Text Summarization and Sentiment Analysis of Drug Reviews: A Transfer Learning Approach

By

Gloria Abuka

APPROVED:

Graduate Committee

Supervisor Dr. Jaishree Ranganathan (Computer Science)

Dr. Zhijiang Dong (Computer Science)

Dr. Yi Gu (Computer Science)

Dr. Medhar Sarkar, Chairperson Computer Science Department

Dr. David Butler, Interim Dean of the College of Graduate Studies

Text Summarization and Sentiment Analysis of Drug Reviews: A Transfer Learning Approach

By

Gloria Abuka

A thesis submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE
in
Computer Science

Middle Tennessee State University

May 2023

Thesis Committee:

Dr. Jaishree Ranganathan

Dr. Zhijiang Dong

Dr. Yi Gu

ACKNOWLEDGEMENTS

Now to the King eternal, immortal, invisible, the only wise God, be honor and glory forever and ever.

I would like to sincerely appreciate my thesis advisor, Dr. Jaishree Ranganathan, for her contributions, guidance, patience, and willingness to answer my questions, which have been crucial to the success of this project.

I would also like to extend my heartfelt thanks to my advisory committee members, Dr. Zhijiang Dong and Dr. Yi Gu, for their insightful input, which has helped to improve this work.

I am immensely grateful to my husband, Dr. Samuel Haruna, and my sons, Jesse and Asher, for their unwavering love and all-around support that have helped me succeed. I couldn't have asked for a better team. Thank you, guys!!!

Special thanks to my parents and parents-in-law for their constant prayers and support. I want to express my gratitude to my siblings Lilian, Gabriel, and Hope, as well as my friends Helen and Isaac, for their prayers and emotional support.

God bless you all, I could not have done this without you, Thank you!!!

ABSTRACT

Transfer learning is a machine learning method where a model that has been trained on a specific or general task (source domain) is reused as a starting point for a similar task in a new model (target domain). This is an important concept in the Natural Language Processing field because of its ability to produce remarkable results from small datasets. Text summarization produces a concise and meaningful form of text from a larger one while sentiment analysis distinguishes the polarity present in the text. News and scientific articles have been used in text summarization models over the years, but drug reviews have gotten considerably less attention. This study proposes a text summarization and sentiment analysis method based on the transformer architecture for the 10 most useful reviews for 500 different drugs from a dataset of drugs reviews. We created human summaries for the drug reviews manually and compared the performance of a fine-tuned Text-to-Text Transfer Transformer (T5) model and Pre-training with extracted gap-sentences for abstractive summarization (PEGASUS) models with that of a Long Short-Term Memory (LSTM) model. Additionally, we assessed the impact of various preprocessing steps on the ROUGE scores. We also fine-tuned the Bidirectional Encoder Representation from Transformers (BERT) model for sentiment analysis in comparison to an LSTM model. Our T5-Base model had the best results with average ROUGE1, ROUGE2, and ROUGEL scores of 50.31, 29.14, and 40.06 respectively while the BERT model achieved an accuracy of 84% for the sentiment analysis task. We evaluated our fine-tuned models on a dataset of BBC news summaries for text summarization and we achieved average ROUGE1, ROUGE2, and ROUGEL scores of 72.20, 63.59, and 57.42 respectively. Our models outperformed two previous works, which had ROUGE1, ROUGE2, and ROUGEL of 47.0, 33.0, 42.0 and 47.30, 26.50 and 36.10 respectively.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

LIST OF SYMBOLS AND ABBREVIATIONS viii

CHAPTER I. **INTRODUCTION** 1

Transfer Learning 1

Text Summarization and Sentiment Analysis 2

CHAPTER II. **BACKGROUND** 6

Text Summarization 6

Sentiment Analysis 9

CHAPTER III. **OVERVIEW OF MODELS** 12

Bi-directional Encoder Representation from Transformers (BERT) 12

Pegasus Model Overview 12

Text-to-Text Transfer Transformer (T5) model 13

Long Short-Term Memory (LSTM) 14

CHAPTER IV. **METHODS** 15

Dataset 15

Data Preprocessing 15

Fine-Tuning of the Pre-trained Transformer Models for Text Summarization 17

Sentiment Analysis Models (Fine-tuned BERT and the LSTM Model 20

CHAPTER V. **RESULTS AND DISCUSSIONS** 22

Overview of Evaluation Metrics 22

T5-Small 23

Statistical analysis for the T5-Small model 24

T5-Base 25

Pegasus Base 27

LSTM for Text Summarization 29

Sentiment Analysis (BERT and LSTM) 35

Limitations 37

CHAPTER VI. **CONCLUSIONS** 38

BIBLIOGRAPHY 40

LIST OF TABLES

Table 1 – List of hyperparameters and values 18

Table 2 – Explanation of the list of hyperparameters 19

Table 3 – Evaluation metrics 22

Table 4 – Average ROUGE scores for the T5-small model 24

Table 5 – UCI drug reviews dataset 24

Table 6 – BBC news dataset 25

Table 7 – Average ROUGE scores(Drug reviews) 25

Table 8 – Confidence interval(Drug reviews) 26

Table 9 – Average ROUGE scores(BBC news) 26

Table 10 – Confidence interval(BBC news) 26

Table 11 – Performance comparison on previous work on BBC news dataset . . . 27

Table 12 – Average ROUGE scores (Drug reviews) 27

Table 13 – Confidence interval(Drug reviews) 28

Table 14 – Average ROUGE scores (BBC news) 28

Table 15 – Confidence interval(BBC news) 28

Table 16 – Average ROUGE scores(Drug reviews) 29

Table 17 – Confidence interval(Drug reviews) 29

Table 18 – Average ROUGE scores(BBC news) 29

Table 19 – Confidence interval(BBC news) 30

Table 20 – Example summaries (Drug reviews) 32

Table 21 – Example summaries (BBC news) 34

Table 22 – The F1-scores for the BERT model with the preprocessing steps . . . 35

Table 23 – The F1-scores for the BERT model without the preprocessing steps . 35

Table 24 – The F1-scores for the LSTM model with the preprocessing steps . . . 36

Table 25 – The F1-scores for the LSTM model without the preprocessing steps . 36

LIST OF FIGURES

Figure 1 – Sentiment Analysis at a glance 4

Figure 2 – Overview of the proposed aproach 5

Figure 3 – Word cloud representation of words in the dataset 15

Figure 4 – Word count plots for the review text and summary for the UCI drug reviews dataset 17

Figure 5 – Distribution of the summaries after regrouping 20

Figure 6 – Positive sentiment 36

Figure 7 – Neutral sentiment 37

Figure 8 – Negative sentiment 37

LIST OF SYMBOLS AND ABBREVIATIONS

BERT - Bidirectional Encoder Representation from Transformers

UCI - University of California Irvin

LSTM - Long Short-Term memory

T5 - Text-To-Text Transfer Transformer

PEGASUS - Pre-training with Extracted Gap-sentences for Abstractive Summarization

ROUGE - Recall-Oriented Understudy for Gisting Evaluation

CNN - Convolutional Neural Network

CHAPTER I.

INTRODUCTION

Transfer Learning

To effectively train a machine learning model requires an enormous amount of data and computational power, especially for tasks in Natural Language and Image processing. In reality, these two important factors mentioned above are not always readily available as the need arises. The process of collecting and labeling data can be difficult and time-consuming, this problem can be significantly addressed with a concept in deep learning known as Transfer Learning. Transfer Learning is a machine learning method where a model that has been trained on a specific or general task (source domain) is reused as a starting point for a similar task in a new model (target domain). A model is pre-trained on a large amount of data and can be fine-tuned on a downstream task [29]. This method saves time as it enables small datasets to achieve astonishing results they otherwise would not have achieved. This technique works by transferring knowledge from a specific or general task to improve the rate of learning of another model performing a similar task [36].

The training data and the downstream task data belong to different sub-domains but are connected by a larger high-level domain [40]. Assuming there is a source domain *Ds* and a source task *Ts*, a target domain *Dt* and a target task *Tt*. Transfer learning can be viewed as a way of improving a target learning function *Ft(.)* by the use of those details common to both domains *Ds* and *Dt* where *Ds* != *Dt* and *Ts* != *Tt* (both domains and tasks are not exactly the same) [40].

Transfer learning just like regular machine learning models can be either supervised, semi-supervised, or unsupervised. In a supervised method, you have a large amount of labeled source data and a few labeled target data [5]. The semi-supervised method is similar to the supervised method with a large set of labeled source data and also a large set of unlabeled target data [5] while the unsupervised method works with a set of completely unlabeled target data [10].

In non-technical terms, assuming two individuals desire to learn how to play piano and one of them has prior knowledge from playing guitar while the other person knows nothing about music [27]. The knowledge of guitar for that first individual may become helpful in the piano domain and will be used as a starting point for the new skill being acquired [27]. This will enable him to learn faster than the other person with no prior knowledge of any musical instrument. The guitar and piano can be seen as two sub-domains of a large music domain. Similarly, a model that was pre-trained on a dataset of camera customer reviews can help another model intended to be trained on food reviews to learn faster [40]. These two domains are not exactly the same but they belong to a large domain of customer reviews [40]. This method has been used in tasks like image processing [9, 20], sentiment analysis [4, 28], and text summarization [13, 35, 38] and will be useful to this study considering the limited amount of labeled data available.

Text Summarization and Sentiment Analysis

With the advancement in technology and easy access to online feedback/review tools, there is a need for effective ways to process these reviews to generate insights that are useful to both the manufacturers and the consumers of these goods and services. It has become a common practice for individuals to check online for the opinion of others that have used a particular product or enjoyed a service before they decide on buying a product or paying for a service. These online review platforms are valuable tools to potential customers enabling them to know if a product is worthy of purchase. Also, reviews from these platforms can guide product manufacturers on adjustment decisions to better satisfy their customers. Regardless of the ability of human beings to effectively communicate, we still take things out of context and misunderstand each other sometimes. This happens to even the smartest of humans making it a much harder task for a computer program but researchers in the field of Natural language processing have provided some answers to these needs over the years.

Text summarization deals with creating a concise form of a text from a larger text while preserving its intended meaning [32]. There are two types of summarization methods that have been studied by various researchers in the past. Extractive summarization generates summaries by taking out important sentences from the original text and putting them together to form a concise and meaningful summary. In contrast, Abstractive summarization creates summaries afresh without having to reuse phrases from the original text [3, 34].

Rule-based methods have been used for text summarization tasks as seen in [31, 42]. Neural network architectures like the convolutional Neural Network (CNN), Long-short Term Memory (LSTM), Recurrent Neural Networks (RNN), and Autoencoders have also experienced a wide variety of usage [7, 16, 42] among many researchers. Transformers have greatly improved the Natural language processing community and the text summarization field [13, 18, 26, 35, 38, 45]. Most of these experiments in the papers cited above were done on news articles like the CNN dataset, DailyMail, BBC news dataset, and clinical reports dataset but not on drug reviews.

Sentiment analysis is the aspect of Natural language/Text processing that is concerned with determining the polarity or opinion in a text [33]. This opinion could either be positive, negative, or neutral as captured in Figure 1.

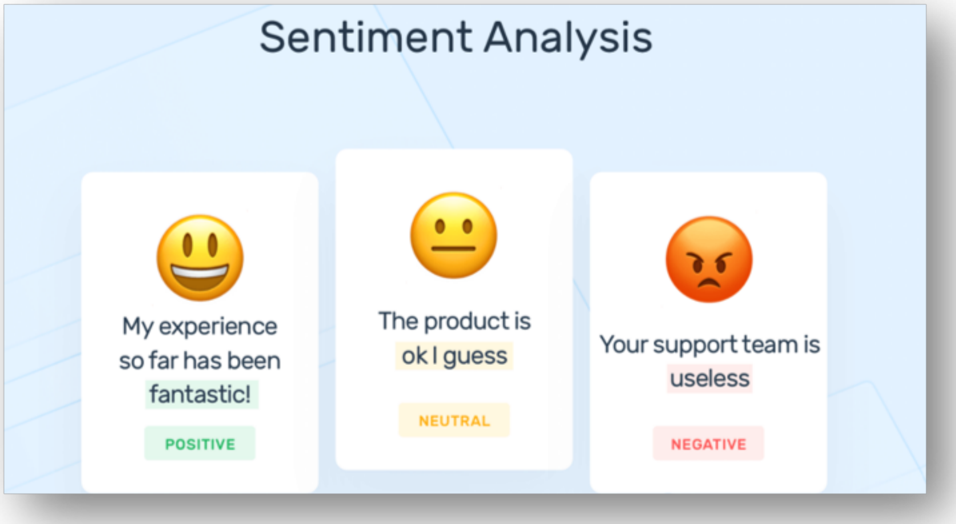

Sentiment Analysis
My experience so far has been fantastic!
POSITIVE
The product is ok I guess
NEUTRAL
Your support team is useless
NEGATIVE

Figure 1: Sentiment Analysis at a glance. Image Source [1]

Despite the duration for which sentiment analysis has been available, it is still a widely researched area because of the insight it provides into customer reviews and the like. The traditional Bag of words model [24], supervised learning algorithms like Naïve Bayes, Maximum Entropy, Support Vector Machines, K- Nearest Neighbor (KNN), and Random Forest[17, 21, 24, 33] have been used for this task by various researchers. Deep learning models like Long Short Term Memory (LSTM) and Convolutional Neural Networks [2, 24] have also gained popularity in the sentiment analysis research field. The fine-tuning of pretrained transformer architecture models have also helped to produce remarkable results as demonstrated in [2, 28].

This work proposes a text summarization method based on the transformer architecture that summarizes the ten most useful reviews of 500 drugs from a database of about 3671 drugs. We used the University of California, Irvine (UCI) drug reviews dataset that was introduced by Gräßer et al. [8, 11]. We compared the results from the fine-tuning of the Text-to-Text Transfer Transformer (T5), [29] and Pegasus [43], to that of an LSTM model for the text summarization task and examined the effects of various preprocessing steps on the results. We created human summaries for the 500 drugs to train and evaluate the model. We also trained and tested our models on the BBC news dataset [12]. We further conducted sentiment analysis on the review summaries. For each of the summaries, we assigned sentiment scores with the pipeline of a model that was fine-tuned on a dataset of product reviews [15] from the hugging face hub. We fine-tune the BERT model to perform sentiment analysis on this labeled dataset.

Figure 2 shows the summary of the proposed approach involving the text summarization and sentiment analysis phases.

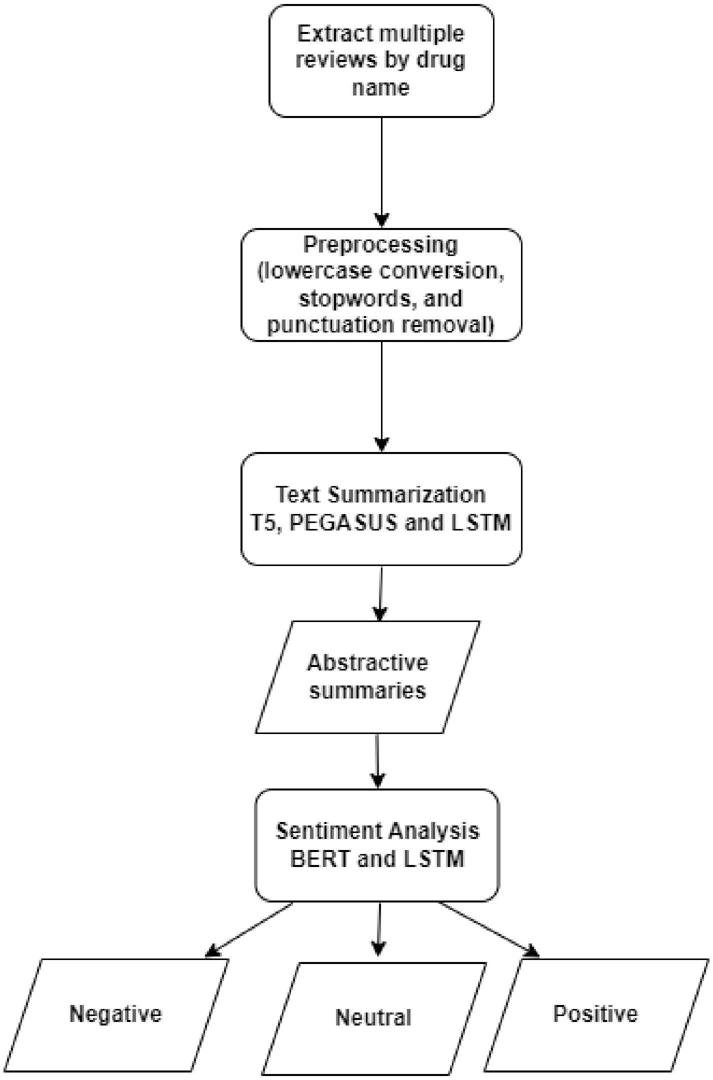

Extract multiple reviews by drug name
Preprocessing (lowercase conversion, stopwords, and punctuation removal)
Text Summarization T5, PEGASUS and LSTM
Abstractive summaries
Sentiment Analysis BERT and LSTM
Negative
Neutral
Positive

Figure 2: Overview of the proposed aproach

CHAPTER II.

BACKGROUND

Text Summarization

A majority of humans are naturally inclined to read short texts with rich information than lengthy ones. Considering how important and useful text summarization is, it has drawn the attention of many researchers over the years. Rule-Based Methods, Supervised Learning Algorithms, and Neural Network Architectures have been used for text summarization tasks. In the past, traditional rule-based methods [42] that rely on a set of rules to determine the importance of sentences to be included in the final summary were prevalent. The advent of deep learning architectures like the long short-term memory model (LSTM) and recurrent neural networks (RNN) [7] has also been useful for performing text summarization tasks. Joshi et al. [16] compared several deep learning models for text summarization and discovered that Convolutional Neural Network based approaches performed better with extractive summarization while RNN methods did well with abstractive summarization.

N. Yadav and N. Chatterjee [42] involved sentiment analysis in their text summarization process. Sentiment analysis helps to clearly distinguish between a positive or negative emotion in a text [21]. In [42], they performed sentiment analysis on the document understanding conferences (DUC) dataset. The scores from that process helped to determine the sentences to be part of the final summary text according to the specified number of the desired summary sentences. Their method performed better in recall values as compared with Random Indexing based summarizer and Latent Semantic Analysis based summarizer.

Shirwandkar et al. [32] used feature extraction methods to determine the score of a sentence to know if it should belong in the final summary. These methods include sentence position, sentence length, numerical token, term frequency-inverse document frequency, cosine similarity between sentence and centroid, bi-gram, tri-gram, and proper noun. They implemented the Restricted Boltzmann Machine (RBM) with fuzzy Logic to produce two different summaries of one document and combined the output from both according to a set of rules to form the final summary. Their method demonstrated a significant improvement over the use of RBM alone with an average precision, recall, and F-measure of 0.88, 0.80, and 0.84 respectively.

Krishnan et al.[19] also used feature extraction methods to determine sentence scores on the text in the dataset. They implemented supervised learning algorithms like naive Bayes, k-nearest neighbor (KNN), random forest, sequential minimal optimization (SMO), j48, and bagging on the BBC news summary data set to perform extractive text summarization. KNN, Bagging, and Random Forest performed better than SMO, Naive Bayes, and J48. The average precision ROUGE1, ROUGE2, and ROUGEL scores across all classifiers were 0.597, 0.470, and 0.583 respectively while that of the recall were 0.488, 0.368, and 0.477.

Sharaff et al. [31] proposed an extractive text summarization model using the bell membership function, triangular membership function, and fuzzy rule to determine sentences in the final summary. The evaluation parameters (precision, recall, and f-measure) of their model outperformed the other past approaches referenced in their paper.

Recently, the use of transformers has brought about a revolution in the Natural Language Processing field. Various researchers have fine-tuned pre-trained transformer models to attain state-of-the-art model performance. Authors Yang Liu and Mirella Lapata [26] performed abstractive and extractive text summarization on multiple datasets (CNN DailyMail, NYT, and Xsum). They proposed a novel model based on the BERT architecture for extractive summarization. For abstractive summarization, they fine-tuned the pre-trained BERTSUM model. The BERT-based model outperformed the LEAD-3 baselines and some other older approaches referenced in this article.

Khandelwal et al. [18] trained a unidirectional transformer-based Language model on a 2-billion-word corpus based on Wikipedia. This model uses one network for source encoding and target generation. They fine-tuned this model with an encoder-decoder architecture model to perform text summarization on the CNN/Daily Mail dataset. They evaluated their models with pre-training and without pre-training and their approach had significantly satisfactory results with the Language model achieving ROUGE1, ROUGE2, and ROUGEL scores of 39.65, 17.74, and 36.85 respectively.

Torres in [35] used the pre-trained BERTSUM model for extractive summarization of the CNN/Daily Mail dataset. BERTSUM is an extension of the BERT model with little variation. it can learn sentence representations and has the ability to embed pairs of sentences to learn adjacency patterns between them. The author used the LEAD-3 as a benchmark to compare with the fine-tuned model summaries and discovered that LEAD3 summaries were twice the length of the reference summary. The precision of the BERTSUM model was lower, but the recall was better.

Vinod et al. [38] also fine-tuned the BERTSUM model to perform text summarization on a dataset of clinical reports. In [38], the BERTSUM model which was trained initially on a corpus of news articles is further trained with specific strategies to improve performance on the medical dataset. They used @highlights in all target files, @highlights help specify those sentences of the report that are part of a good summary. A doctor who was consulted for human evaluation deemed 76.2% of the summaries generated as effective.

Zolotareva et al.[45] built a Seq2Seq model with LSTM layers for the encoder and decoder network based on the concept of the Text-to-Text transformer model to perform text summarization on a dataset of BBC news articles. To compare results, they also fine-tuned the Text-to-Text Transfer Transformer(T5) model on the same task. Their model performed well but the fine-tuned T5 model performed better with ROUGE1, ROUGE2, and ROUGEL scores of 0.473, 0.265, and 0.361 respectively.

Gupta et al. [13] fine-tuned various transformer architecture models like the T5[29], Bidirectional and Auto-Regressive Transformers (BART)[23], and Pre-training with Extracted Gap-sentences for Abstractive Summarization (PEGASUS)[43] and trained these models on the BBC news summary dataset. The T5 model outperformed the other models with ROUGE1, ROUGE2, and ROUGEL scores of 0.47, 0.33, and 0.42 respectively.

In the literature examined above, the summarization of drug reviews is lacking. With the advancement of medicine, drugs have become vital in saving lives and reducing the progression of numerous diseases. We present text summarization models trained on the University of California, Irvine (UCI) drug reviews dataset [8, 11].

Sentiment Analysis

Sentiment Analysis is an aspect of text processing that has been around for quite some time, the Bag-Of-Words model has been used extensively [24, 33, 37] along some supervised learning algorithms like Naïve Bayes, Maximum Entropy, Support Vector Machines, and K-Nearest Neighbor (KNN) [17, 21, 24, 33].

Li Menghzhe et al. [24] did sentiment analysis on a dataset of some Reddit comments from the transgender community to enable the health care professional to have an insight into the mental health of such individuals. They used the Bag-Of-Words model for tokenization and TF_IDF for weight adjustment. They built machine learning classifiers like Naive Bayes, Random Forest, Support Vector Machine, Logistic Regression, K-Nearest Neighbour, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). Their CNN and LSTM models outperformed other models with an accuracy of 0.861 and 0.852 respectively.

The use of the hybrid feature extraction method has also helped to improve the efficiency of some supervised learning algorithms as seen with K. Srujan et al. [33] on a dataset of Amazon book reviews of eight popular books. They transformed their preprocessed data into a Term Document Matrix (TDM), after which they implemented the Bag of words and Term Frequency-Inverse Document Frequency (TF-IDF). For the classification task, they implemented classifiers like K-Nearest Neighbours (KNN), Random Forest (RF), Naive Bayes (NB), Decision Tree, and Support Vector Machine (SVM). The Random Forest model outperformed other classifiers in about six of the eight books while the K-Nearest Neighbours model outperformed others in the remaining two books.

Kumar et al in [21] used a hybrid feature extraction method by combining machine learning and lexicon-based feature extraction methods on a dataset of IMDb Movie Reviews. Feature selection was done with Information Gain, Correlation, Chi-Square, and RLPI. They implemented supervised learning algorithms like Naïve Bayes, Maximum Entropy, Support, Vector Machines, and K- Nearest Neighbor (KNN) for the sentiment analysis classification task. The Maximum Entropy classifier outperformed all other classifiers in this experiment.

Kumar and Haris [17] explored the effects of preprocessing techniques like URL removal, username replacement with white space, hashtag removal, handle negation, characters normalization, punctuation removal, stop word elimination, and stemming on the Stanford Twitter Sentiment Dataset. For sentiment classification, they used support vector machine (SVM) and K-nearest neighbor (KNN) models. The accuracy of the models increased as a result of the preprocessing with the unigram and bigram representations.

Research has also gone into comparing some deep learning architectures to Transformer based models like the Bidirectional Encoder representation from Transformers (BERT) [6] as done in [4]. Colon-Ruiz et al. [4] did a classification with the UCI drug review dataset, their research compared the performance of different deep learning architectures on the sentiment analysis. They built models like CNN, bidirectional LSTM, and different variations of hybrid models made up of CNN and bidirectional LSTM, and BERT with an LSTM classifier. They found that the hybrid BiLSTM with CNN obtained the best result.

Alaparthi and Mishra in [2] compared the performance of Sent WordNet lexicon, logistic regression, LSTM, and BERT models for sentiment analysis on a dataset of 50,000 IMDB movie reviews. They found that the BERT model outperformed the other models with an Accuracy and F1-score of 0.9231.

P. Ns and K.[28] built models to perform sentiment analysis on the UCI drug review dataset. The Neural Network Models used ELMO Word Embeddings as input compared with different Transformers-based models such as BERT, XLNET, Bio_Clinical BERT, and ConvBERT. The ConvBert model outperformed all others with an accuracy of 94.3

Although the UCI drug review dataset [8, 11] has been used for sentiment analysis by some of the researchers [28, 4], none of them aforementioned performed text summarization before the sentiment analysis process. In this study, We performed sentiment analysis on the summaries of drug reviews by fine-tuning the Bidirectional Encoder Representation from Transformers (BERT) model [6].

CHAPTER III.

OVERVIEW OF MODELS

Bi-directional Encoder Representation from Transformers (BERT)

BERT [6] is a transformer model that was released by the Google team in 2017. Unlike a regular transformer model with encoders and decoders, BERT is made up of a stack of encoders. This model overcomes the directional limitation of models like RNN and LSTM by learning the context of words in a bi-directional manner which makes it more efficient. Given a sentence like *”On my way to the **bank** for a transaction, I saw a bird at the **bank** of a river”*, this model has the capability to distinguish the context in which the two occurrences of ”bank” was used. It combines three different embeddings (token embeddings, segment embedding, and positional embedding) as input. BERT was trained using the concept of Mask Language Modeling alongside Next Sentence Prediction [6]. It was pre-trained on large datasets like BooksCorpus (800M words) [44] and English Wikipedia data (2,500M words) [6]. It can be fine-tuned on tasks like question answering, sentiment analysis, and text summarization.

It comes in two different forms, BERT-large and BERT-base. The large model has 24 encoders, 16 attention heads, 1024 hidden layers, and about 340 million parameters while the base model has 12 encoders, 12 attention heads, 768 hidden layers, and about 110 million parameters. Each of these two forms have the cased version and the uncased version.

Pegasus Model Overview

The PEGASUS model [43] is another model that was released by the Google research team with a special pre-training objective for abstractive text summarization. The architecture of this model is that of a standard transformer model made up of encoders and decoders. They used the concept of Gap Sentence Generation together with Masked Language Modeling and trained the model on the C4 dataset (750 GB) introduced in [29] and the Hugenews dataset (3.8 TB). The hugenews dataset is a dataset of about 1.5 billion articles collected from news-like websites from 2013 to 2019. Unlike the Masked Language modeling strategy in models like BERT [6] where individual tokens are masked, PEGASUS proposed a new pre-training objective that is better aligned with the downstream task of abstractive text summarization. This objective uses gap sentence generation where you mask some fraction of sentences and the transformer predicts the masked sentences.

The model is of two variants known as the PEGASUS Base and PEGASUS Large. The base model has 12 encoder and decoder layers, 768 hidden sizes, 3072 feed-forward layers, and 12 attention heads while the PEGASUS Large model has 16 encoder and decoder layers, 1024 hidden size, 4096 feed-forward layers, and 16 attention heads. [43]. The base model has about 223 million parameters while the large model has 565 million parameters.

Text-to-Text Transfer Transformer (T5) model

Raffel et al.[29] proposed a transformer-based sequence-to-sequence model. It is an encoder-decoder model architecture trained in a unified objective manner that models every problem in a text-to-text format. The input and output are both in text format. The stack of encoders (each made up of a self-attention layer and a feed-forward network) takes input as a sequence of tokens that maps to a sequence of embedding [13, 29]. The decoder's architecture is like that of the encoder but has a standard attention mechanism after every self-attention layer. The output of the last decoder passes into a dense layer with a softmax activation function, and the weights are shared with the input embedding matrix [29].

The model was trained on the colossal clean crawled corpus(C4) dataset, which is about 700GB and can be fine-tuned for tasks like summarization, classification, translation, and question answering. The model works by receiving a text input containing the desired task to be performed as the prefix, and it produces an output in text format. The T5 model has 5 variants which are the T5-small, T5-base, T5-large, T5-3B, and T5-11B. The T5-small model's architecture consists of 6 layers in each encoder and decoder, 8 attention heads, and about 60 million parameters while the base version has about 220 million parameters [29].

Long Short-Term Memory (LSTM)

An LSTM model is a special type of Recurrent Neural Network (RNN) model with the ability to remember long-term dependencies and this property addresses the exploding gradient problems of a regular RNN [39]. They were introduced by Hochreiter and Schmidhuber (1997) [14] and have been very effective in handling many Natural Language Processing tasks [39]. The architecture of the LSTM model is made up of three gates namely; the input gate, the output gate, and the memory gate. These gates help to avoid or resolve the input and output weight conflicts and regulate the flow of error into the input unit and out of the output cell states [14]. Since LSTMs have proven to be quite effective for text-processing tasks, we decided to compare the results from our fine-tuned models to the performance of LSTM models.

CHAPTER IV.

METHODS

Dataset

We used the University of California, Irvine (UCI) drug reviews dataset that was introduced by Gräßer et al. [8, 11]. It was crawled from pharmaceutical websites like drugs.com and druglib.com. This dataset has been widely used for research studies in sentiment analysis [4, 11, 28]. It contains about 215,063 records with six attributes (drug name, condition, useful count, review text, date collected, and rating). Figure 3 shows the word cloud representation of words in the dataset. Word Cloud is a text visualization method in python where the size of the word is directly proportional to its frequency or importance.

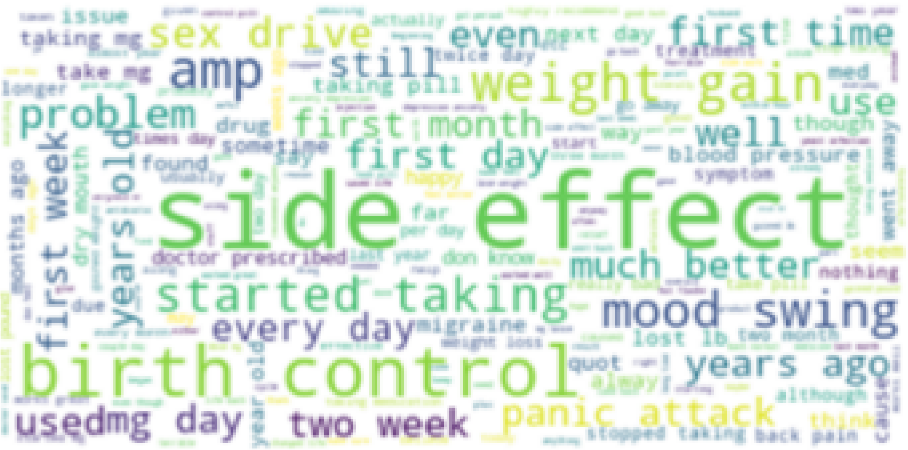


Figure 3: Word cloud representation of words in the UCI drug reviews dataset.

Data Preprocessing

We worked with the ten most useful reviews for the first 500 drug names(after sorting the drug names alphabetically) in the dataset. We extracted these ten reviews based on the useful counts for each drug name and combined the review text for summarization. For each drug name, we manually created human summaries for the combined reviews to be presented to the model. We performed the following preprocessing steps on the review and target summary text.

Conversion to lowercase letters: We converted all review text and target summary text to lowercase letters.

Removal of punctuation: We removed all punctuation and special characters, excluding periods. This is because periods indicate the end of a sentence and will be important in the summarization process.

Removal of stop-words: Stop words are a set of commonly used words in a language that does not add to the overall meaning of a sentence. Examples of some stop words in English are: “a”, “the”, “is”, “are” Et cetera.

For the summarization task with the T5 model, we added the keyword “summarize” as a prefix to all the reviews to specify the intended task. We took an average of the review text and the target summary, which informed our choice of a maximum input length of 2048 and a maximum target length of 128 words for padding and truncation as needed. For the T5-small model, the maximum input length of 2048 executed well without any issues but for the larger models (T5-base and PEGASUS-base), the maximum input length had to be adjusted. The batch size and maximum input lengths were reduced to 2 and 1024 respectively. Aside from those two changes, every other hyperparameter remains the same as seen in Table 1.

All experiments were performed on Google Colab Tesla T4 graphics processing unit (GPU). From Figure 4 and word count analysis, 95.4% of the review texts are less than or equal to 1024 while 98.8% of the summary texts are less than or equal to 128 for the drug reviews dataset. This validates our decision to reduce input lengths to accommodate experiments with larger models. We split the datasets into 80% for training, 10% for testing, and 10% for validation.

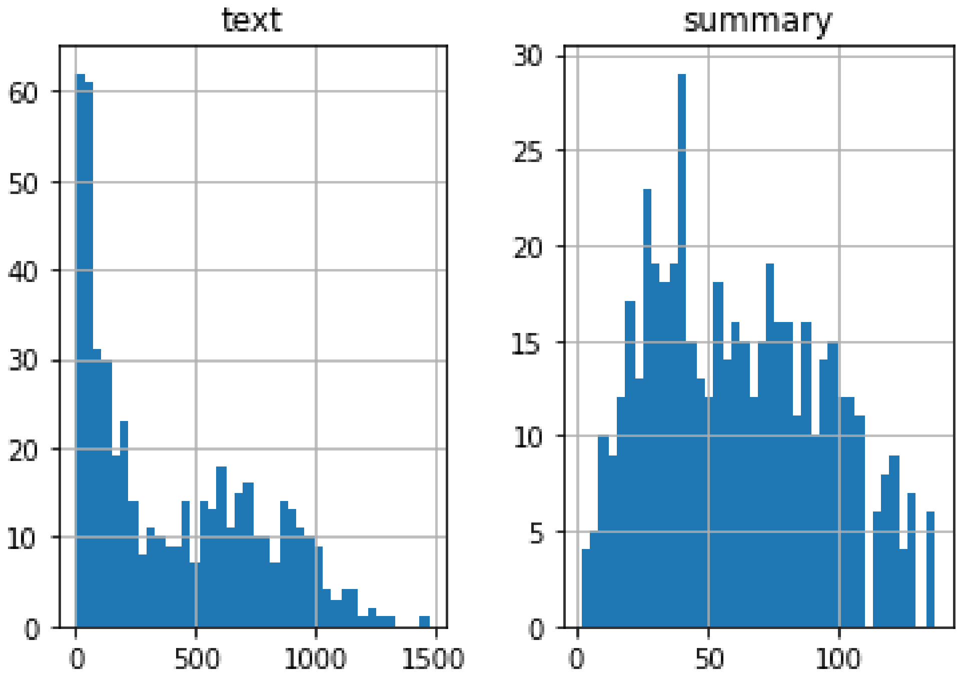


Figure 4: Word count plots for the review text and summary for the UCI drug reviews dataset

Fine-Tuning of the Pre-trained Transformer Models for Text Summarization

For the text summarization task, we fine-tuned the T5 and pegasus models. Our initial experiment was with the T5-small variant [30]. For this experiment, the preprocessing steps involved were conversion to lowercase, removal of special characters, and stop words. Due to a lack of adequate computational resources, there is little variation in the batch size and the maximum input length for the first experiment and the subsequent ones as done in [22]. In each case, we created instances of the seq2seq model from the Hugging Face's transformers library [41] and trained with the parameters in Table 1.

In Table 2, we briefly describe the hyperparameters listed in Table 1. We also examined the effects of the various preprocessing methods mentioned above on the ROUGE scores from the text summarization process.

To compare results from the pre-trained transformer models, we built an LSTM sequence-to-sequence (seq2seq) model [45]. This model's architecture consists of encoders and a

Table 1: List of hyperparameters and values

Hyperparameter	**Value**
evaluation strategy	epoch
learning rate	0.0005
per device train batch size	4
per device eval batch size	4
weight decay	0.001
save total limits	3
num train epochs	5
predict with generate	True
fp16	True

decoder, it takes in a sequence of text as input and produces an output of a different sequence length.The encoder takes in a sequence of text, processes it in each time step, and learns the context of words while the decoder takes in the target sequence of words and tries to predict the next word with the given previous word. The LSTM model has 4 stacked LSTM encoder layers (with a dropout size of 0.2), a decoder layer with a softmax activation function, and an attention layer. We used the “rmsprop” optimizer, batch size of 8 and trained for 5 epochs just like the transformer models.

Table 2: Explanation of the list of hyperparameters

Hyperparameter	**Description**
evaluation strategy	specifies the mode in which evaluation is done during training. it can be any of none, steps, or epochs.
learning rate	specifies the desired learning rate which determines the level of weight updates during training.
per_device_train_batch_size	refers to the batch size per GPU/TPU Core/CPU for training.
per_device_train_eval_size	refers to the batch size per GPU/TPU Core/CPU for evaluation.
weight decay	specifies the weight decay value to be applied to all layers excluding the bias and layerNorm weights.
save total limits	helps to regulate the number of checkpoints to be saved.
num train epochs	refers to the number of passes the training dataset makes through the model.
predict with generate	helps to generate summaries and enables mixed precision training.
fp16	specifies the use of fp16 16bits precision training as opposed to the default of 32bits to help speed up training.

Sentiment Analysis Models (Fine-tuned BERT and the LSTM Model

We labeled the sentiment scores of each of the 500 summaries with a BERT based-model from the hugging face hub. This model was trained on a dataset of product reviews [15] (URL: https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment). It assigns sentiment scores between 1 and 5 with 5 being the most positive and 1 being the most negative. We regrouped scores between 4 and 5 as positive, a score of 3 as neutral, and a score of 2 or less as negative. Figure 5 shows the distribution of the summaries after regrouping, which indicates an imbalance in the resulting dataset.

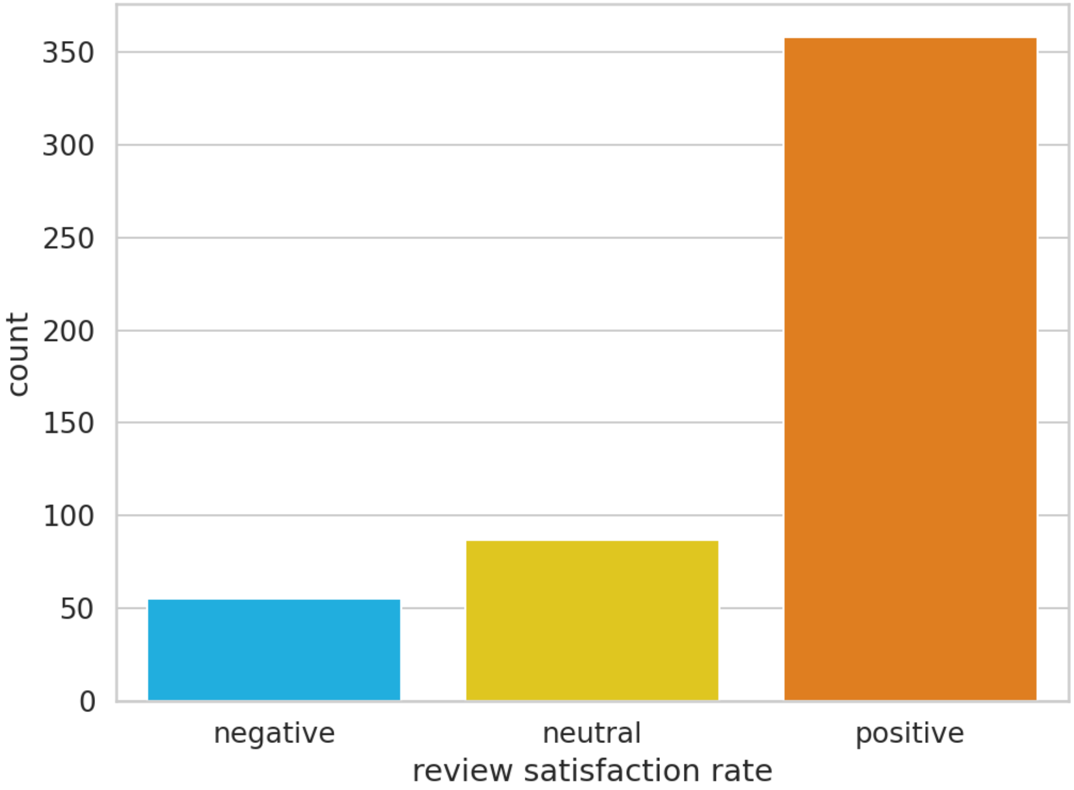


Figure 5: Distribution of the summaries after regrouping

We added special tokens like the [CLS] tokens to indicate a classification task, [SEP] token to specify the end of a sentence, and the [PAD] token to fill up sentences that are less than the maximum length specified making sure that all inputs are of equal length. After this, we passed this input to the BERT tokenizer to prepare our input for the model.The token embedding assigns different tokens to individual words in the sentence, the positional embedding specifies the position of a word in the sentence while the segment embedding specifies which sentence a word belongs to in the text. The output from the summation of these three different embeddings is then passed into the stack of encoders which will produce an output of vectors of the hidden layer size which is then passed to the classifier. We used the BERT base uncased version from the Hugging Face's transformers library [41] with a batch size of 16, Maximum input length of 256(for the BERT base model), and trained for 5 epochs.

For the LSTM model, the preprocessed data is passed to an embedding layer to transform the data into a set of vectors that can be understood by the model, then it is passed to a spatial dropout layer to encourage regularization and prevent over-fitting. The spatial dropout layer is followed by two LSTM layers of 32 units each and has dropout and recurrent dropout values of 0.2. This is followed by a dense layer that uses the softMax activation function, which categorizes the text to either of the negative, neutral, or positive class.

CHAPTER V.

RESULTS AND DISCUSSIONS

Overview of Evaluation Metrics

We used the ROUGE metrics (Recall-Oriented Understudy for Gisting Evaluation) [25] to evaluate the text summarization models. This evaluation metric helps to assess the quality of a model-generated summary by comparing it with a corresponding human summary. We specifically worked with the ROUGE1, ROUGE2, and ROUGEL scores. Table 3 gives an overview of these ROUGE metrics.

Table 3: Evaluation metrics

Name	**Description**
ROUGE1	calculates the unigram overlap between the model-generated summaries and a set of reference human-generated summaries.[25]
ROUGE2	calculates the bigram overlap between the model-generated summaries and a set of reference human-generated summaries.[25]
ROUGEL	calculates the longest common sub-sequence overlap between the model-generated summaries and a set of reference human-generated summaries.[25]

These scores are calculated with the recall and precision values to get the F-measure score which is the corresponding ROUGE score [25].

Let NOV = Number Of Overlapping Words

Let HS = Total Number Of Words In Human Summary

Let MS = Total Number Of Words In Model Generated Summary

$$Recall = NOV/HS$$

$$Precision = NOV/MS$$

$$ROUGEScore = (2 * precision * Recall)/(precision + Recall)$$

For the sentiment analysis models, we evaluated the results with accuracy and the F1-score metrics. The F1-scores will give us further insights into how the model performed in each of the classes and this is important considering our imbalanced dataset.
Accuracy is calculated as the number of correct predictions divided by the total number of samples in the test dataset while the F1-score is explained in the equations below. True positives are the data points that were correctly classified while False negatives are those that were wrongly classified.

$$Recall = (TruePositive)/(TruePositive + FalseNegative)$$

$$Precision = (TruePositive)/(TruePositive + Falsepositive)$$

$$F1Score = 2 * (recall * precision)/(recall + precision)$$

T5-Small

Since we manually summarized the drug review dataset, we tested our model on the BBC news dataset [12] for better evaluation [30]. This dataset consists of 2225 documents from the BBC news website in 5 topic areas from 2004 - 2005. These topic areas are business, entertainment, politics, sports, and tech. Table 4 shows the average of the ROUGE scores obtained from the test sets on the UCI drug review dataset [8, 11] and the BBC news dataset [12].

The training process of deep learning models generally includes pseudo-random numbers (like initial weights and random mini-batch training), we ran all the models multiple times under the same conditions to obtain the average ROUGE scores. Table 4 shows the average ROUGE scores from the fine-tuned T5 small [30] model for both the Drug Reviews and the BBC datasets.

Table 4: Average ROUGE scores for the T5-small model

Dataset	**ROUGE1**	**ROUGE2**	**ROUGEL**
Drug Reviews	45.62	25.58	36.53
BBC News	69.05	59.70	52.97

Statistical analysis for the T5-Small model

Table 5: UCI drug reviews dataset

ROUGE category	**Confidence interval**
ROUGE1	43.80 - 47.44
ROUGE2	22.86 - 28.30
ROUGEL	34.18 - 38.88

Table 6: BBC news dataset

ROUGE category	Confidence interval
ROUGE1	68.49 - 69.61
ROUGE2	59.35 - 60.04
ROUGEL	52.35 - 53.59

Table 5 and Table 6 show the confidence intervals for the experiments [30] in Table 4. At 95% confidence level, the true mean of the ROUGE scores of the population will lie within the values specified as confidence intervals. For all other confidence interval tables, the same 95% confidence level is used to calculate the confidence intervals for the average ROUGE scores.

T5-Base

For the T5-Base and Pegasus models, we examined the effects of various preprocessing steps on the final ROUGE score. STEP 1 represents Lowercase conversion, STEP 2 represents the combination of STEP 1 and stop words removal, and STEP 3 represents a combination of STEP 2 and the removal of special characters and punctuation. These steps are used interchangeably with the preprocessing operations they represent in the confidence interval tables.

Table 7: Average ROUGE scores(Drug reviews)

Preprocessing Steps	ROUGE1	ROUGE2	ROUGEL
Lower case	48.85	27.63	38.15
Lower case and stop words	50.31	29.14	40.06
Lower case, stop words and special characters	47.51	25.75	37.03

Table 8: Confidence interval(Drug reviews)

STEPS	ROUGE1	ROUGE2	ROUGEL
1	48.64 - 49.05	26.75 - 28.51	36.94 - 39.36
2	47.79 - 52.83	25.30 - 32.98	37.03 - 43.09
3	43.07 - 51.97	20.67 - 30.83	33.43 - 40.63

Table 9: Average ROUGE scores(BBC news)

Preprocessing Steps	ROUGE1	ROUGE2	ROUGEL
Lower case	70.04	60.96	55.27
Lower case and stop words	72.07	63.65	57.05
Lower case, stop words and special characters	72.20	63.59	57.42

Table 10: Confidence interval(BBC news)

STEPS	ROUGE1	ROUGE2	ROUGEL
1	68.26 - 71.82	58.29 - 63.63	52.59 - 57.95
2	71.29 - 72.85	62.75 - 64.55	56.34 - 57.76
3	70.02 - 74.38	60.91 - 66.27	55.85 - 58.99

The ROUGE scores from the experiment performed with the T5-Base in Table 7 and Table 9 outperformed that of T5-small in Table 4 regardless of the fact that the text was truncated more with the T5-Base model than it was in the previous experiment. The highest ROUGE scores in all categories for the T5-base model were obtained with the conversion to lowercase and removal of stop words as preprocessing steps. The drug reviews had ROUGE1, ROUGE2, and ROUGEL scores of 50.31, 29.14, and 40.06 respectively. The

BBC news dataset had best results in majority of the ROUGE categories in step 3 (all preprocessing steps) with ROUGE1, ROUGE2, and ROUGEL scores as 72.20, 63.59 , and 57.42 respectively. Table 8 and Table 10 contain the confidence intervals for average ROUGE scores.

The results from the UCI drug reviews dataset were not as impressive as those from the BBC news dataset because they had fewer training samples. Creating human summaries takes a lot of time and effort, making it difficult for us to use the entire drugs and reviews in the dataset.

On the BBC news dataset, Table 11 shows that our fine-tuned model's performance was an improvement over some of the papers cited in the background section [13, 45]. They also fine-tuned the T5 model for text summarization on BBC news dataset.

Table 11: Performance comparison on previous work on BBC news dataset

Author	**ROUGE1**	**ROUGE2**	**ROUGEL**
Gupta et al [13]	47.00	33.00	42.00
Zolotareva et al [45]	47.30	26.50	36.10
Our work	72.20	63.59	57.42

Pegasus Base

Table 12: Average ROUGE scores (Drug reviews)

Preprocessing Steps	**ROUGE1**	**ROUGE2**	**ROUGEL**
Lower case	46.69	24.23	35.66
Lower case and stop words	43.99	23.25	35.34
Lower case, stop words and special characters	43..18	21.40	32.87

Table 13: Confidence interval(Drug reviews)

STEPS	ROUGE1	ROUGE2	ROUGEL
1	42.91 - 50.47	20.50 - 27.96	31.73 - 39. 59
2	41.72 - 46.26	20.69 - 25.81	31.82 - 38.86
3	40.45 - 45.91	18.23 - 24.57	29.48 - 36.26

Table 14: Average ROUGE scores (BBC news)

Preprocessing Steps	ROUGE1	ROUGE2	ROUGEL
Lower case	66.51	60.38	51.01
Lower case and stop words	65.53	59.57	50.29
Lower case, stop words and special characters	66.62	60.47	49.69

Table 15: Confidence interval(BBC news)

STEPS	ROUGE1	ROUGE2	ROUGEL
1	64.83 - 68.19	58.75-62.01	50.48 - 51.54
2	64.07 - 66.69	58.14 - 61.00	48.52 - 52.06
3	66.07 - 67.16	59.77 - 61.17	48.71 - 50.66

Table 12 and Table 14 show the results of the experiment with the Pegasus-base model for the drug reviews and BBC news datasets respectively. The best results for the drug reviews were obtained with only conversion to lowercase with The ROUGE1, ROUGE2, and ROUGEL scores are 46.69, 24.23, and 35.66 respectively. For the BBC news dataset, it repeated the pattern noticed from the T5-base model where the best in majority of the categories came from step 3 which included all three preprocessing steps. The ROUGE1, ROUGE2, and ROUGEL scores are 66.62, 60.47, and 49.69 respectively. Table 13 and Table 15 show the confidence interval for the average ROUGE scores from the PEGASUS-base model.

LSTM for Text Summarization

Table 16: Average ROUGE scores(Drug reviews)

Preprocessing Steps	ROUGE1	ROUGE2	ROUGEL
Lower case	22.79	3.05	18.31
Lower case and stop words	24.37	4.32	19.66
Lower case, stop words and special characters	25.28	4.85	20.07

Table 17: Confidence interval(Drug reviews)

STEPS	ROUGE1	ROUGE2	ROUGEL
1	19.93 - 25.65	2.47 - 3.63	15.42 - 21.20
2	19.96 - 28.78	1.91 - 6.73	16.47 - 22.85
3	21.06 - 29.05	3.64 - 6.06	16.95 - 23. 19

Table 18: Average ROUGE scores(BBC news)

Preprocessing Steps	ROUGE1	ROUGE2	ROUGEL
Lower case	6.13	1.35	5.82
Lower case and stop words	12.49	1.95	10.49
Lower case, stop words and special characters	11.73	2.12	9.31

Table 19: Confidence interval(BBC news)

STEPS	**ROUGE1**	**ROUGE2**	**ROUGEL**
1	5.24 - 7.02	1.20 - 1.50	4.97 - 6.67
2	10.54 - 14.44	1.81 - 2.09	6.66 - 14.32
3	8.31 - 15.15	1.47 - 2.77	7.00 - 11.62

As expected, The LSTM model did not perform as well as the others regardless of its encoder-decoder architecture. This is because it had to learn from scratch without any prior domain knowledge and with very few data points. The best results for the drug reviews were ROUGE1, ROUGE2, and ROUGEL of 25.28, 4.85, and 20.07 respectively as shown in Table 16. These results were obtained with all three (conversion to lowercase, removal of stop words, and removals of special characters) preprocessing steps. Table 17 shows the confidence interval of the average ROUGE scores from the drug reviews dataset.

The results from the BBC news dataset as shown in Table 18 were not as good as expected compared to the drug reviews. Although it had more data points to train with, the test data also had more data points than the test data of the drug reviews dataset. This may also be as a result of a slight change in model architecture. The model architecture had 3 encoders had instead of the 4 encoders used for the Drug reviews Dataset to accommodate computational resources. The best ROUGE1 and ROUGEL scores were 12.49 and 10.49 where the preprocessing step involved conversion to lowercase and removal of stop words. The best ROUGE2 score was 2.12 achieved with all three preprocessing steps. Table 19 shows the confidence interval for the average ROUGE scores in table 18.

Given the human-generated summary ***"Treanda Is a chemotherapy medication that works well to treat symptoms of CLL. It has helped to shrink tumors and help get some users into remission after a few rounds of treatment. It has some unpleasant side effects as reported by past users. These side effects are nausea, constipation, dry and flaky skin,***

***negative effects on liver function, loss of appetite, and weight loss. Some users had to stop the treatment because of these negative side effects.*”** from the drug reviews dataset, Table 20 shows an example summary of each model. The LSTM model could not generate any meaningful summary after training for 5 epochs like the transformer models. We tried to improve its generated summaries by training for for more epochs than we did for the fine-tuned models. From Table 20 and table 21, we can see that the fine-tuned T5 and Pegasus models produced meaningful summaries but the LSTM model was not so good because it trained from scratch with insufficient data points.

Table 20: Example summaries (Drug reviews)

Model	**Example**
Pegasus	Treanda is a great medication for treating symptoms of cancer. it helps to eliminate the symptoms within a few days of use and gets them to remission in some cases. some people use it in combination with chemotherapy and get satisfactory results while some used it in combination with another cancer medication. reported side effects are nausea, constipation, dry flaky skin, liver function being negatively affected (recovered 3 or so months after treatment), bone marrow cellularity going from 65% to 20% which is slowly recovering.
T5	Treanda works well to treat symptoms of CLL. it works well to reduce tumors and get users into remission after a few rounds of treatment. reported side effects are nausea, constipation, dry and flaky skin, negative effects on liver function, loss of appetite, and difficulty walking.
LSTM	is an effective medication for treating symptoms of it has helped users to relieve the of of it has helped users to reduce the and it has helped users to no side effects that have made other medications medications perform similar functions it is not expensive.

Consider the human-generated summary ***"if admitted into the eu turkey would contribute almost 6bn euros $8bn £6bn to its budget by 2014 according to a recent impact study by the country's state planning organisation.critics insist that much of the optimism***

about turkey’s economic road map has been over-egged - an argument amplified by a 134% rise in the country’s current account deficit to $10.7bn during the first 10 months of this year.as turkey’s gross domestic output gdp is set to grow by 6% per year on average its contribution would rise from less than 5bn euros in 2014 to almost 9bn euros by 2020.three years after a gruelling economic crisis turkey has dressed its economy to impress.however turkey’s assertions are confirmed by brussels’ own impact studies which indeed say that turkish membership would be good news for the eu economy.a key factor for eu decision makers should be whether or not turkey has met its economic criteria.the point of all this has been to convince europe’s decision makers that rather than being a phenomenally costly exercise for the eu allowing turkey in would in fact bring masses of economic benefits.turkey will not be a burden for the eu budget. the country’s massive debt - which includes $23bn owed to the imf and billions borrowed via the international bond markets - also remains a major obstacle to its ambition of joining the eu.costs are projected to be vast during the early years of turkey’s membership with subsidies alone estimated to exceed 16.5bn euros and according to some predictions balloon to 33.5bn euros. at about 80% of gdp turkey’s gross debt is double that figure. turkey’s debts have largely arisen from its efforts to push through banking reform after a run on the banks in 2001 caused the country’s devastating recession.the tourism industry is booming and revenues from visitors should more than double to $21bn £10.8bn in three years.in the new member states of the european union gross public debt is typically about 40% of gross domestic product says reza moghadam assistant director of the imf’s european department.such high initial expenses would be coupled with risks that the benefits flagged up by turkey’s government would never be delivered say those who feel the turkish project should be shunned.and although the state of turkey’s economy is important as is its pace of reform the final decision on 17 december will be taken by politicians who will of course be guided by their political instincts.” from the BBC news dataset.

Table 21: Example summaries (BBC news)

Model	**Example**
Pegasus	as turkey's gross domestic output gdp is set to grow by 6% per year on average its contribution would rise from less than 5bn euros in 2014 to almost 9bn euros by 2020 turkey could also help alleviate a labour shortage in "old europe" once its population comes of age.if admitted into the eu turkey would contribute almost 6bn euros $8bn £6bn to its budget by 2014 according to a recent impact study by the country's state planning organisation."there is no question that although turkey is doing much better than in the past it remains quite vulnerable " says michael deppler director of the imf's european'.
T5	if admitted into the eu, turkey would contribute almost 6bn euros ($8bn; â£6bn) to its budget by 2014, according to a recent impact study by the country's state planning organisation.critics insist that much of the optimism about turkey's economic road map has been over-egged - an argument amplified by a 134% rise in the country's current account deficit to $10.7bn during the first 10 months of this year.as turkey's gross domestic output (gdp) is set to grow by 6% per".
LSTM	the uk and the uk was the of the uk and the uk the government was a of the uk and the government is to be to be the of the uk and the government is to be to be a lot of the uk and the government is to be to be to the first of the world of the world and the government is to be to be to the way to the government to the government to be the government to be to be to be a of the way to be to be to be to be a lot of the way to be to be a lot of the way to be to be a lot of the way.

Table 21 shows an example summary produced by each model for the BBC news dataset.

Sentiment Analysis (BERT and LSTM)

Table 23 and Table 22 show the F1-scores from the fine-tuned BERT model with and without the preprocessing steps respectively. The model achieved an accuracy of 74% with the preprocessing steps and 84% without those steps. This could be because the removal of stop words may sometimes change the meaning of the text thereby affecting the results.

Table 22: The F1-scores for the BERT model with the preprocessing steps

Class	**F1-Score**
Negative	0.57
Neutral	0.50
Positive	0.84

Table 23: The F1-scores for the BERT model without the preprocessing steps

Class	**F1-Score**
Negative	0.57
Neutral	0.67
Positive	0.93

The LSTM model ran with and without the preprocessing steps and achieved an accuracy of 66% and 60% respectively. In contrast to the BERT model, the LSTM model seem to have benefited more from the preprocessing steps as evident in the accuracy values and the F1-scores shown in Table 24 and Table 25. The models performed best in the Positive class, followed by the neutral class, but performed really poorly in the negative. This is because of the availability of more data points to learn from in the positive and neutral classes than they were in the negative class.

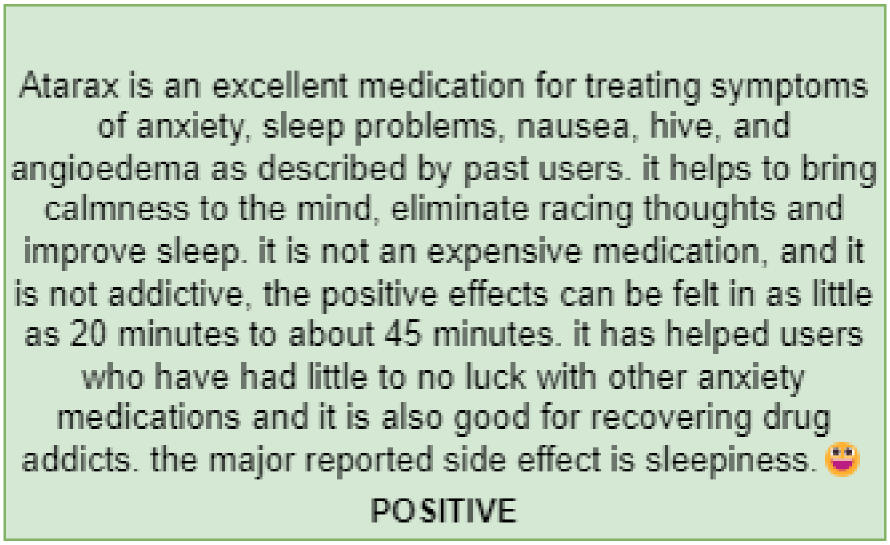
Table 24: The F1-scores for the LSTM model with the preprocessing steps

Class	**F1-Score**	**Confidence interval**
Negative	0.11	Not applicable
Neutral	0.31	0.29 - 0.33
Positive	0.78	0.76 - 0.81

Table 25: The F1-scores for the LSTM model without the preprocessing steps

Class	**F1-Score**	**Confidence interval**
Negative	0	Not applicable
Neutral	0.31	0.18 - 0.44
Positive	0.74	0.69 - 0.78

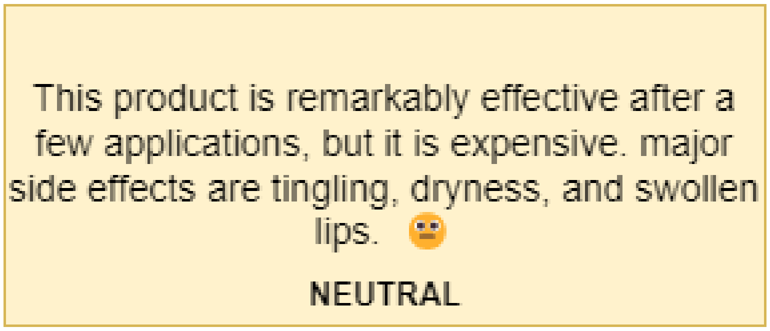
Figure 6, Figure 7, and Figure 8 show examples of positive, neutral and negative sentiment classification results from the fine-tuned BERT model.



Atarax is an excellent medication for treating symptoms of anxiety, sleep problems, nausea, hive, and angioedema as described by past users. it helps to bring calmness to the mind, eliminate racing thoughts and improve sleep. it is not an expensive medication, and it is not addictive, the positive effects can be felt in as little as 20 minutes to about 45 minutes. it has helped users who have had little to no luck with other anxiety medications and it is also good for recovering drug addicts. the major reported side effect is sleepiness. 😀

POSITIVE

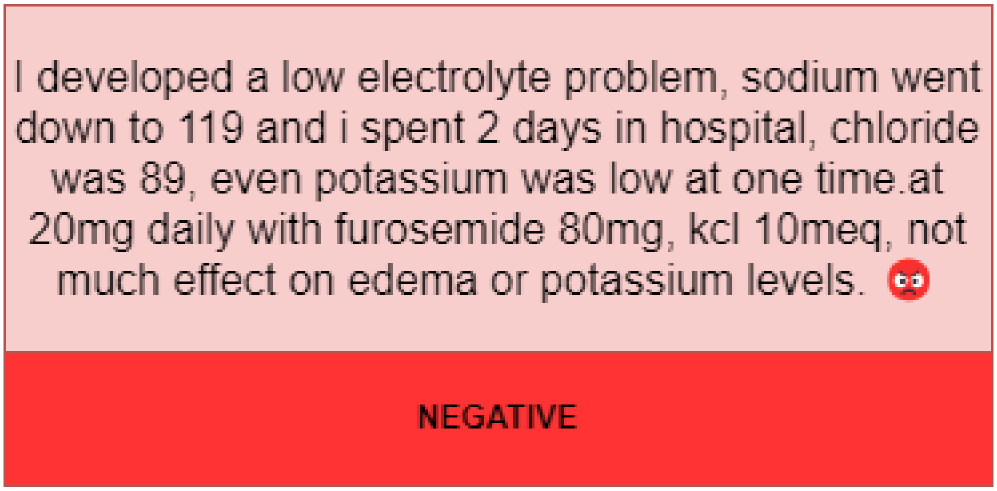
Figure 6: Positive sentiment



This product is remarkably effective after a few applications, but it is expensive. major side effects are tingling, dryness, and swollen lips. 😐

NEUTRAL

Figure 7: Neutral sentiment



I developed a low electrolyte problem, sodium went down to 119 and i spent 2 days in hospital, chloride was 89, even potassium was low at one time.at 20mg daily with furosemide 80mg, kcl 10meq, not much effect on edema or potassium levels. 😡

NEGATIVE

Figure 8: Negative sentiment

Limitations

The major challenges encountered during this project were the lack of adequate processing power for all desired experiments and insufficient data points. We could not experiment with larger versions of the transformer models because the lack of adequate computational power. To effectively evaluate the quality of model-generated summaries, there is a need for human-generated summaries to serve as the basis for comparison. This process requires so many hours of work, this limitation compelled us to use a subset of the drug reviews dataset rather than the entire dataset . More data points will guarantee better results in an ideal situation as seen with the BBC news dataset (On the fine-tuned transformer models) in comparison to the drug reviews.

CHAPTER VI.

CONCLUSIONS

Text summarization and sentiment analysis have remained pivotal areas in the Natural Language Processing field for the useful insights they provide from the analysis of text data. Text summarization methods have been used to reduce large text into a condensed but meaningful form while sentiment analysis has been used to mine emotions from text data for some decades now. These methods have evolved over time to produce better results. As information availability increases, it creates a need for better methods for analysis but drug reviews have gotten less attention than news articles and the like.

In this study, we created human summaries for 500 drugs from the UCI drug reviews dataset. We fine-tuned the T5-Small, T5-Base, and Pegasus-Base models to generate summaries automatically from the 10 most useful reviews for each of those 500 drugs and conducted sentiment analysis on the summaries using BERT and LSTM. We also evaluated the impact of three text preprocessing steps on the ROUGE Scores of these models. For comparison, we built an encoder-decoder LSTM model for Text summarization. The T5-Base model had the best results with average ROUGE1, ROUGE2, and ROUGEL scores of 50.31, 29.14, and 40.06 respectively for the Drug Reviews dataset and ROUGE1, ROUGE2, and ROUGEL scores of 72.20, 63.59, and 57.42 respectively for the BBC News dataset in the text summarization task.

The fine-tuned BERT model for sentiment analysis had the best performance with an accuracy of 84%. The regular deep learning model (LSTM) used benefited more from the preprocessing steps than the transformer models in most cases. With these results, we have been able to demonstrate the power of transfer learning and the effects of some preprocessing steps on the results gotten from text summarization and sentiment analysis.

These models will help potential users of these drugs to have access to a concise form of useful information and the emotion present in these summaries of the drug of interest from past users. It will help them make informed choices about these drugs from a larger audience of users beyond the spheres of clinical trials. When an intending user visits a website where these models are deployed, instead of having to search through so many reviews, they can save time by just looking at the summaries from the reviews provided by real-life users of the drug of interest.

In the future, we intend to explore automatic ways to produce ground truth summaries for large datasets and more effective evaluation metrics that may not require human summaries. This evaluation metric will reduce the dependency on human-generated summaries thereby increasing the availability of datasets for text summarization research.

BIBLIOGRAPHY

[1] Ajitesh Kumar,. Sentiment analysis & machine learning techniques, 2021. `https://vitalflux.com/sentiment-analysis-machine-learning-techniques/`, Last accessed on 2023-01-16.

[2] Alaparthi, S. and Mishra, M. Bidirectional encoder representations from transformers (bert): A sentiment analysis odyssey. *arXiv preprint arXiv:2007.01127*, 2020.

[3] Chopra, S., Auli, M., and Rush, A. M. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 93–98, 2016.

[4] Colón-Ruiz, C. and Segura-Bedmar, I. Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics*, 110:pages 103539, 2020.

[5] Daumé III, H. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[6] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Doğan, E. and Kaya, B. Deep learning based sentiment analysis and text summarization in social networks. In *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–6. IEEE, 2019.

[8] Dua, D. and Graff, C. Uci machine learning repository, 2017. *URL http://archive. ics. uci. edu/ml*, 7(1), 2017.

[9] Duan, L., Xu, D., and Tsang, I. Learning with augmented features for heterogeneous domain adaptation. *arXiv preprint arXiv:1206.4660*, 2012.

[10] Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[11] Gräßer, F., Kallumadi, S., Malberg, H., and Zaunseder, S. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health*, pages 121–125, 2018.

[12] Greene, D. and Cunningham, P. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384, 2006.

[13] Gupta, A., Chugh, D., Katarya, R., and others,. Automated news summarization using transformers. In *Sustainable Advanced Computing*, pages 249–259. Springer, 2022.

[14] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):pages 1735–1780, 1997.

[15] Hugging Face Hub model,. A pre-trained model from hugging face hub, 2023. `https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment`, Last accessed on 2023-01-14.

[16] Joshi, A., Fidalgo, E., Alegre, E., and de León, U. Deep learning based text summarization: approaches, databases and evaluation measures. In *International Conference of Applications of Intelligent Systems*, 2018.

[17] Keerthi Kumar, H. and Harish, B. Classification of short text using various preprocessing techniques: An empirical evaluation. In *Recent findings in intelligent computing techniques*, pages 19–30. Springer, 2018.

[18] Khandelwal, U., Clark, K., Jurafsky, D., and Kaiser, L. Sample efficient text summarization using a single pre-trained transformer. *arXiv preprint arXiv:1905.08836*, 2019.

[19] Krishnan, D., Bharathy, P., Venugopalan, M., and others,. A supervised approach for extractive text summarization using minimal robust features. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 521–527. IEEE, 2019.

[20] Kulis, B., Saenko, K., and Darrell, T. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR 2011*, pages 1785–1792. IEEE, 2011.

[21] Kumar, H., Harish, B., and Darshan, H. Sentiment analysis on imdb movie reviews using hybrid feature extraction method. *International Journal of Interactive Multimedia & Artificial Intelligence*, 5(5), 2019.

[22] Lee, L.-H., Chen, P.-H., Zeng, Y.-X., Lee, P.-L., and Shyu, K.-K. Ncuee-nlp at mediqa 2021: Health question summarization using pegasus transformers. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 268–272, 2021.

[23] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[24] Li, M., Wang, Y., Zhao, Y., and Li, Z. Transgender community sentiment analysis from social media data: A natural language processing approach. *arXiv preprint arXiv:2010.13062*, 2020.

[25] Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[26] Liu, Y. and Lapata, M. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

[27] Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):pages 1345–1359, 2010.

[28] Punith, N. and Raketla, K. Sentiment analysis of drug reviews using transfer learning. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1794–1799. IEEE, 2021.

[29] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[30] Ranganathan, J. and Abuka, G. Text summarization using transformer model. In *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–5, 2022.

[31] Sharaff, A., Khaire, A. S., and Sharma, D. Analysing fuzzy based approach for extractive text summarization. In *2019 International conference on intelligent computing and control systems (ICCS)*, pages 906–910. IEEE, 2019.

[32] Shirwandkar, N. S. and Kulkarni, S. Extractive text summarization using deep learning. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–5, 2018.

[33] Srujan, K., Nikhil, S., Raghav Rao, H., Karthik, K., Harish, B., and Keerthi Kumar, H. Classification of amazon book reviews based on sentiment analysis. In *Information Systems Design and Intelligent Applications*, pages 401–411. Springer, 2018.

[34] Tas, O. and Kiyani, F. A survey automatic text summarization. *PressAcademia Procedia*, 5(1):pages 205–213, 2007.

[35] Torres, S. Evaluating extractive text summarization with bertsum. 2021.

[36] Torrey, L. and Shavlik, J. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[37] Tutubalina, E., Miftahutdinov, Z. S., Nugmanov, R., Madzhidov, T., Nikolenko, S., Alimova, I., and Tropsha, A. Using semantic analysis of texts for the identification of drugs with similar therapeutic effects. *Russian Chemical Bulletin*, 66(11):pages 2180–2189, 2017.

[38] Vinod, P., Safar, S., Mathew, D., Venugopal, P., Joly, L. M., and George, J. Fine-tuning the bertsumext model for clinical report summarization. In *2020 International Conference for Emerging Technology (INCET)*, pages 1–7. IEEE, 2020.

[39] Wang, Y., Huang, M., Zhu, X., and Zhao, L. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[40] Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big data*, 3(1):pages 1–40, 2016.

[41] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[42] Yadav, N. and Chatterjee, N. Text summarization using sentiment analysis for duc data. In *2016 International Conference on Information Technology (ICIT)*, pages 229–234, 2016.

[43] Zhang, J., Zhao, Y., Saleh, M., and Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages pages 11328–11339. PMLR, 2020.

[44] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

[45] Zolotareva, E., Tashu, T. M., and Horváth, T. Abstractive text summarization using transfer learning. In *ITAT*, pages 75–80, 2020.