National College of Ireland

Extractive Text Summarization of News Reports Leveraging Transfer Learning Contextual Embedders

MSc Research Project
Data Analytics

Savin Vishwas Karkada
Student ID: x20184727

School of Computing
National College of Ireland

Supervisor: Dr. Christian Horn

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Savin Vishwas Karkada
Student ID:	x20184727
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Christian Horn
Submission Due Date:	15/08/2022
Project Title:	Extractive Text Summarization of News Reports Leveraging Transfer Learning Contextual Embedders
Word Count:	XXX
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Extractive Text Summarization of News Reports Leveraging Transfer Learning Contextual Embedders

Savin Vishwas Karkada
x20184727

Abstract

The Surge in textual data has been on an all time high in the recent past in a variety of forms both physically and digitally. One of the leading sources for these data points are the news data which hold enormous potential insights that can transform business operations.One of the key tasks extractive text summarization to provide consumption friendly news reports. The following study investigates the use of contextual embedders to semantically capture the meaning while effectively summarizing the news reports. The contextual embedders have been utilized to perform the task of word embedding while K-Means clustering has been implemented to generate summary out of the embeddings. The pre-trained models BERT, RoBERTa, ELMo and Word2Vec are used to compare the effectiveness and the influence they have on summarization through contextual embedding is studied and measured statistically using ROUGE scores.

1 Introduction

The influx in textual data is on a rise exponentially and are generated in tremendous quantities on different platforms both digitally and physically. The crucial aspect of collecting these data points is to leverage them inorder to derive meaningful insights that can be aligned with the business understanding and sustain these businesses. One of the leading forms of these data points is news data which is being generated continually through various resources.The two notable methodologies for summarization are mainly categorized as Extractive Text Summarization and Abstractive Text Summarization. Extractive Text Summarization picks out sentences that are most relevant to the context of the original corpus, stitching them together to inturn provide the same contextual representation. On the other hand Abstractive Text Summarization rephrases the sentences while summarizing the data taking into account the similar context of the original data.

The following study majorly deals with the study of extractive text summarization of news reports to create the subset features of the input data that are in accordance with its context. The output thus received from the extractive text summarizer contains similar sentences joined together but with relevant coherence and semantic accuracy. The premise of the research is mainly leveraging the semantic aspect of the contextual embedders to attain the contextual quality in the generated summary that can stay relevant to the input news report. Although semantics and contextuality are subjective and cannot be precisely measured with statistical metrics, there are measurements that consider the

overlap of unigrams and bigrams along with longest sentence matching to closely determine the performance of the embedders. The focal point of this research is to study the influence of the contextual embedders in generating semantically qualitative summary. There have been studies carried out with the use of BERT summarizers to summarize textual data where the embedding operation takes place. Similarly studies on extractive text summarization prove that K-Means provides a higher degree of accuracy in stitching sentences in creating summaries. The above studies not only discuss the implementation of these models as a summarizer but also show that attention based models have a higher capacity to improve the contextuality of the summarized output. The studies with future work suggest the implementation of contextual embedders such as ELMo, and Transfer Learning techniques which can potentially improve the contextual quality of summary. This research paper aims to determine and address the following research question:

"How effective is the utilization of transfer learning models as contextual embedders in extractive text summarization of news reports?"
The focal objectives of the research are identified as following.

- Scrutinizing the existing research works to support and validate the following study.
- Pre-processing news data obtained from the valid repository using Python language.
- Fine tuning and implementation of a series of contextual embedders to achieve word embedding in the process of generating summary.
- Analysing the system generated summary by comparing it with the reference summary produced by the authors. Discussing the effectiveness of contextual embedders in combination with K-Means clustering to achieve extractive text summarization using ROUGE metrics.

The following study contributes mainly in affirming a thorough understanding of the effectiveness of contextual embedders in extractive text summarization of news articles which is essentially the basis of summarization techniques. The detailed flow of the research is as follows. The Section 2 details the related work that was previously carried out in the space of Natural Language Processing narrowing down to text summarization and news reports to validate the key implementations done during the stages of research. Further, the methodology followed in the research is detailed in Section 3 while Section 4 provides information on the design specifications of techniques used in the study. Section 5 details on the implementation of the research which then is evaluated and explained in Section 6. The final conclusion and future work associated with the research is detailed in Section 7 of the paper.

2 Related Work

2.1 Contextual Word Embedding

One of the vital processes that is carried out succeeding the data preprocessing stage in order to apply any machine learning tasks on textual data is word embedding. As per the study conducted by Easwar and Uthra (2021), it is evident that the quality of word embedding has a direct effect on the significance of the contextuality developed in the output. The authors also uphold the vitality of word embedding in any automatic text

summarization process. The process of transforming textual units into numerical vectors is known as word embedding. A clear representation of how the text is mapped into its numerical form and the extent of its impact can be understood through this study. There are several word embedders such as TF-IDF, Word2Vec and many more that are traditionally employed to create word embeddings. A study by Wang et al. (2019) denotes how complex structured word embedders outperform the regular word embedders in terms of their contextuality. A clear representation of how BERT could lead the ROUGE scores against several other word embedders can be found in this research.

The above studies show how embedders can show varying results based on their structure. However, there can be a solid validation provided to affirm that the models that have denser network structure that can pick up contextuality in an effective manner can inturn perform well than that of the regular embedders. A major study which was carried out by Naredla and Adedoyin (2022) proved that the models with deeply trained neural net structures have gave better results as compared to word embedders that were built on bag of words. A similar study conducted by Bestgen (2019) and Chen et al. (2019) where the usecase was to determine hyperpartisan in news articles, a trained ELMo provided significant result of 80%.However, a similar study conducted by Huang and Lee (2019) where a combination of ELMo and BERT was implemented on the same dataset showed a clear difference in results proving a dual combination of contextual embedders worked sufficiently well with 68.4% and 60.4% respectively. This study validates that ELMo and BERT can be thus categorized as contextual embedders.

2.2 Study of Contextual Embedding in News Data

News reports possess a higher degree of semantics which generally comes from the style in which the authors present their work. This is essentially an amalgamation of a human opinion and ground truth, Due to this human intervention, there can arise a situation where the factual substance from the news is hindered. However, a contextual embedder that takes into account the contextuality should ideally generate the summary while preserving its factual integrity. A study by Huang and Lee (2019) represents how a style of writing can influence and lead to contextual disharmony and adoption of contextual embedder was functional. Similarly, the study performed by Naredla and Adedoyin (2022) shows how the implementation of ELMo with size above 350 MB was employed to embed the words with a batch size of 100 for over 645 articles. The forward and the backward pass structure effectively took the semantic aspect into account thus giving robust results. Here, ELMo was only used as an embedder and Random Forest was used as a classifier.

2.3 Effect of Contextual Embedders on News Text

There have been substantial studies conducted on the role of contextual embedders when employed on news related data. One such notable research was done by Büyüköz et al. (2020) where the use of DistilBERT and ELMo to embed news data is shown. The vectors attained from the embedding of text was then fed into a consecutive layers of forward and backward Long Short Term Memory(LSTM). Dropout layers were introduced in between to bring in computational stability, Rectified Linear Unit was used as an activation function and to finally classify the data, Softmax activation function was implemented. A key resource to validate this study was given by Hürriyetoglu et al. (2019) where a sim-

ilar model lineup was deployed to to classify protest information where the data was in the form of news statements. Here ELMo and DistilBERT were used both in combined format and individually. The results from these embeddings were evaluated both on their combined effect and also their individual influence on the data. The results clearly showed that when ELMo and DistilBERT were finetuned and implemented on their own as embedders, the performance of the models were significantly higher. The NTest, CTest and Drop for EMLo were identified as 82, 72 and 12.2 and on the other hand DistilBERT showed 81.8, 72.2 and 11.8 respectively. This gives a clear inference that the use of ELMo and variants of BERT can add contextual significance to news data. Further, a study conducted with the use of pointer generator network that used ELMo to embed the news data where the dataset was CNN-Daily mail, gave an indication that the results were promising and hence the author of Mastronardo and Tamburini (2019) stated "The representation is the result of a weighted combination of the hidden states of the language modeling architecture". The steps followed in the research contained pre-processing of the data such as lower casing of news text using NLTK toolkit. The ELMo model took 1024 dimensional embedding with model size being 5.5 MB. The embeddings from ELMo was passed onto LSTM layer of 512 on top of a linear layer. ROUGE toolkit was chosen to be the right mode of measurement on CNN-Daily Mail and the results obtained were ROUGE-1 38.96, ROUGE-2 16.25 and ROUGE-L 34.32. This gives a strong validation on both the grounds that the use of ELMo as embedder on CNN-Daily Mail dataset can be seen effective as the following research uses CNN-Daily Mail as the dataset throughout the research.

2.4 Use of Transfer Learning in Text Summarization.

Transfer learning techniques rose to popularity when a vast host of problems were able to be addressed putting into use the strong pre-trained aspect they possessed. This enabled to carry out tasks that required a huge amount of data without necessarily training them drastically saving the computation cost involved in training a machine learning model. Most transfer learning models show higher performance rates when the model is attention based. Attention based models are essentially the a series of encoders and decoders. The input text thus fed into these layer traverse through these alternative layers which results in the consideration of the semantic nature of the textual data. The structure of the attention based models can be well understood through a path breaking study that was done by Vaswani et al. (2017). Although there are studies that exhibit the effectiveness of higher complex models such as GPT-1, GPT-2 and GPT-3 in terms of automatic text summarization, the models that are extensively used in major natural language processing use cases is known to be BERT. BERT and its variants however have been consistently showing substantial results when it comes to automatic text summarization when carried out with required finetuning Wolf et al. (2019). In one of the studies conducted by Kieuvongngam et al. (2020) where the data was the medical reports on Covid-19, the authors employed GPT-2 and BERT to achieve automatic text summarization. Here the BERT model used was a pretrained unsupervised transformer having 12 layers of attention heads and 6 layers of encoder. In addition to this a pretrained GPT model was employed to observe and benchmark the model performance. As stated by the researchers in the study, the authenticity of the model performance in terms of capturing the contextual coherence was highly subjective and be only gauged by human intervention. However, the quantitative results drawn from the models are measured by

statistical metrics and are portrayed in the research. The embeddings drawn from these models were then applied onto K-Means clustering and K-Nearest Neighbours to stitch the segregated vectors to derive summary. The extractive summarization showed 40% to 60% compression ratios on their ROUGE scores. From this the authors arrived that the results produced by BERT and GPT were much stronger than that of abstractive text summarization. In a research conducted by Weng et al. (2021) to determine the influence of contextual embedding in speech summarization, where pre-trained BERT was fine tuned on the data to enhance the model performance. Also certain embedding techniques like positional embedding and confidence scores were applied to keep the robustness of the model in check. However, the accuracy of the model was determined using the ROUGE toolkit. To benchmark the results produced by BERT, classic unsupervised LSA and VSM were deployed in the initial phase. To further make the validations concrete, the training of deep neural nets were done. To attain further solidified results, the model was used to perform summarization on textual data. The results strongly upheld the performance of BERT on extractive summarization of both text and speech. The finetuned model however was applied on CNN-Daily Mail dataset to summarize the news data and BERT showed promising results. The above studies thus establishes a concrete foothold to validate that the use of BERT in textual data for summarization can provide optimistic results and hence base the following research on contextual embedders.

In another study where BERT was implemented to achieve extractive text summarization, this model was primarily employed as a word embedder. The author in the study Miller (2019) states that BERT was used to compare against traditional methods such as TextRank. The data was mainly based on lectures and from the observation of results both quantitatively and human evaluation, it was concluded that BERT showed better performance than that of TextRank concluding BERT and BERT variants can deliver promising results when used on domain specific datasets. One such claim can be validated with a support of a study that was conducted by Du et al. (2020) where a variant of BERT BioBERTSum was used which essentially is a domain aware model to perform extractive text summarization of medical reports. This model has been fine tuned on medical reports on top of a BERT based model. PubMed was used as reference model to benchmark the results. The proposed model architecture gave outstanding results outperforming SOTA models on ROUGE scores. Through above studies there can be drawn a clear inference that transfer learning techniques have proven to show promising results in extractive text summarization. Domain specific models or models fine tuned on the domain data can drastically improve the model performance and help in increased contextual quality in the output. Also the studies can be a valid justification to employ K-Means clustering as it effectively picks contextually rich sentences in generating the summary.

2.5 Role of BERT Variants as Contextual Embedders

Transfer learning techniques have been widely utilized across a variety of domains and datasets. Although BERT and their variants have been showing significant results as summarizers themselves there are enough studies validating the use of transfer learning pre-trained models as contextual embedders. There have been instances as recorded in the previous subsection the studies from Huang and Lee (2019) and Bianchi et al. (2020) proves that BERT as contextual embedders have significantly been showing improved

results. The study also shows that in an ecommerce dataset, a variant of BERT known as Prod2BERT that was incorporated and was seen to produce much better results in comparison to Word2Vec. Using SBERT as sentence transformers have also shown to produce improved results in terms of deploying pre-trained BERT models which can also be trained on domain specific custom dataset. A study by Suryadjaja and Mandala (2021) shows that use of BERT embedding using SBERT sentence transformer which aloows to load a pre-trained model, was further fed into a topic modelling system after embedding to undergo density peak clustering with the aid of cosine similarity. The architecture produced ROUGE scores to be 0.33, 0.07 and 0.101 on ROUGE-1, ROUGE-2 and ROUGE-3 respectively outperforming baseline models such as LDA, VSM and DPC. This gives a strong foothold on the usage SBERT as the primary method to implement pre-trained BERT models. To validate the use of BERT variants another study performed by Pavlov and Mirceva (2022) explains the use of RoBERTa to classify news data related to covid. In this study the authors uphold the significance of contextuality brought in by BERT and RoBERTa on news text. Just the use of pre-trained model shows that RoBERTa provided higher accuracy with 0.5 while BERT showed 0.3. However, after the fine tuning of the model, the data tested on validation test showed increased result of BERT than RoBERTa with accuracies showing 0.98 and 0.97 respectively. This research provides a clear inference to validate the use of RoBERTa as one the models which can be implemented to study the contextual embedding significance in news related data.

2.6 Conclusion

As a result of curating and understanding the above researches to justify and validate the approach proposed towards identifying the effectiveness of contextual embedders using transfer learning techniques, a strong foothold to justify all the steps that are proposed to carry out in this research is established. The papers studied above show their contribution in various areas of machine learning and some more specific to news domain which can act as a major standpoint to justify the claims that will be done through the results of this research. The key learning from the above papers can be identified that as the use of BERT and other BERT variants as the vital models in comparison of embedding process. There are enough evidences to support the fact that the K-Means clustering however could be on of the relevant means to generate summary after the word embedding process. From the studies that compared various embedders to determine the performances of embedders it is evident that having baseline models in order to benchmark the performances of higher complex models is extremely crucial. And as per the researches shown above most papers consider the likes of TextRank, TF-IDF and Word2Vec as the baseline models. However, as this research deals with determination of the effectiveness of contextual embedders, Word2Vec will be chosen as the model base model which is then followed by a slightly complex structure ELMo and then goes on to observe the performances of BERT and RoBERTa.

3 Methodology

3.1 Business Understanding

Having news data in abundance, there is surge the distribution of data in various forms and levels be it digitally or physically. Summarization of data is a widely implemented

use case where several applications that depend on running summarization engines in the back end to develop summarized version of the text. As short news snippets are being extensively used in the media business it is essential that the obtained news data is contextually precise. This research delves into understanding the use of contextual embedders that convert textual data into numerical vector entities with semantic ability. These different embedders are trained and their performance is studied when employed to summarized textual data. The process of research thus follows the KDD methodology to aid seamless execution.

3.2 Understanding the Data

The CNN-Daily Mail dataset is one of the largest English language news data repositories made available by Hugging Face[1]. The dataset contains close to 300,000 unique news articles curated for the purpose of machine learning research and made open source. The curated set of news articles are originally authored by journalists from CNN. They are given by unique fields of 'id', 'articles' and 'highlights'. Articles and highlights are the actual news article and their summary representations. Hugging Face provides a custom split of data into train test and validation set for machine learning and research purposes[2].

The curated set of articles were originally written by CNN journalists within the span of April 2007 to April 2015. These datasets however do not possess any annotations. The most recent version of the data provides the authors name unlike the previously curated versions. The dataset also used to train pre-trained summarization models that are widely being used. The dataset also is said to show a slight bias in terms of gender however tremendously lower than that of most research news data[3]. This is said to have happened as the news articles were written for United States and UK covering events locally which would have picked such bias learning. Inorder to prevent this bias only a small subset is chosen to train the model in this research as the models being used are pretrained on a variety of textual data which will be sufficient to carry out embedding purposes.

3.3 Data Preparation

The data used to train the model is extremely important to align the model to the domain that is being worked on. CNN-Daily Mail dataset is however a large data which has been previously split by Hugging Face for train, test and validation. As there is a huge amount of training data which requires significantly powerful computational resources, the data is made into a small subset with which the pre-trained model is further trained. The data is cleaned and related checks for the evenness of the data is performed. The most important process involves removal contractions, and stop words using NLTK library. This adds to the quality of the data during the training stage. Further the HTML tags are removed to obtain a clean set of training data. This data is further tokenized and lemmatized before being fed into the model in the implementation stage.

[1] https://huggingface.co/datasets/cnn$_d ailymail$

[2] https://huggingface.co/datasets/cnn$_d ailymaildata - instances$

[3] https://huggingface.co/datasets/cnn$_d ailymaildiscussion - of - biases$

3.4 Model Training Phase

As the main aim of the research is to identify the effectiveness of the pretrained contextual word embedders, the primary considerations for embedders are ELMo, RoBERTa, BERT and Word2Vec. All these models possess the ability to pick context from the input data. Word2Vec and ELMo are slightly lower in their performance levels, as they are not trained or finetuned on CNN-Daily Mail dataset Mastronardo and Tamburini (2019). However, on the other hand BERT and RoBERTa carry higher contextual ability. ELMo, BERT and RoBERTa are pre-trained with a vast amounts of data as they are fine tuned with CNN-Daily Mail dataset on a small subset to align the model to the news domain. This can be used to study the performance of the model in determining the cohesiveness of the model generated summary. Apart from contextual embedders K-Means clustering is used to stitch the sentences together after the contextual embedders convert textual data into numerical vectors.

3.5 Model Evaluation

After the training phase the model performance is tested using the test data given by Hugging Face. The input article is chosen from the test split and provided as an input data. The model performance is determined by comparing the model generated summary and the 'highlights' which is a summary provided by the author of the original news text. The key metric to identify the model performance is ROUGE Score[4]. ROUGE Score is obtained from the Rouge library. Recall-Oriented Understudy of Gisting Evaluation also abbreviated for ROUGE measures the overlap of n-grams between the machine and author generated summaries. There are three main metrics that ROUGE calculates and they are ROUGE-1, ROUGE-2 and ROUGE-L. Average ROUGE gives the average of all the ROUGE scores Weng et al. (2021).

ROUGE-1 measures the overlap of unigram or the single words between the model and author generated summaries. However, ROUGE-2 refers to the overlap of bigram between the model and the reference summaries. ROUGE-L refers to Longest Common Subsequence, [LCS]. This takes into consideration the longest sentences that lie common in both the reference and the system generated summary. Finally the average ROUGE gives the average of all the ROUGE scores. The ROUGE values are broken down into F1 measure and Precision where F1 score is simply the harmonic mean of precision and recall. Higher value of F1 score represents better performance.Recall represents the percentage of terms from the reference summary that were included in the generated summary. Precision represents the percentage of n-grams or bigrams present in the model generated summary which is also present in the reference summary from the author.

4 Design Specifications

This section represents the design flow of the research and its specification with respect to all the key entities used in the implementation.

[4]https://pypi.org/project/rouge/

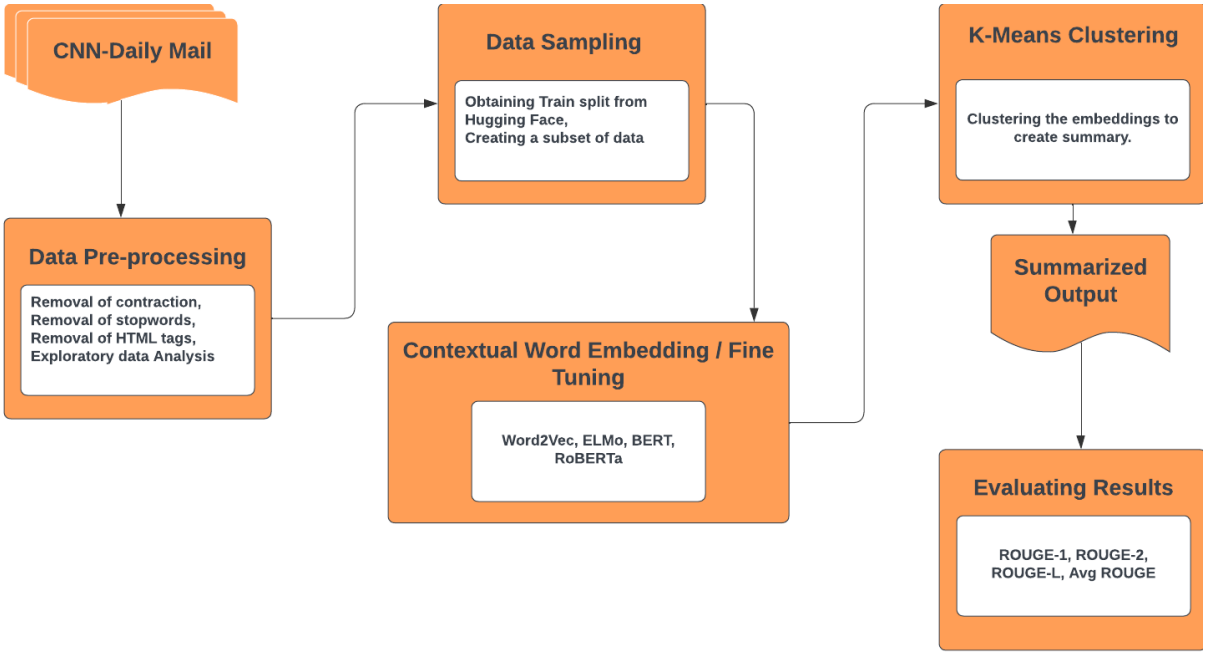

Figure 1: Research Flowchart

4.1 Contextual and Traditional Embedders

The research is primarily set on grounds to identify and determine the effectiveness of contextual embedders when used as word embedders to perform summarization of news articles. The primary use of embedders in any natural language processing implementation is to convert the natural language into numerical vectors that is suitable to be given as inputs to any machine learning model. Numerical vectors however should poses the exact same representation as that of the natural language word in the form of numerical vectors. This is essentially the process grouping terms that are highly similar in terms of the context they belong to. In this study the use of different varieties of contextual embedders are identified and their effect on the summarization is studied. However, these embedders are the not corpus specific and hence are not trained on any news related datasets. The goal here is to perform embedding operation on news data and hence the use these models are not fine tuned on CNN-Daily Mail dataset. However, as here lies an interesting scope to observe how similar models summarize the data when the models are trained with CNN-Daily Mail dataset, we have considered the transformer based summarization models provided by hugging face library which are trained on CNN-Daily Mail. This is a summarization model which performs word embedding and summarization of the text on its own as a single entity. The results from these models can be of a benchmark reference parallel to the proposed solution during comparison and evaluation.

4.2 ELMo Word Embedder

Elmo is a slightly complex contextual word embedder that is built as a two layer structure comprising Bi-directional LSTM Peters et al. (2018). The architecture is designed such that each layer has a forward and a backward pass in character level CNN representing input text in its raw form which is passed onto the first layer. The forward pass in the network considers the first word of the input data and retains the contextual aspect of the preceding word. Similarly, the context of the succeeding word is captured by

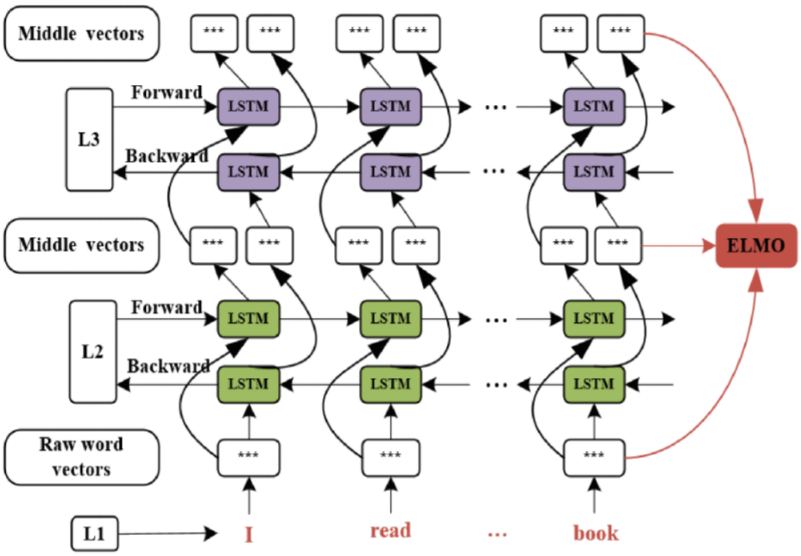

Figure 2: ELMo Architecture, Image Source: Huang and Zhao (2020)

the backward pass of the LSTM network as shown in Figure 2. Further on from the combination of both these networks a new vector is produced called intermediate layer between the stacks. These vectors act as inputs for the following stack which then follows the same process going forward making the data contextually rich. Finally outputs from CNN layer, intermediate layer and the top layer combine to form weighted sum of vectors which is the final embedding.

4.3 Word2Vec

Word2vec word embedder was initially published in the year 2013 with a neural network model to map patterns of contextuality in a large corpus Mikolov et al. (2013). This model once trained on relevant dataset can be used to complete partial sentences. The architecture of Word2Vec mainly contains three salient features. The primary block is the vocabulary builder, which is followed by the context builder which then goes on to neural network with two specified layers. The vocabulary builder extracts words from the raw corpus and fabricates a vocabulary of its own using the unique words. The succeeding stage is where the vectorization or the word embedding of the text happens. The output from the vocabulary builder containing index and count. This does not consider only the particular word but the entire range of context zone which includes the whole spectrum of words received during the input of the words. Thus a word pair is formed at this stage and is fed into the next block which contains two layers of neural networks. First layer in the neural networks contains as many neurons as that of the input words in the vocabulary block. One hot encoding takes place at this point generating an output which is then transformed when passed to the second layer containing a softmax activation function. In this research Word2Vec is used as a basic contextual word embedder using Gensim[5].

[5]https://radimrehurek.com/gensim/

4.4 BERT

BERT or Bidirectional Encoder Representations from Transformer is Google AI's pre-trained model developed to perform a variety of tasks associated with NLP. As per Google AI, the BERT model has been trained on the massive corpus of Wikipedia, which has 2500 million words text paragraphs and 800 million words books corpus. BERT has been produced with two main architectures, BERT Large and BERT Base[6].

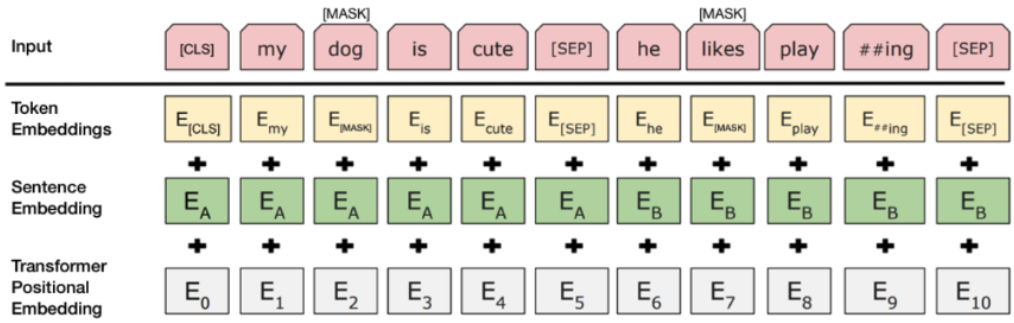

Figure 3: BERT Embedding Architecture, Image Source: Devlin et al. (2018)

BERT is essentially as stack of transformers placed to input sentences through them as in Figure 3. Each transformer model is however a set of encoder decoder network. These networks consist of self attention and attention layers on encoder and decoder sides respectively. BERT base consists of 12 layers on the encoder side and the BERT Large contains 24 layers on the encoder side. In this study BERT Large has been considered as a contextual embedder Devlin et al. (2018). As the stacks are bidirectional, the word embeddings thus produced are contextually sound as the layers consider the sequence of word inputs from both ends of the network. They have a larger feed forward network and have 340M parameters. The Sentence Transformer - BERT from SBERT is used to attain the word embeddings from BERT. This is a framework based on PyTorch and Transformers. At the same time a pretrained model from Hugging Face trained on CNN-Daily Mail Dataset - BERT Large Summarizer is also widely used as a summarizer. This model calculates the word embeddings and summarizes the input. The background of the BERT model training phase will be detailed in the Section 5.3.3.

4.5 RoBERTa

RoBERTa, also abbreviated for Robustly Optimized BERT Pretraining Approach is an extended entity of BERT model with improvised training procedures. RoBERTa has been designed by training the model for extended period of time while having greater mini-batch sizes than usual. The data used to train this model is also significantly large. Longer sequences were used to train the model making them Robust in terms of their contextual abilities. The training phase included a language masking strategy applied to the training data by dynamically changing the masking pattern Liu et al. (2019). These features however enabled RoBERTa to outperform BERT in a significant number of tasks associated with NLP. The model was designed in a way to truncate next sentence prediction objective which greatly reduced the problems encountered with

[6]https://cloud.google.com/ai-platform/training/docs/algorithms/bert-start

NSP in BERT. Similar to BERT, RoBERTa was trained on English Language Wikipedia. However, the training of the model was boosted with an additional set of data of size 160GB on CC-News[7] with suitable sizes to maintain training set size effects. RoBERTa uses Byte Pair Encoding approach for tokenization. The vocabulary size is set to 50,000 unlike the character level encoding as seen in BERT. As RoBERTa is trained on news corpus, the effectiveness of the model aiding contextual significance can be observed in the implementation stage.

4.6 K-Means Clustering For Summarization

K-Means clustering is a methodology where the certain data points are segregated into K number of clusters based on their similarities measures. In this study the K-Means clustering algorithm is incorporated to summarize the document. The textual data - news articles after being transformed into their contextualized numerical vectors, are fed into the K-Means clustering algorithm. The value K defines the number of clusters to be obtained from the randomly initialized set of textual data points present in their numerical form Kieuvongngam et al. (2020). The learning process begins when there are centroids randomly initiated. This marks the beginning stage of clustering procedure. The process carries out on an iterative basis repeating the operation over again to optimize the centroids. The optimization phase terminates when either there is no scope for the centroids to stabilize or after the specified number of iterations have comes to an end. The specified K value inturn acts as the number of sentences required in the output. As the dataset is independent of the value of the output, there is no necessity to identify the optimum number of clusters using the elbow method. Providing the number of sentences to be obtained as output will suffice the K requirements. In this study, NLTK's KMeanClusterer has been used to achieve clusters. Cosine distance can be applied to obtain the distance between two vectors which inturn provides the similarity measurement. The distance between the sentence vector and the mean vector also known as centroids is measured by assigning centroid values to each row of text in the list. Scipy's distance matrix provides suitable measurements. The final stage of K means is grouping of sentences based on the clusters. The sentences are stitched ascending order to match the original text format and the highest value from the each cluster is represented as output which is the summary.

5 Implementation

5.1 Data Loading

Data for this research has been acquired from the Hugging Face repository that contains a vast collection of authentic data curated for research purposes. In this research the domain specific data is CNN-Daily Mail which is a huge collection of news articles from a variety of authors. The data can be instantly loaded from the datasets library from which the entire news data of CNN-Daily mail can be loaded to Jupyter notebook. However, in this research the data is stored locally in CSV format obtained from Hugging Face with sections of splits pre defined and then uploaded to Jupyter using pandas in manually.

[7]https://resources.wolframcloud.com/NeuralNetRepository/resources/RoBERTa-Trained-on-BookCorpus-English-Wikipedia-CC-News-OpenWebText-and-Stories-Datasets/

From this point onwards the data was split into different formats to understand the data structure to aid EDA.

5.2 Data Pre-Processing

The CSV file that is obtained from Hugging Face contains splits of test train and validation and hence for the Data pre-processing stage only train data is considered. the train data is then mounted to Jupyter notebook using pandas library to further the processes. Initially the data loaded consists of the news articles, highlights which is the summarized version of the same dataset provided by the same author and the id. The id points to the original artefact of the article to validate the source. Once the data is loaded into the dataframe, the column 'id' does not contribute to the training process and hence the column is removed.

5.2.1 Analyzing the News Length

Firstly, to identify the structure of the sentences the columns articles and highlights are mapped into a histogram as shown in Figure 4.

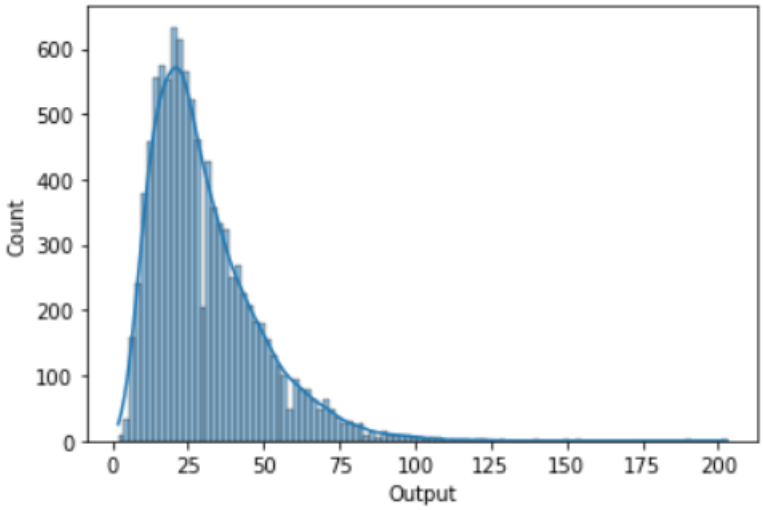

Figure 4: Sentence length of news articles

This provides a generic first understanding of how the words are structured. The mean length of sentences in articles column was 30.58. The distribution of the length of the sentences is slightly skewed showing positive skewness.

This provides an idea of how the sentences are structured and the length of the news articles. The mean count of sentence length here is 1.5. This provides an overview of how the summarized version of the data looks like. The summary given is lesser than 10% of the actual news article. Hence for the purpose of comparison the summary produced ny the trained model should be kept minimal. In this study an arbitrary number five is chosen as the output sentence and is kept constant across all the models for training purposes.

5.2.2 Elimination of Contractions

Contractions are the shortened or abbreviated versions of certain words which are most often used in an informal method of writing. However, as news articles contain a mixture of slightly relaxed style of writing and formal writing, there are some contractions used in the articles. Hence these contractions are expanded using standard Python libraries.

5.2.3 Data Sampling

The CNN-Daily mail dataset provided by Hugging Face provides a generic split of train test and validation. The split ratio is as follows.

Table 1: Dataset split breakdown.

Dataset Split	Number of Instances
Train	287,113
Validation	13,368
Test	11,490

The total of 287,113 instances are however not used to train the model as the models that are going to be used are pre-trained on a vast amount of dataset. Hence inorder to align the pre-trained model to a domain specific tangent, a small subset if 10 instances are picked in random as they all belong to the same domain. Each instance has average of 30 sentences which mark 300 trainable sentences in approximation.

5.3 Model Training

5.3.1 Word2Vec Word Embedding and Summarization

Initially for the comparison of word embedders inorder to understand the effectiveness of the contextual word embedders a conventional method of word embedding is used to set a benchmark the results with which there will be a distinct determination as to how effective can the contextual word embedders be and hence Word2Vec is employed. The NLTK's punkt is imported[8].

- The sentence tokenizer from NLTK is used to tokenize the sentence to prepare the sentence to be embedded. Once the sentences are tokenized now the regular expressions are used to strip off the characters that are unimportant. Every element that is not a letter by space is replaced using RegEX. The capital letters present in the sentences are lowercased and removal of stopwords.

- Proceeding this step the gensim module is used to import the Word2Vec embedder. The Word2Vec embedder ranks above in the traditional word embedders compared to bag of words and TF-IDF as Word2Vec to an extent captures the context of the input sentences. Vector representation of all the constituent words are grabbed and averaged to derive an amalgamation of vectors.

- Once the vector embeddings are obtained the vectors are fed into the K-Means clustering algorithm to arrive at a summary. The K value of the summary is kept

[8]https://www.nltk.org/api/nltk.tokenize.punkt.html

as 5. This is an arbitrary value chosen to define the number of sentences required as an output. k = 5 will be kept constant across models. After the clustering of sentences, news texts with similar context are grouped accordingly.

- This is then arranged as per the values of its centroidal distance. One sentence from each cluster is picked. The sentence with the least Euclidean Distance acts as a representative of the cluster it is present in using SciPy's distance matrix[9]. This further joined in ascending order to match the original text coherence.

5.3.2 ELMo Contextual Word Embedding and Summarization

ELMo embedding is a contextual embedder than is used to convert textual data points into numerical vectors contextually. To implement word embedding using ELMo word embedding, ELMo2 from Allen NLP is utilized[10]. In order to seamlessly implement ELMo2 pretrained model, the implementation is carried out using Tensorflow 1.6, as Tensorflow 2 does not suitably support ELMo2. Hence the version is stepped down on the Jupyter notebook IDE in Google Colaboratory. The dataset curated by Hugging face offers splitup of test, train and validation. The input sample is taken from the test split.

- The ELMo model is initialized from the tfhub with required signatures parameters updated. The trainable parameters are set to TRUE such that 4 numbers of scalar weights and all LSTM cell variables are equally trained. For this implementation all the other parameters are kept static. This enables the model to produce the embeddings of words that are not previously exposed to the model during the training phase. In this study ELMo receives input signature as default. The pre-processed data is converted into a list of each sentences as the ELMo model considers list values as input.

- The list of sentences are then transformed into numerical vectors of shape (10, 1, 1024). 10 represents the batch size that is input to the model and 1 is the sequence length. 1024 denotes the dimension of each ELMo vector. Once the ELMo vectors are obtained these contextually rich vectors are clustered using K-Means cluster. In the implementation of ELMo, K-Means is obtained from the SKlearn library[11]. Here, in the first iteration the K-value is considered as 5.

- Here 5 is considered arbitrarily to suit the size of the output thus making it relevant for comparison across different models. The clusters thus created are arranged as per the values of their centroids mean using pairwise distances argmin which then stitches the sentences from each of these clusters to generate summary of 5 sentences picked from each of the clusters. The results from the ELMo contextual embedders are as follows.

5.3.3 BERT Contextual Embedding and Summarization

The comparison of word embedders are furthered by using another contextual word embedder known as BERT. Possessing a bidirectional transformer based architecture BERT has the ability to capture contextual elements from the textual data. In this study BERT

[9]https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance$_matrix.html$
[10]https://allenai.org/allennlp/software/elmo
[11]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

is used as a contextual embedder to embed textual data. BERT embedder for this study is from SBERT.net[12] a sentence transformer hub that hosts a variety of transformer based embedding models. SBERT is a PyTorch and Transformer based python framework with a collection of pre-trained models. The model bert-base-uncased is used as the primary embedding model. However, the model has to be trained on CNN-Daily Mail dataset in order to make it domain specific. The Hugging Face library provides, train test and validation spit on the dataset. For the fine tuning of BERT in this study a subset of training data is considered. As this training phase will be an unsupervised learning approach, the TSDAE - Transformer Based Denoising AutoEncoder[13] method is implemented. Figure 5 shows the flow of BERT based embedding architecture where a pooling layer situated between encoder and the decoder stacks. The sentence transformer module from is imported load the encoding model. Data loader from sentence embedder is also loaded to batch the data. batch size is set to 128 to allow a set of 128 entities during the fine tuning phase. The model name is specified from the set of pre-trained models available from the Hugging Face library. The bert-base uncased is trained on 4 cloud CPU with pod configuration.

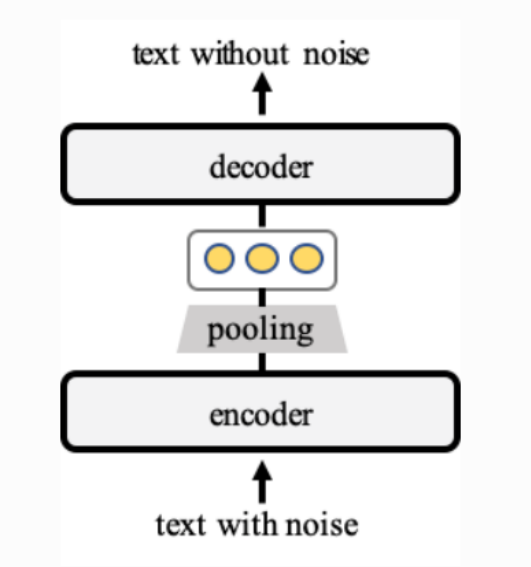

Figure 5: TSADE Fine tuning Architecture, Image Source: Wang et al. (2021)

- The first layer is the word embedding layer where the model is loaded is initialized with bert-base-uncased. This is a transformer layer for embedding. Further a pooling layer is added which is assigned with [CLS] tokenizer. With this the embedding model is completely initialized and is ready to to be trained on the data. In the next stage the data that is already loaded into the notebook is assigned as a list after which the list of news sentences are introduced with a minimal loss function.

- The DenoisingAutoEncoderDataset package from the sentence transformers module induces noise into the dataset on the go. The dataloader then batches the data with the specified batch size which in this case is 128. Thus the required model framework is structured and set. The final finetuning step is done with the model.fit method

[12]https://www.sbert.net/
[13]https://www.sbert.net/examples/unsupervised$_l earning/TSDAE/README.html$

with 100 epochs. The learning rate is set to lr:3e-5. Once the training process is complete the model is saved for further use. Thus the fine tuned bert model is now used as a contextual embedder before being fed into K-means clusters for summarization.

- Once the embeddings are obtained, the data is clustered using the K-means clustering algorithm. The algorithm. The arbitrary value of cluster size 5 is defined and assigned to the clusters to keep the sentence outputs common across the models for comparison. The K-means clusterer sorts the sentence list into clusters based on the cosine distance obtained from the package. The following step calculates the centroid and the distance from the centroid using the distance matrix from the scipy.spatial module. The summary to original news mapped by the machine is as shown in Figure 6.

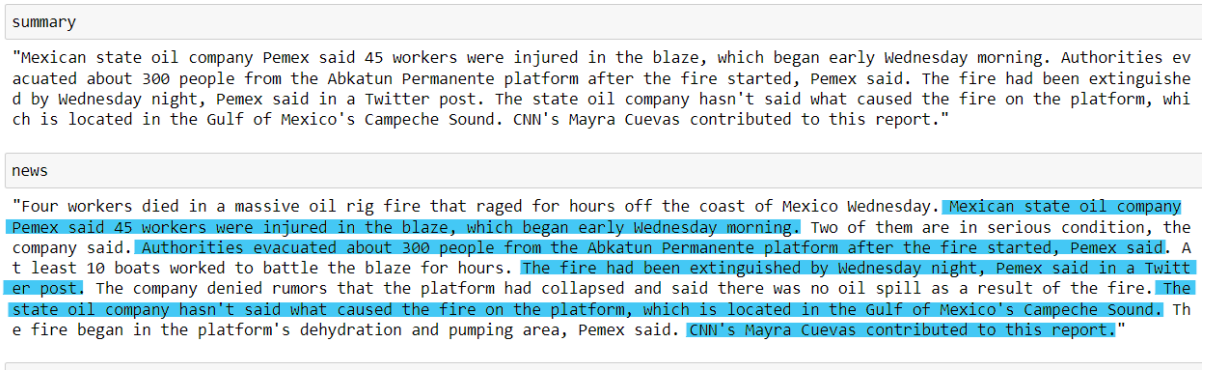

Figure 6: ELMo Architecture

5.3.4 RoBERTa Contextual Embedding and Summarization

The RoBERTa model is used to contextually embed the news article and the model used in the training phase is 'roberta-base'. This model is attained from the Hugging Face library and loaded using the SBERT sentence transformer to create the embedding architecture. The architecture remains same as that of BERT embedder with an encoder layer, a pooling layer and the decoder layer. The tokenization carried out in the training phase is [CLS] as there is an encoder and decoder layer structure present in the embedding layer.

- The learning rate and the batch size are lr:3e-5 and 128 for the finetuning of the RoBERTa model. These parameters are kept similar to that of BERT inorder to bring in consistency in the training of the model so as to achieve a fair comparison.

- The further the finetuning procedure is similarly by creating a list of input sentences. The the input text is kept common across all the models to differentiate the results from all the embedders. Further, the addition noise using denoising package is done so that sufficient ,loss function is brought in. Thus the sentence transformer is structured to train on the data.

- Post this step the model.fit is applied to train the model until convergence. Once the model is trained the model is used to obtain the contextual embedding from the

news text. The embeddings obtained are represented as shown in Figure 7 in the form of embeddings and their centroid values along with distance from the centroid after which these embeddings are clustered as per their contextual significance. Finally, the SciPy package stitches the sentences from the clusters contextually to generate the summary.

	sentence	embeddings
0	Four workers died in a massive oil rig fire th...	[0.11393637, 0.1010204, -0.022377394, 0.089156...
1	Mexican state oil company Pemex said 45 worker...	[-0.432263, -0.590296, 0.111419626, -0.0771262...
2	Two of them are in serious condition, the comp...	[0.020905137, 0.19470415, 0.31530708, -0.13897...
3	Authorities evacuated about 300 people from th...	[-0.20823689, -0.02571734, -0.015867688, -0.19...
4	At least 10 boats worked to battle the blaze f...	[-0.10255332, -0.21487913, 0.3210289, -0.19523...
5	The fire had been extinguished by Wednesday ni...	[0.15050356, 0.123558596, 0.07323398, 0.016661...
6	The company denied rumors that the platform ha...	[-0.5023808, -0.34442577, -0.11295362, -0.0616...
7	The state oil company hasn't said what caused ...	[-0.4914506, -0.292371, 0.13042194, -0.1312533...
8	The fire began in the platform's dehydration a...	[0.018947579, 0.38712886, -0.13495126, 0.12417...
9	CNN's Mayra Cuevas contributed to this report.	[-0.22802967, -0.37490198, -0.27882448, -0.047...

Figure 7: News article and its word embeddings

6 Evaluation

The performances of the contextual word embedders implemented in Section 5 to embed news data and then summarize it using K-Means clustering are critically evaluated in this section to identify and determine the effectiveness of these models. The key metrics to evaluate the results are the ROUGE Scores. The ROUGE Scores are further divided into F1 measure, Precision and Recall. The final performance considerations of these models depend on the average ROUGE scores as the result of the above breakdown.

6.1 Experiment with Word2Vec

As mentioned in Section 5.4.1, Word2Vec embedder does not effectively take into consideration the contextuality of the data. However, the vectors are created when news articles are fed into the model which when clustered using K-Means generates summary with 5 sentences. The results of this combination is given in Table 3.

Table 2: ROUGE Scores for Word2Vec

ROUGE	F1	Precision	Recall	Score
ROUGE-1	0.32	0.25	0.42	0.32
ROUGE-2	0.11	0.09	0.16	0.12
ROUGE-L	0.32	0.25	0.42	0.12
Avg ROUGE				0.25

The table shows that the ROUGE-2 and ROUGE-L values is extremely low compared to that of the value of ROUGE-1. This represents that the bigrams and the longest

sentence in the generated data poorly overlap with the summary provided by the author. However, the F1 value of ROUGE-1 shows sufficiently higher value representing that 32% of the unigrams in Word2Vec generated summary are overlapping that of the highlighted summary.

6.2 Experiment with ELMo

Similarly the results from ELMo contextual embedders can be observed to have produced significantly better results. The results from ELMo embedding and K-Means provide summary with 5 sentences and the results are represented in Table 4.

Table 3: ROUGE Scores for ELMo

ROUGE	**F1**	**Precision**	**Recall**	**Score**
ROUGE-1	0.31	0.26	0.39	0.32
ROUGE-2	0.14	0.11	0.18	0.14
ROUGE-L	0.31	0.26	0.39	0.14
Avg ROUGE				0.26

The ELMo model however did not enhance the performance largely as the ROUGE-2 score similarly reduced drastically showing that the bigrams are still weaker than the number of overlapping unigrams. The longest sentence similarity value also showed a slight jump in its precision value giving an average of 0.14. This depicts that only 14% of the generated summary overlaps with the highlight of the author as far as the longest sentence term is considered. However, the spike is visible in average ROUGE giving 26% in all ROUGE scores.

6.3 Experiment with BERT

The implementation of BERT embedding as mentioned is Section 5 proves to provide better embedding as the finetuning of the model works considerably well. This can be understood by the results of ROUGE-1. This term gives the maximum similarity score of 0.51 describing that the unigrams or the single token terms present in the generated summary hold on an average, 50% of the terms similar to that of the authors reference summary.

Table 4: ROUGE Scores for BERT

ROUGE	**F1**	**Precision**	**Recall**	**Score**
ROUGE-1	0.50	0.39	0.69	0.51
ROUGE-2	0.24	0.18	0.35	0.24
ROUGE-L	0.48	0.37	0.66	0.24
Avg ROUGE				0.41

This value gets its boost from a strong F1 value across ROUGE-1 and ROUGE-2 showing that the similar terms both unigrams and bigrams are strongly correlated hence raising the average ROUGE score to 41% . However, F1 score with respect to ROUGE-L representing the longest sentence similarity is also significantly high proving that there is a major step up in picking the contextuality of the data and hence showing maximum

similarity between generated and the reference summary. Thus implementation of fine tuned BERT significantly raised the contextual capacity of the summarizer.

6.4 Experiment with RoBERTa

Finally the implementation of RoBERTa as discussed in the Section 5 also shows the performance of finetuned contextual embedder. The results from the application of RoBERTa as embedder with K-Means is given in Table 6 below.

Table 5: ROUGE Scores for RoBERTa

ROUGE	**F1**	**Precision**	**Recall**	**Score**
ROUGE-1	0.43	0.33	0.60	0.43
ROUGE-2	0.18	0.13	0.27	0.18
ROUGE-L	0.40	0.31	0.57	0.18
Avg ROUGE				0.34

The results shown however have a slight dip in the values of precision and recall for the values of ROUGE-2. Although the training parameters remain the same, there was no major difference found in the F1 scores, keeping the average ROUGE score 0.34. Although there exists a strong ROUGE-1 scores which reduces when averages for lower bigram overlapping score. The contextuality of the model is however greater and performed well in mapping longest sentence overlap giving a decent ROUGE-L score.

6.5 Discussion

As per the implementation of four different embedders, Word2Vec was relatively a weaker contextual embedder which did not take in the majority of the context into considerations. At the same time ELMo being a complex structure, only the pre-trained version was implemented however, it possessed a greater degree of contextual ability. Further the BERT and RoBERTa are the advanced contextual embedders that took into consideration a vast amount of contextual aspect during the embedding phase.

Word2Vec showed to have the least average ROUGE score compared to all the implemented word embedders with Avg ROUGE being 0.25. The precision score for all the three ROUGE scores were considerably low thus bringing down the average ROUGE. When the contextuality aspect of the embedder is relatively lower, the overlap of unigrams and bigrams would be extremely low resulting is lesser ROUGE-L as well. Further, considering the results from ELMo contextual embedder, the results do not vary significantly. The average ROUGE however shows an insignificant spike compared to Word2Vec - 0.26. Although for the test data the ROUGE-1 scores remain the same across Word2Vec and ELMo there was slight difference seen in ROUGE-2 and ROUGE-L. This represents that a slightly complex structure of pretrained embedding model did not greatly add to the cohesiveness of the summarized data.

Further on RoBERTa, a complex pre-trained model fine tuned on CNN-Daily mail dataset gave results relatively higher than both Word2Vec and ELMo. The ROUGE scores were much higher than both the previous models giving the average ROUGE score 0.31 which is largely that of the preceding word embedders. The ratio of precision to recall

is significantly larger throughout all the three ROUGE values in RoBERTa. The final contextual embedding model implemented was BERT. Here the results from BERT was significantly higher than all the other contextual word embedders with the average 0.41. There was a significant spike in F1 and Recall values of all the ROUGE metrics when compared to RoBERTa. The training set, test data and the training pattern were kept constant for both BERT and RoBERTa to observe a fair comparison. Although the parameters were similar, the finetuned BERT model showed higher average ROUGE score depicting that the there is a 41% overlap in the textual content on all ROUGE parameters such as unigrams, bigrams and longest sentences between BERT generated summary and the author generated summary validating its contextual cohesiveness.

7 Conclusion and Future Work

As the generation news data rapidly increases, the medium in which it is distributed becomes on of the key aspects in developing businesses associated with this domain. Summarization of news data is widely used in different usecases in news domain which has a greater potential in aiding these businesses. The foremost intent of the research was to identify the how the adoption of contextual embedders aids in the process of summarization. This research studies the architecture of different contextual embedders with the hierarchy of their complexity and embedding ability starting from a traditional embedding model Word2Vec to a slightly complex structure ELMo and finally the complex BERT and its variant RoBERTa. The results from the implementation of these models show that higher complexity and domain specific fine tuning does aid in picking contextual attribute from the data. It also gives a clear inference as to how these models rank in terms of their performance when used on news related data. This results from this research could be further authenticated by using higher degree of finetuning by tweaking the training parameters such as the learning rate, epochs and train data with higher computation resources. Also a comparison with summarization models that are pretrained on CNN-Daily Mail dataset from Hugging Face repository could be a valid approach to further substantiate the results of this research.

8 Acknowledgement

The author of this research would like to thank Dr. Christian Horn for the valuable insights and assistance provided in accomplishing this work. His constant support and expertise in the domain aided in bringing out the best from this research both technically and academically.

References

Bestgen, Y. (2019). Tintin at semeval-2019 task 4: Detecting hyperpartisan news article with only simple tokens, *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 1062–1066.

Bianchi, F., Yu, B. and Tagliabue, J. (2020). Bert goes shopping: Comparing distributional models for product representations.
URL: *https://arxiv.org/abs/2012.09807*

Büyüköz, B., Hürriyetoğlu, A. and Özgür, A. (2020). Analyzing ELMo and DistilBERT on socio-political news classification, *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, European Language Resources Association (ELRA), Marseille, France, pp. 9–18.
URL: *https://aclanthology.org/2020.aespen-1.4*

Chen, C., Park, C., Dwyer, J. and Medero, J. (2019). Harvey mudd college at semeval-2019 task 4: The carl kolchak hyperpartisan news detector, *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 957–961.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.
URL: *https://arxiv.org/abs/1810.04805*

Du, Y., Li, Q., Wang, L. and He, Y. (2020). Biomedical-domain pre-trained language model for extractive summarization, *Knowledge-Based Systems* **199**: 105964.
URL: *https://www.sciencedirect.com/science/article/pii/S0950705120302859*

Easwar, A. and Uthra, A. (2021). Automatic text summarization using word embeddings, *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 1065–1079.

Huang, G. K. W. and Lee, J. C. (2019). Hyperpartisan news and articles detection using bert and elmo, *2019 International Conference on Computer and Drone Applications (IConDA)*, IEEE, pp. 29–32.

Huang, Z. and Zhao, W. (2020). Combination of elmo representation and cnn approaches to enhance service discovery, *IEEE Access* **8**: 130782–130796.

Hürriyetoglu, A., Yörük, E., Yüret, D., Yoltar, , Gürel, B., Duruşan, F. and Mutlu, O. (2019). *A Task Set Proposal for Automatic Protest Information Collection Across Multiple Countries*, pp. 316–323.

Kieuvongngam, V., Tan, B. and Niu, Y. (2020). Automatic text summarization of covid-19 medical research articles using bert and gpt-2.
URL: *https://arxiv.org/abs/2006.01997*

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.
URL: *https://arxiv.org/abs/1907.11692*

Mastronardo, C. and Tamburini, F. (2019). Enhancing a text summarization system with elmo., *CLiC-it*.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space.
URL: *https://arxiv.org/abs/1301.3781*

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures.
URL: *https://arxiv.org/abs/1906.04165*

Naredla, N. R. and Adedoyin, F. F. (2022). Detection of hyperpartisan news articles using natural language processing technique, *International Journal of Information Management Data Insights* **2**(1): 100064.
URL: *https://www.sciencedirect.com/science/article/pii/S2667096822000088*

Pavlov, T. and Mirceva, G. (2022). Covid-19 fake news detection by using bert and roberta models, *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 312–316.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018). Deep contextualized word representations.
URL: *https://arxiv.org/abs/1802.05365*

Suryadjaja, P. S. and Mandala, R. (2021). Improving the performance of the extractive text summarization by a novel topic modeling and sentence embedding technique using sbert, *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, pp. 1–6.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention is all you need.
URL: *https://arxiv.org/abs/1706.03762*

Wang, K., Reimers, N. and Gurevych, I. (2021). Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning.
URL: *https://arxiv.org/abs/2104.06979*

Wang, Q., Liu, P., Zhu, Z., Yin, H., Zhang, Q. and Zhang, L. (2019). A text abstraction summary model based on bert word embedding and reinforcement learning, *Applied Sciences* **9**(21): 4701.
URL: *http://dx.doi.org/10.3390/app9214701*

Weng, S.-Y., Lo, T.-H. and Chen, B. (2021). An effective contextual language modeling framework for speech summarization with augmented features, *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 316–320.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q. and Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing.
URL: *https://arxiv.org/abs/1910.03771*