faithfulness did not have a detrimental impact on the summarization score. However, BART-XFA and BART-AFA achieved more favorable outcomes compared to typical BART-MLM results across nearly all summary metrics.

In addition, we explored the impact of the reliability of named entity detection methods on the faithfulness of the results obtained by our customized pretraining approach. We claim this conclusion based on the restricted faithfulness score of BART-AFA. This is because BART-AFA was pretrained by masking tokens using a general named entity recognition method, but on a scientifically specialized domain dataset known as the ARXIV dataset. We deliberated on the implementation of a customized QuestEval metric in order to be able to more accurately measure the degree of faithfulness in abstractive text summarization.

Furthermore, we demonstrated that our custom masking function may be further tailored by incorporating other measures of faithfulness. This typically necessitates powerful processing resources due to the time-consuming nature and intensive computations required by current faithfulness measurements. We can adjust the

masking technique to achieve a compromise between maintaining faithfulness and providing summarization. This can be done by utilizing the scalars defined in our custom function.

The results of our study clearly illustrate the substantial influence of pretraining techniques on the faithfulness of abstractive text summarization. This advancement clears the path for the creation of more reliable and precise summarization systems, promoting a more profound comprehension of the subtleties and intricacies of natural language.

6.2. Future Works

The findings have opened up a number of different doors for further investigation in the future. For your consideration, below are a few important directions:

- Expand the scope of the study: Perform an extensive inquiry into faithfulness-aware pretraining techniques by assessing them on a wider array of summarization tasks and datasets, encompassing various areas, genres, and languages.
- Investigate domain-specific NER: Investigate the potential of domain-specific NER techniques to further refine faithfulness in abstractive summarization, particularly in specialized fields with unique terminology and concepts.
- Optimize the custom masking function: Enhance the efficiency and effectiveness of the custom masking function by exploring alternative approaches to measuring faithfulness and optimizing the trade-off between faithfulness and summarization quality.
- Perform further preliminary experiments using various hyperparameters.

These lines of inquiry should lead us to a better understanding of abstractive text summary faithfulness and the development of more effective strategies for creating summaries that accurately capture the content of the source material.

.

REFERENCES

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., & others. (2023). Palm 2 technical report. *ArXiv Preprint ArXiv:2305.10403*.

Berezin, S., & Batura, T. (2023). Named entity inclusion in abstractive text summarization. *ArXiv Preprint ArXiv:2307.02570*.

Cao, S., & Wang, L. (2021). CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. *ArXiv Preprint ArXiv:2109.09209*.

Chen, S., Zhang, F., Sone, K., & Roth, D. (2021). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. *ArXiv Preprint ArXiv:2104.09061*.

Chen, X., Li, M., Gao, X., & Zhang, X. (2022). Towards improving faithfulness in abstractive summarization. *Advances in Neural Information Processing Systems*, *35*, 24516–24528.

Clement, C. B., Bierbaum, M., O'Keeffe, K. P., & Alemi, A. A. (2019). On the use of arxiv as a dataset. *ArXiv Preprint ArXiv:1905.00075*.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). *A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv Preprint ArXiv:1810.04805*.

Dreyer, M., Liu, M., Nan, F., Atluri, S., & Ravi, S. (2023). Evaluating the tradeoff between abstractiveness and factuality in abstractive summarization. *Findings of the Association for Computational Linguistics: EACL 2023*, 2044–2060.

Durmus, E., He, H., & Diab, M. (2020). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *ArXiv Preprint ArXiv:2005.03754*.

Falke, T., Ribeiro, L. F. R., Utama, P. A., Dagan, I., & Gurevych, I. (2019). Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2214–2220.

Fischer, T., Remus, S., & Biemann, C. (2022). Measuring faithfulness of abstractive summaries. *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, 63–73.

Goyal, T., & Durrett, G. (2021). Annotating and modeling fine-grained factuality in summarization. *ArXiv Preprint ArXiv:2104.04302*.

Gu, J., Lu, Z., Li, H., & Li, V. O. K. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *ArXiv Preprint ArXiv:1603.06393*.

Kale, M., & Rastogi, A. (2020). Text-to-text pre-training for data-to-text tasks. *ArXiv Preprint ArXiv:2005.10433*.

Ladhak, F., Durmus, E., He, H., Cardie, C., & McKeown, K. (2021). Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. *ArXiv Preprint ArXiv:2108.13684*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv Preprint ArXiv:1910.13461*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 74–81.

Liu, Y., & Liu, P. (2021). SimCLS: A simple framework for contrastive learning of abstractive summarization. *ArXiv Preprint ArXiv:2106.01890*.

Liu, Y., Liu, P., Radev, D., & Neubig, G. (2022). BRIO: Bringing order to abstractive summarization. *ArXiv Preprint ArXiv:2203.16804*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv Preprint ArXiv:1907.11692*.

Louis, A., & Nenkova, A. (2013). Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, *39*(2), 267–300.

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. *ArXiv Preprint ArXiv:2005.00661*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*.

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., & others. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *ArXiv Preprint ArXiv:1602.06023*.

Narayan, S., Cohen, S. B., & Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv Preprint ArXiv:1808.08745*.

Pagnoni, A., Balachandran, V., & Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. *ArXiv Preprint ArXiv:2104.13346*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. *ArXiv Preprint ArXiv:1705.00108*.

Ravaut, M., Joty, S., & Chen, N. F. (2022). SummaReranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. *ArXiv Preprint ArXiv:2203.06569*.

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. *ArXiv Preprint ArXiv:1509.00685*.

Scialom, T., Dray, P.-A., Gallinari, P., Lamprier, S., Piwowarski, B., Staiano, J., & Wang, A. (2021). Questeval: Summarization asks for fact-based evaluation. *ArXiv Preprint ArXiv:2103.12693*.

See, A., Liu, P. J., & Manning, C. D. (2017). *Get To The Point: Summarization with Pointer-Generator Networks*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wan, D., & Bansal, M. (2022). FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization. *ArXiv Preprint ArXiv:2205.07830*.

Wan, D., Liu, M., McKeown, K., Dreyer, M., & Bansal, M. (2023). Faithfulness-aware decoding strategies for abstractive summarization. *ArXiv Preprint ArXiv:2303.03278*.

Xiao, W., & Carenini, G. (2022). Entity-based spancopy for abstractive summarization to improve the factual consistency. *ArXiv Preprint ArXiv:2209.03479*.

Xu, S., Li, H., Yuan, P., Wu, Y., He, X., & Zhou, B. (2020). Self-attention guided copy mechanism for abstractive summarization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1355–1362.

Zhang, H., Yavuz, S., Kryscinski, W., Hashimoto, K., & Zhou, Y. (2022). Improving the faithfulness of abstractive summarization via entity coverage control. *ArXiv Preprint ArXiv:2207.02263*.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *International Conference on Machine Learning*, 11328–11339.

Zhang, M., Zhou, G., Yu, W., Huang, N., & Liu, W. (2022). A comprehensive survey of abstractive text summarization based on deep learning. *Computational Intelligence and Neuroscience*, *2022*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *ArXiv Preprint ArXiv:1904.09675*.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., & others. (2023). A survey of large language models. *ArXiv Preprint ArXiv:2303.18223*.

CURRICULUM VITAE

Name Surname : Mohanad ALREFAAI

EDUCATION:

- **Undergraduate** : 2017, Near East University, Faculty of Engineering, Computer Engineering Department
- **Graduate** : 2023, Sakarya University, Computer and Information Engineering Department, Computer Engineering Program

PROFESSIONAL EXPERIENCE AND AWARDS:

- Software Engineer, Al Muraqeb Technology, Dubai, UAE (2018 – 2023)

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Alrefaai M. and Akgün D. (2023, 23-25, November). A Pretraining Strategy to Improve Faithfulness in Abstractive Text Summarization. *7th International Symposium on Innovative Approaches in Smart Technologies*, Istanbul, Turkey.