National College of Ireland

Abstractive Summarization of Multi-Documents using Fairseq

MSc Research Project
Data Analytics

Akash Senthil Kumar
Student ID: x21175641

School of Computing
National College of Ireland

Supervisor: Mr. Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Akash Senthil Kumar		
Student ID:	x21175641		
Programme:	MSc Data Analytics	**Year:**	2023-24
Module:	MSc Research Project		
Supervisor:	Mr. Hicham Rifai		
Submission Due Date:	14/08/2023		
Project Title:	Abstractive Summarization of Multi Documents using Fairseq		
Word Count:	7085		
Page Count	17		

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: Akash Senthil Kumar

Date: 14/08/2023

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):	

Abstractive Summarization of Multi Documents using Fairseq

Akash Senthil Kumar
x21175641

Abstract

Multi document summarization can save time browsing several sources of information . The aim of this research proposal is to perform multi document summarization where the objective is to summarize text obtained from different sources/documents and the summary generated by the model will be a combined summary of multiple the sources. Another objective is to minimise redundancy of information that might result from the process of multi document summarization. Especially, the same information might be repeated across documents.. NLP based sequence to sequence models are deep learning model that are becoming in generating summary. . Fairseq is used in this research project to generate the summary where the input is given from the multiple sources. Rouge score has been used as the evaluation metric resulting to Rouge 1:- 0.31, Rouge 2:- 0.08, Rouge L:- 0.15 are the obtained ROUGE Metric scores.

1 Introduction

Text summarization is the process of writing summary for a passage, document or a long paragraph. Nowadays as the language models evolve these jobs are done by the sequence 2 sequence models where the task is done quickly and efficiently. People have started to rely upon these AI tools to generate the summary for any type of texts and it is being used to all type of work from personal to professional. Till now there was a clear way to do a single document summarization but when it comes to multi document summarization there is no a correct methodology to follow and so performing this type of summarization can save time, energy and the summary will be effective consists of all the multiple sources. Generally, the summarization is of two types abstractive and extractive. The extractive summarization does the summary by picking up the exact words from the source and then only displays the key aspects or keywords present in the source and the other type is abstractive type of summarization where it recreates the source in a shorter form with different words and still not changing the meaning of the source. The NLP models are capable to both of the types and each of them will be used in different use cases. For this research project the abstractive approach has been chosen as the model will be able to rewrite the whole source in different words as the extractive approach will pick up the words from the source that has been gathered from all the documents and the end summary will be lengthier or if the length is controlled it can go meaningless. The motivation of this research project is the advantages when it is used in the real time applications where Abstractive multi document

summarization can be used in real time for a variety of tasks, including summarizing meeting minutes, reviewing patient history records, and summarizing several releases of application documentation. These may make use of multiple document summary in situations where it would be laborious to read through numerous papers in real time. The abstractive approach generates new vocabulary, making the substance of a lengthy paper more comprehensible. This multi-document approach can be utilized for academic duties like grading, effective teaching, and understanding tasks that may lighten the workload of the faculty if it is integrated with education apps.

The contribution of this research project will be summarization of multiple documents where the multiple documents are combined, and the redundancy has been handled through a similarity matrix that will be constructed later in this project and then further leading to model training and testing and also dealing with an active research problem where there is no correct or proper methodology to summarize multiple documents under a single architecture. The proposed methodology has the fairseq package developed by Meta Ai where its encoder, decoder, model can be custom built with lstm, bart, transformers and other leading hugging face pre trained models as well. (N *et al.*, 2022) posted an active research problem till date un their research paper and it leads to an active research question in this domain that "How well can fairseq summarize multi documents with multiple lengths and redundancy?".

Since there are not much related works for this multi document summarization there is an assumption that has been considered in this research to combine all the multi document that has been separated in the dataset. Since all the data from multiple sources are about a specific event (for a particular row) in the dataset and have a corresponding single summary they all will have same content causing redundancy of the data. This assumption is based on the literature review where the past papers that explain how to combine documents while handling the redundancy and other common errors that may occur while combining documents. There are multiple steps taken to deal with the redundancy of information as this will be the key aspect and the difference between single summarization and multi summarization.

Real time applications are evolving from this multi document summarization like it can be used for business-to-business organizations to summarize multiple invoices and also can integrate with a public website where users can just enter content from various sources and get a short gist of it in no time. Also it can be used in various domains like Legal Documents, Hospital Documents and much more.

The further structure of this thesis report will consist of related works, methodology of this research which will explain all the preprocessing and combining steps of the project, then it explains the model building process and the evaluation. Lastly, the reference section has also been included in this report for all the evidence-based steps that has been taken for this research.

2 Related Work

There are lot of related works present in abstractive and extractive summarization where it has been done using Machine learning, and deep learning algorithms. In terms multi document summarization there are some methodologies in the past that has been carried out.

2.1 Abstractive Summarization

(N *et al.*, 2022) has proposed both extractive and abstractive for the text summarization, It also explains about the different methodologies like attention function, encoder-decoder models, transformers and deep learning models. The paper proposed an abstractive approach and there was no proper explanation for the choice of abstractive summarization. The dataset used for this research is BBC news dataset and final model was T5 Transformers. However, there was no comparison, or any quantitative information found to validate the results. The future scope of this paper was to perform the same methodology for multiple documents. (Yan and Zhou, 2022) suggested a hybrid model which proposed both abstractive and extractive. The first step of this project is to cluster the sentences and for that K means clustering has been used and Cw2Vec has been used to vectorize and embed the sentences. The text rank which is extractive way has been fused with LSTM which is an abstractive model to generate summary. The proposed work has a slight increase in performance when compared with other algorithms when it was measured using the Rouge scores.

(Ngamcharoen, Sanglerdsinlapachai and Vejjanugraha, 2022) proposed a bidirectional LSTM to summarize Thai Language using keyword based method. The methodology of this paper is to extract keywords initially and based on the keywords the fine-tuned Bidirectional LSTM will be able to predict the generated sequence. Th authors suggest that instead of extracting keywords to formulate the sentence a cosine similarity matrix can be constructed to score highly similar sentences from the source and target summary and then train the model to generate the summary. The usage of cosine similarity has increased the model performance considerably.

(Singhal *et al.*, 2020) proposed an abstractive methodology to summarize the meeting conversations. Two models have been built in this project one is RNN based LSTM and the other model is Transformer based encoder-decoder. These two models have been built over this dataset and the author has proposed an Anaphora technique to identify whether the speakers in the meeting refer to other speakers in the conversation. Based on the rouge score evaluation the RNN based LSTM performance was not that great and had very low rouge score as well. The future direction of this paper is to use pointer generation and fine tune the transformer model based on pointer in the google dialogue dataset to evaluate its summarization. (Chen, 2022) proposed a deep learning model to perform abstractive summarization as discussed earlier this paper also discusses about the keyword based abstractive approach and to extract the keywords NER (Named Entity Recognition) model has been used and the model will pick up the impact words that can alter the meaning of the source then the decoder has been given to the deep learning model to generate the summary. The main objective of this research is to perform word segmentation and summarization on a

Chinese Dataset. The author concludes that deep learning techniques works well for the word segmentation but not for summarization. Still NER model was not able to outperform the transformers techniques and still needs to be fine tuned for the summarization tasks. (Sheik and S, 2021) has suggested a techniques for the abstractive summarization in deep learning which is neural network based where a pointer generator attention model. The author has trained the model on 80,000 court cases and have also generated around 4000 tokens for the purpose of training. The future direction of this paper is to create a fusion based deep learning model to generate summary.

2.2 Extractive Summarization

(Raundale and Himanshu, 2021) has proposed the same with TF-IDF and Text Rank to perform the extractive summarization. The paper uses TF -IDF to embed the words and uses Text Rank to rank and choose the sentences to display in the generated summary. Thus, this paper uses extractive summary. As the extractive approach is to choose the sentences from source by the way of scoring and then pasting it as a summary. It won't re phrase or rewrite the source. The evaluation metric chosen by this paper was Rouge Metrics to validate the results obtained from TF-IDF and text rank.

The next paper involves summarizing content from business documentations from (B and Abraham, 2022) and the authors have chosen to go with extractive approach as re writing technical reports of a business may change the meaning some times. The authors have also introduced the title feature extraction to grab the important topics or keywords present in the documentation. The authors have proposed LSTM to perform the summarization and in this case the inputs for the model are very long and the output should be small to get the gist of the whole documentation in a small amount of time. So, this project has chosen the extractive approach but to compare with the previous paper where the authors have used TF-IDF and text rank this has performed not well as the LSTM was also not fine-tuned for the longer inputs and shorter summary.

2.3 Multi Document Summarization Techniques

(R and K, 2021) has proposed cuckoo algorithm to search and find the optimal words that will be used to summarize. DUC dataset has been used to perform this technique. The author is conveying that multi document summarization is a broader spectrum of single document summarization. The methodology proposed by the author is start initially with the cuckoo search further goes to nest population after that they form a levy equation and then finally to the sentence selection part to generate the summary. The author explained why the necessary preprocessing techniques are essential to do like removing stop words, stemming, performing Regex Operations, word embedding and other necessary pre processing techniques that needs to be performed based on the dataset.

(Malik, Khan and Nawaz, 2023) proposed a new approach to minimize the repeated content when dealt with multiple documents. The problem author found is to subgraph mining tend to produce more repeated words and to solve this problem maximal approach has been chosen. The author has assumed the text as graphs and also subgraphs has been formed from

where the sentences will be selected. The future direction of this paper includes selection of sentences and also sentence ranking strategies in future multi document summarization projects. Converting texts into graphs to perform summarization has given a partial solution to solve the problem of redundancy while summarizing multiple documents.

(Sana and Akhtar, 2023)In this proposed model, these issues are addressed using two well-known natural language processing techniques: Bidirectional Encoder Representations from Transformers (BERT) and Gated Recurrent Unit (GRU). The Document Understanding Conference (DUC) dataset, has been used a well-known benchmark dataset for multi-document summarization, was used to train and evaluate the model's performamce. By using BERT to create contextual embeddings which will vectorize the words and GRU to record sequence for further passes. This study demonstrates how combining the BERT and GRU models can effectively capture the sequential patterns and context information in multi-document summarization, resulting in summaries of high quality that address the drawbacks of earlier methods. The author has concluded this research and there was an active problem faced and not been solved fully is the combining stage of all multiple documents and rather combining them explicitly the repeated content should be handled or removed as there was more repeated content int the generated summary of Bert. The author also suggest to use different transformer techniques that can capture the patterns in the text for further analysis and can have the ability to reproduce the words

2.4 Related Works Based on Cosine Similarity

Cosine Similarity is a technique to identify the similarity between documents to perform cosine similarity there needs to be necessary pre processing works done like vectorizing and embedding. (Hartanto, Pristyanto and Saputra, 2021) paper has explained the use cases of cosine similarity in the text analysis. The author is proposing multiple techniques to construct cosine similarity. To perform this cosine similarity Rabin Karp algorithm has been used to construct the cosine similarity. Author also explained various real time usages like it has been used plagiarism software and other copy detection software as well. The cosine similarity detects similarity between the documents, and it gives out the score ranges between 0 to 1.

(Fauzan, Atha Labib and Noor, 2022) have performed similarity analysis between Indonesian sentences and they have proposed cosine similarity to perform this task. They acquire the dataset and further they post tagging, tokenization, removing of stop words and then stemming. Later a cosine similarity matrix is constructed, and the similarity value is obtained from the matrix. The proposed cosine similarity methodology was able to identify the similarity between the sentences with more than 80 percent of accuracy. The author is suggesting to use this cosine similarity technique in various NLP project as there are more use cases lie with them.

(Zhang *et al.*, 2022) explains how cosine similarity works by measuring distance between two vectors and also explains how the cosine angle relies on the included angle which can be obtained after text vectorization. The proposed algorithm considers the change of each dimension of text similarity of the measured cosine distance. Authors claim that performance

of cosine similarity has much more increased. Future direction of this paper is to improve this algorithm based on other optimizing algorithm and tune it for more complex scenarios.

2.5 Fairseq Model

Based on the literature review and research the fairseq was built by facebook AI development team based on the transformers. (Ng *et al.*, 2019) Its an extensible NLP toolkit developed by transformers and can be actively used for text summarization and also for language translation. From the documentation of fairseq, it is confirmed that text summarization using fairseq is fully possible using fairseq based Bart (Bi Directional Auto Regressive Transformer). There are few difference between Bart and Fairseq Bart. The normal Bart is pre noised auto encoder where the model learns from the corrupted version of the data kind of like an unsupervised learning whereas fairseq Bart comes with the fairseq framework that allows researchers to fine tune for a specific need. Creators suggest to use to use fairseq bart as the fairseq toolkit is performing well in the language translation and summarization as well with other algorithm. Meanwhile the authors have also mentioned in this research journal that fairseq based lstm encoder-decoder can also be used to generate text from training. However, there is a clear support for fairseq based bart over fairseq based lstm in this paper and also in the documentation present for this fairseq toolkit provided by facebook.

2.6 Choice of Evaluation Metrics

Based on the literature review there are two evaluation metrics one is (Chatoui and Ata, 2021) Rouge and other one is Bleu. All the papers used rouge scores in the past and it has more types of metrics as well. It is one of the best evaluation metric for text summarization whereas, BLEU score is mostly used in machine translation tasks. So, to conclude the Rouge metrics has been finalized for this research.

After extensive literature review handling redundancy before feeding the data into the model handling redundancy has been an active research problem while combining the text data and authors in the past have recommended to use more transformer-based models to perform the summarization. Fairseq built based on the transformer has been used in this research as it is built based on the transformer architecture. Also, Rouge metrics has been chosen in order to evaluate the generated summary.

3 Research Methodology

Multi document summarization will follow the KDD approach which is Knowledge Discovery in Databases.

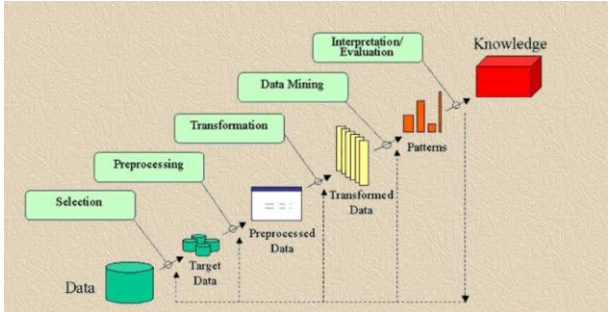

Figure 1 KDD Methodology

The above figure explains the project's methodology and approach where at the last step is gaining knowledge at then end. NLP project to perform summarization and has a separate evaluation methods like ROUGE and BLEU scores.

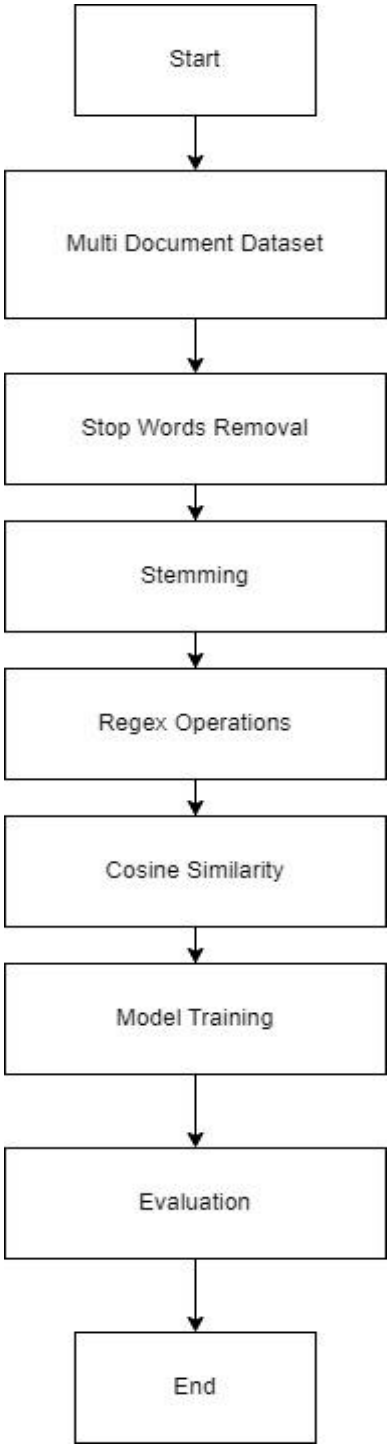

Figure 2:Methodology flowchart

There are about 5 steps covered in this research starting from Data Gathering & Data Combining, Text Cleaning, Combining Documents, Model Building, & Training and Evaluation.The dataset for this research has been downloaded from the (Fabbri *et al.*, 2019) ArXiv dataset paper. The data downloaded were 4 files namely, train.src, train.tgt, test.src,

test.tgt where src files contain the source text and tgt files contain the target text which is the summaries of those src files. The src file has the multi document separation within each row of the dataset by this '|||||' symbol. Both the src and tgt files of the train are loaded to a jupyter notebook and with the help of pandas, a dummy variable was created on both of the src and tgt files and then those two data frames has been joined with the help of that dummy common column. Now the final data frame has been saved as a csv file for further cleaning. The same procedure has been followed for the test src and tgt file of test data.

3.1 Text Cleaning

Text Cleaning is the data preprocessing phase of this research, and it includes various steps that has been taken from the evidence collected from plethora of research papers. The first step that has been taken is removing of stop words where the most occurred/repeated words has been removed where in those words does not have any meaning nor change the contest of the sentence. In this dataset 'is', 'in', 'a', 'an' kind of words has been removed as it has no value-added contribution towards the overall meaning of the sentence. By performing this, the dimension of the text data has been reduced and helped in the representing the easy to interpret text data leading to reduce the processing time when it fed has input to the model. It helped to focus on the main content for the model to generate output rather than compiling all the stop words and increasing the processing power while training. This technique has been actively used in all the NLP research tasks where these stops words have no use case so removal of them can be beneficial.

Text Normalization is the next step in text cleaning. To perform the text normalization stemming method is used, which will minimize the words to their base root and scale down the different variations of a particular word also considers different tense form of a word. In this dataset there are more variations of several words which conveys the same meaning like use – 'used', 'using', 'will use', happen- happening, happened, will happen, was happened. By performing stemming the vocabulary size has been reduced in the dataset and the number of unique words has been reduced. If a model is trained on a stemmed data, it can easily identify the different form a word while performing the inference. So, the text normalization helped to reduced the overall dimensionality of the dataset while not compromising the actual meaning of the text. The port stemmer function has been used to apply stemming to the training data using the lambda function. Stemming is a heuristic approach and should be used properly, for this project training data was not over stemmed nor less stemmed so that there is no change in the text and it doesn't change the meaning of the sentences. Simple stemming has been used so that dimension of the data is not fully reduced, and the data is ready for further preprocessing stages. Stemming also considered the morphological and parts of speech of the dataset to not to change the meaning and the logical order of the sentences.
Removing the unwanted email id's and website links is the next important step in text preprocessing of this research. Email id's and websites are usually not important in the summarization task and it does not provide any information to the text data that needs to be summarized and might also distract the model to produce something. Both email id's and websites does not come under natural language components so including them may affect the overall coherence, inference and quality of the model generated summary. Any summarization algorithms will try to analyse the patterns present in the text to generate sequence in a shorter form, now the presence of emails and website will change the patterns as it has its own instance and the algorithm will consider it as another pattern resulting in changing the context of summary produced by model. If the email ids and the websites are present in the training and testing data particularly specific to domain, it may change the

whole summary into a biased summary. Removing this help, the model to produce a neutral summary and analyse all the patterns present in the text. To perform this removal of email id and websites REGEX has been used in python. A custom regex function has been created and then applied on both training and testing data. Regex function can be used to perform text editing operations like manipulations, text matching, and pattern matching.

The next step in the preprocessing is removing special symbols like parentheses, hyphens, and other special symbols. This removal has also been done using regular expressions where a custom function was created and then applied to the training and testing dataset.

In the dataset there was this unusual word found in two cases which is 'newline_char' and 'NEWLINE_CHAR' after several checks and analysis it was found that this particular wors was repeating and spread all over the dataset in between the words and it was a minor kind of an error that occurred in the data collection phase. So, to remove that again a regex function has been specifically created and applied to the training dataset to remove them. Based on the literature review conducted these were all the techniques that has been used for all the NLP task in the past and based on the initial impressions on this dataset the above preprocessing techniques has been chosen and implemented on the dataset. Since the size of the training data is really big in size and as well in the dimension, the data has been reduced for further phases of this research.

3.2 Combining Documents

In multi document summarization the most crucial part of this research is to combine those multiple documents into a single input so that it can be fed to the model as input. However, the problem here on what basis to combine the documents? To answer this question the multi documents are separated '|||||' but to have a note all the multiple documents are about a same topic for each rows and they have a one single summary. All the content that has been separated as different documents talks about one single topic or event happened so the main problem here to handle is the redundancy of the data. So, feeding the model with non-redundant data will train the model to deliver a quality summary. To perform this cosine similarity matrix has been chosen. Cosine Similarity is the way to compare two vectors through a multi-dimensional space. It treats each vector as a representation of a sentence or a document. Each dimension of a vector will correspond to an attribute, in the vector space formed by the cosine matrix. To formulate cosine similarity, first the texts are transformed into numerical vectors with the help of TF-IDF vectors this works with word frequency counts of the words. The next is calculating cosine similarity for the created vectors has a formula, which will be

Equation 1 Cosine Similarity

$$(A, B) = \frac{(A, B)}{\|A\| \cdot \|B\|}$$

Where, A and B are the two vectors to which the cosine similarity will be calculated A.B are the product of the two vectors. ||A|| and ||B|| are the lengths of the two vectors or can be called it as Euclidean Norms. ||A|| . ||B|| are the product of two Euclidean. The value of the cosine

similarity will be ranging in between -1 to 1. If the value is near to 1 it means that sentence/ words are more identical to each other and if the value is near to zero it means the similarity between the sentences are less and there is more unique and non identical content in the input. However, that can also mean that text are not that similar and may also have opposite directions of opinions and conclusions. These scores can be called as Similarity Ranking where it can be used to rank the sentences.

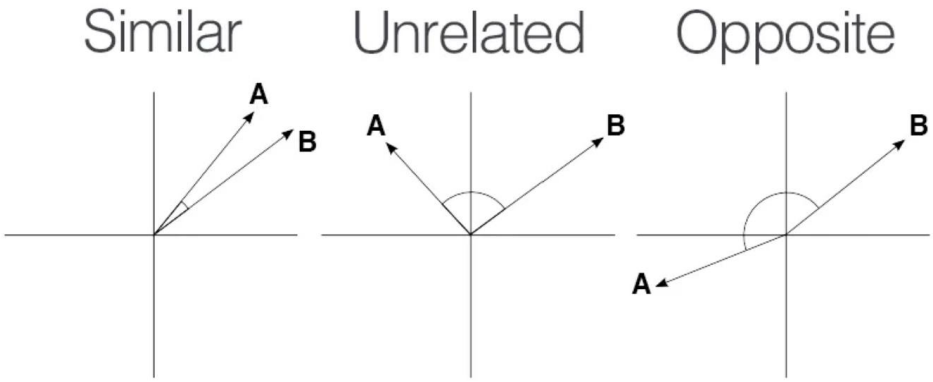

Figure 3 Cosine Similarity Vector Representation

The above graph figure will explain how the vectors are similar with each other, in a 4 dimensional graph, the acute angle represents that values are somewhere in between positive values and if the angle slightly go off from the acute and lie in between acute and obtuse then the sentence can be unique or may be unrelated to the previous sentences. That means the values can lie between negative and positive and it indicates that there might be unique information present in the combined data. If one of the vector value goes to the fourth quadrant then the content is totally opposite and it does not show any relativity that result into no point of summarizing it.

All those converted vectors based on TF-IDF vectorizer are transformed into a similarity matrix and matrix computes the similarity between the sentences and a separated list is created to store the non-redundant sentences. When the function loops over each sentence in the similarity matrix and then it compares with the previous sentences to check the similarity and the threshold has been set to 0.9. If the value exceeds the threshold value, then it is considered as high similarity and then those sentences are removed from the data. If the value is considerably lesser than the threshold value, it is considered that the current sentence differs in providing information when compared to the previous sentence. Also created a function for all this so that every time the loop runs, the function will be called to remove the redundancy. Then all of the results are replaced into the same 'src_content' of the dataset and the data frame has been saved into csv file.

The similarity matrix output that was obtained while calculating the cosine similarity matrix. As you can see the above matrix where the numbers lie between 0 to 1 and then each row of the matrix represents each row in the dataset. For each row in the dataset the matrix will be created and based on the similarity it is decided whether to keep the sentence or not to keep.

3.3 Model Defining & Training

Fairseq Models has been used to perform the summarization for this research. From the literature review it is found that fairseq has a lot of nlp models but specifically for the

purpose of translation and summarization there are two models that has been chosen as two experiments. One is custom fairseq model with LSTM encoder and decoder other one is again custom fairseq model with BART encoder and decoder. Both of the models are fine tuned for the summarization purpose. For training, the epochs methods have been used for both models and then the results are evaluated. This is the last step of this project, and the training has been done for two models and the results are analysed.

4 Design Specification

The fairseq models are built from the transformer-based encoder and decoders. They both share the same architecture for the model. These encoder and decoder can be fine tuned for a custom purpose.

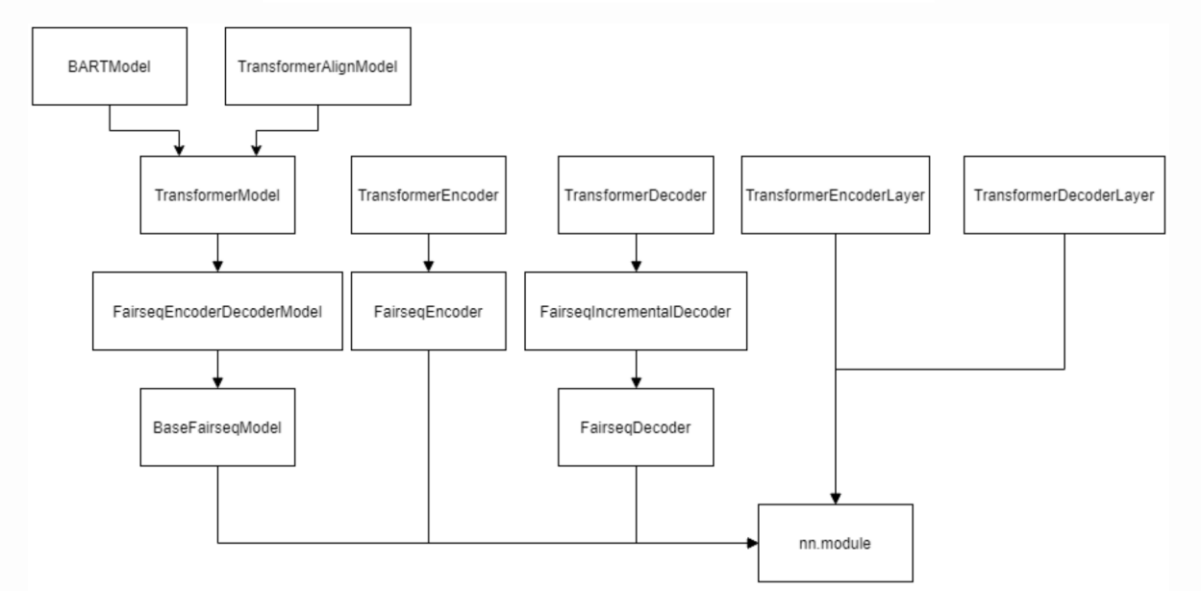

Figure 4 Flowchart of Fairseq

A component of the Fairseq architecture, the Fairseq encoder which is same as the transformer encoder, is in charge of processing input data, frequently text, and converting it into a variety of context-rich representations that capture the pertinent information in the input. The encoder is essential in identifying key characteristics from the source data that the decoder can utilize to build the target sequences in the context of sequence-to-sequence activities like machine translation or text summarization. This proposed transformer-based fairseq encoder processes input sequences by employing several layers of feedforward and self-attention neural networks. By accurately extracting contextual information and features from the input text, the decoder is set up to output meaningful sequences in a variety of sequence-to-sequence tasks for the text summarization process. The Fairseq decoder, a crucial part of the model, collaborates with the Fairseq encoder to carry out tasks for text summarization. The target sequences are incrementally created by the decoder using the context-rich representations from the encoder results. The Fairseq decoder, which has been constructed d using a transformer architecture, will be able to produce the target sequences using the context-rich representations that the encoder has provided. It captures dependencies and links for a variety of sequence-to-sequence actions using mechanisms including self-attention, cross-attention, and feedforward networks, producing coherent and considerable output sequences. The next in the flowgraph is base fairseq model to elaborate this A general-purpose neural network design is typically part of the basic Fairseq model, which may be fine tuned and customized for this research tasks which is text summarization. Typically,

transformer models are the base of this design. By offering pre-implemented, highly optimized designs that are available for tweaking, this technique significantly reduces the time when attempted for training.

Now the finalized model Bart (Bi Directional Auto Regressive Transformers) which is also included in the architecture diagram and it belongs to the fairseq. For several Natural Language Processing issues, Facebook AI Research (FAIR) created the BART (Bidirectional and Auto-Regressive Transformers) sequence-to-sequence model architecture. BART combines the benefits of both auto-regressive and bidirectional techniques. BART is based on the transformer architecture and is a part of the Fairseq library. Its popularity and use case have increased because it can provide ground-breaking results in the field of text summarization. BART also uses the same encoder and decoder used by the transformer models. BART can integrate both bidirectional and auto-regressive learning, making it ideal for a wide range of applications. Its pretrained representations can be tuned using task-specific data to produce impressive results, particularly for tasks involving the creation and manipulation of sequences. By successfully fulfilling a variety of tasks, including as text summarization, text generation, BART has shown its adaptability and performance in past based on literature review. This is the overall explanation for the Bart using fairseq encoder and decoder and the reason why fairseq custom model was not finalized for this research will be explained in the experiment section of this research.

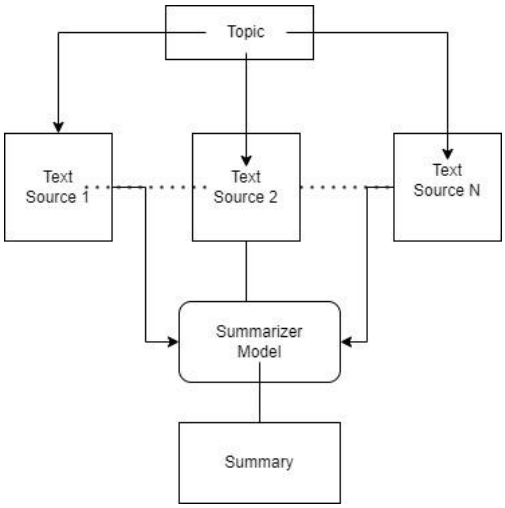

Figure 5 High Level Architecture Diagram of the Project

The above figure will illustrate the overall idea and real time working of this project where multi document summarization is done for a single event happened or about a single topic where it has multiple sources and need to summarize it. If a application is developed on top of this sequence model this will be an overall understanding that will be given to the end users. These are the design specifications of this research project and in next section last part of the implementation will be discussed.

5 Implementation

Fine tuned fairseq Bart has been used to summarize the combined documents to produce a summary. Bart tokenizers has been used to tokenize the words and further those tokenized words has been converted into tensors. So the tensors will be given as input to the input. A data loader function has been created with the batch size of 4 for the model input. The

maximum squeeze length has been set to 512 for the model and as the model was trained in a local pc with CPU and so it becomes CPU based Bart. Number of epochs has been set to 1 as the model takes too long to train for 1 epoch and after training the total loss from the training was 0.02. And the model has been saved with its tokenizer as well so that whenever custom input has been given the same tokenizer.

The BART tokenizer, a part of the BART model architecture, is used to preprocess and tokenize input text data before it is fed into the BART model for various natural language processing tasks. A fundamental step in NLP is tokenization, which includes breaking the text into smaller units which is called tokens, which can be words, subwords, or characters. The input text is handled correctly by the BART tokenizer and put into a format that the BART model can understand and interpret. The BART tokenizer is intended to provide reversible tokenization. Since tokens can be converted back into language that can be understood by humans, they are valuable for activities like producing writing. Fixed-length sequences are widely applied in BART models. For sequences to be the proper input length, the tokenizer helped to pad or truncate them. In order to provide effective batch processing during training and inference, padding makes sure that every sequence is the same length. This made easy to preprocess the data and tokenize text thanks to the BART tokenizer's smooth inclusion with the Fairseq library.

6 Evaluation

The results have been evaluated using Rouge Scores based on the literature review conducted in the previous sections the rouge score has been chosen as the best evaluation metric as the accuracy and the test loss will not convey how the model works with the custom input data. From the rouge scores the Rouge 1, Rouge 2, and Rouge L has been filtered out as these three metrics will be able decide the model's performance.

The formula for Rouge 1 will be,

Equation 2 Rouge 1 Formula

$$Rouge\ 1 = \frac{Count\ of\ Overlapping\ Unigrams}{Total\ Count\ of\ Unigrams\ in\ Reference}$$

Where Unigrams would be words present in the sentence to explain it in a definition 'An element from a given space or a passage can be considered as Unigram'.

The formula for Rouge 2 will be,

Equation 3 Rouge 2 Formula

$$Rouge\ 2 = \frac{Count\ of\ Overlapping\ Bigrams}{Total\ Count\ of\ Bigrams\ in\ Reference}$$

Where bigram would be two consecutive words present in the sentence. 'Consecutive Two Elements present in a space, or a passage can be considered as bigram.'

The formula for Rouge L will be,

Equation 4 Rouge L formula

$$Rouge\ L = \frac{Length\ of\ Longest\ Common\ Subsequence}{Total\ number\ of\ words\ present\ in\ the\ sequence}$$

Rouge L is different from the above two metrics, it records the length of longest sequence that conveys the same meaning and takes up the total number of present in that particular sequence.

Usually, the rouge scores will be valued from 0 to 1 which indicates the percentage of overlapped words present in the generated summary and the reference summary.

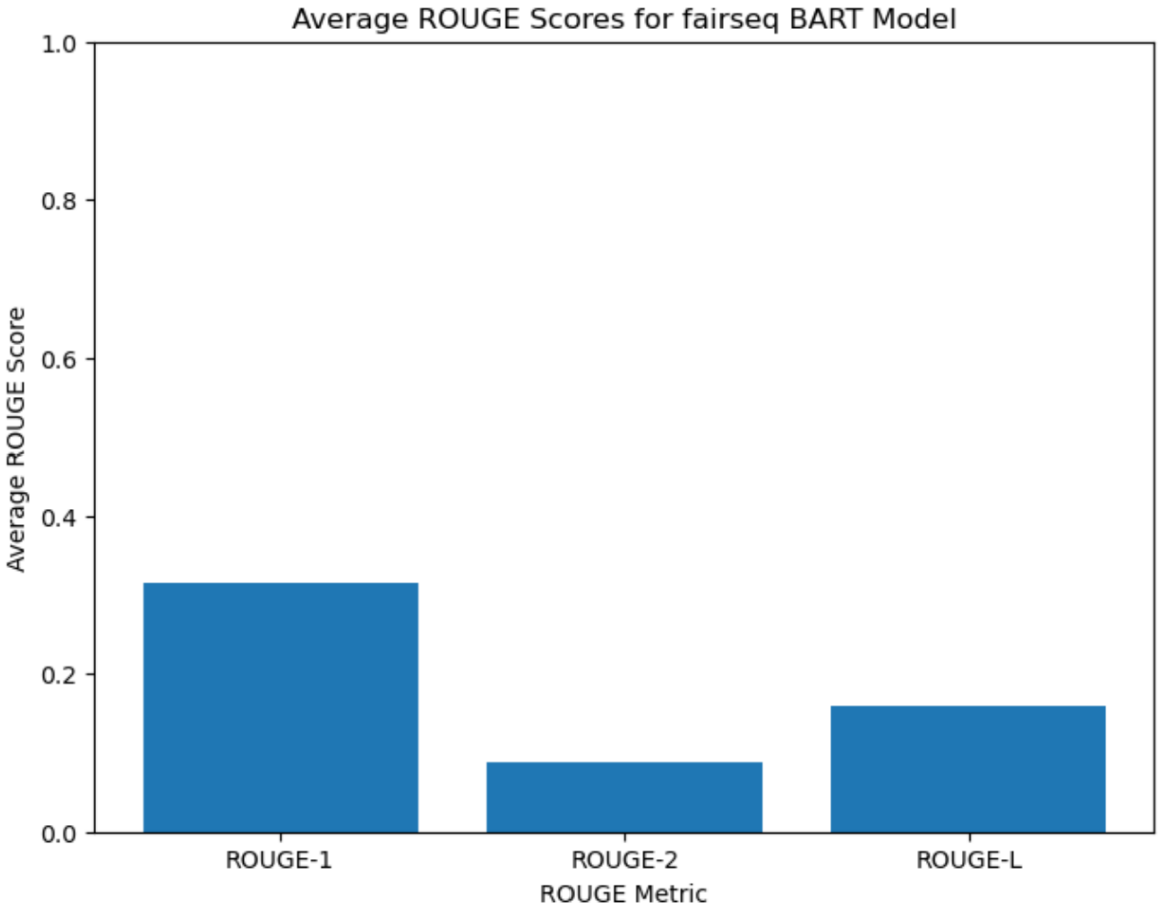

Figure 6 Rouge Metrics Graph

Table 1 Rouge Scores

Rouge Metrics	Scores
Rouge 1	0.31
Rouge 2	0.08
Rouge L	0.15

The above rouge scores are the result that has been obtained from the fairseq Bart model. To interpret the Rouge 1 score, the average amount of 31 percentage of words overlap in the generated summary and the reference summary present. While this amount of overlap should

be good when the model was been able to capture the key points from the reference summary.

To interpret the Rouge 2, The summary has 8.7 percentage of bigrams word that match or overlap with the reference summary. From this it is understood that model is limiting the usage of bigrams present in the generated sequence from the reference summary.

To interpret the Rouge L, the summary has 15.9 percentage of longest subsequence sentences share with the reference summary. This can be further analyzed as how well my summaries captured the structure and layout of the source content and how it is differing from the reference summary.

6.1 Experiment 1

Initially, based on the literature review and fairseq documentation two fairseq models one is fairseq based Bart and the other one is fairseq based LSTM encoder-decoder. A model was able to train with LSTM encoder and decoder based on fairseq but unfortunately the generated sequence from lstm failed to decode. Multiple decoder has been tried, used and defined with the model like greedy decoder and Beam Search decoder but the tensors produced by the model was not able to decode into sequences this is because model was not able to generate the tensors that was present in the training data. This experiment was a trial and error based on the documentation of fairseq. Although, fairseq based Bart was a successful experiment and it has been finalzed as the best model for this research while this one was not able to generate the model's known tensor and hence the summary was not generated in this experiment. The code also will be attached for any further reference for this experiment.

6.2 Experiment 2

Before using cosine similarity, K means clustering was in the option to cluster the relevant sentences or documents. However, the K means clustering run time was too long for the size of the dataset. And the alternative for this was the cosine similarity. And based on the cosine similarity the multiple documents have been combined.

7 Conclusion and Future Work

To conclude this research multi document summarization has been using fairseq and to answer the research question 'How well can fairseq summarize multiple documents with different lengths while handling redundancy ?' Yes fairseq can summarize multiple documents where 31 percentage of words overlap with the human written summary , 8.7 percentage duo words combo match , and 15 percentage of long sequence match in the generated summary. The redundancy is also been successfully handled after pre processing of text and the model was able to successfully generate the summary sequence. Fairseq toolkit handled this summarization task very well and if trained for longer hours can it can produced ground breaking results. Tokenizers offered by the toolkit also performed as well as the fairser bart based decoders

In the future, this whole jupyter notebook implementation can be deployed as real time software application where when the user enter the information first phase would be redundancy removal using cosine similarity and next phase would be generating the summary. Also, in this research due to time constraint and resources availability cpu based

model has been used but generally a gpu based model training will have good performance. And the training period should be longer for more better results. There can be a customizable output for the users to decide how long should be the output.

References

B, F. and Abraham, S. (2022) 'NLP Based Automated Business Report Summarization', *IEEE, 2022 International Conference on Innovative Trends in Information Technology (ICITIIT)* [Preprint]. Available at: https://doi.org/10.1109/ICITIIT54346.2022.9744151.

Chatoui, C. and Ata, O. (2021) 'Automated Evaluation of the Virtual Assistant in Bleu and Rouge Scores'. Available at: https://doi.org/10.1109/HORA52670.2021.9461351.

Chen, Y. (2022) 'Research on Abstractive Summarization Technology Based on Deep Learning', *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)* [Preprint]. Available at: https://doi.org/10.1109/CVIDLICCEA56201.2022.9824030.

Fabbri, R.A. *et al.* (2019) 'Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model', *ARXIV* [Preprint]. Available at: https://doi.org/arXiv:1906.01749v3.

Fauzan, R., Atha Labib, M.I. and Noor, S. (2022) 'Semantic similarity of Indonesian sentences using natural language processing and cosine similarity', *IEEE* [Preprint].

Hartanto, A.D., Pristyanto, Y. and Saputra, A. (2021) 'Document Similarity Detection using Rabin-Karp and Cosine Similarity Algorithms', *IEEE* [Preprint]. Available at: https://doi.org/10.1109/IC2SE52832.2021.9791999.

Malik, R., Khan, K.U. and Nawaz, W. (2023) 'Maximal gSpan: Multi-Document Summarization through Frequent Subgraph Mining', *IEEE, 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM)* [Preprint]. Available at: https://doi.org/10.1109/IMCOM56909.2023.10035618.

N, Balaji *et al.* (2022) 'Text Summarization using NLP Technique', *IEEE,International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics* [Preprint]. Available at: https://doi.org/10.1109/DISCOVER55800.2022.9974823.

Ng, N. *et al.* (2019) 'FAIRSEQ: A Fast, Extensible Toolkit for Sequence Modeling', *Arxiv* [Preprint].

Ngamcharoen, P., Sanglerdsinlapachai, N. and Vejjanugraha, P. (2022) 'Automatic Thai Text Summarization Using Keyword-Based Abstractive Method', *2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)* [Preprint]. Available at: https://doi.org/10.1109/ISAI-NLP56921.2022.9960265.

R, S.S. and K, .Dr.Arutchelvan (2021) 'Improved Cuckoo Search Optimization Algorithm based Multi-document Summarization Model', *IEEE,2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* [Preprint]. Available at: https://doi.org/10.1109/ICCMC51019.2021.9418473.

Raundale, P. and Himanshu, S. (2021) ‘Analytical study of Text Summarization Techniques’, *IEEE,2021 Asian Conference on Innovation in Technology (ASIANCON)* [Preprint]. Available at: https://doi.org/10.1109/ASIANCON51346.2021.9544804.

Sana, E. and Akhtar, N. (2023) ‘Improving Multi-Document Summarization with GRU-BERT Network’, *IEEE* [Preprint]. Available at: https://doi.org/10.1109/REEDCON57544.2023.10151372.

Sheik, R. and S, Dr.J.N. (2021) ‘Deep Learning Techniques for Legal Text Summarization’, *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* [Preprint]. Available at: https://doi.org/10.1109/UPCON52273.2021.9667640.

Singhal, D. *et al.* (2020) ‘Abstractive Summarization of Meeting Conversations’, *2020 IEEE International Conference for Innovation in Technology (INOCON)* [Preprint].

Yan, J. and Zhou, S. (2022) ‘A Text Structure-based Extractive And Abstractive Summarization Method’, *IEEE,2022 7th International Conference on Intelligent Computing and Signal Processing* [Preprint]. Available at: https://doi.org/10.1109/ICSP54964.2022.9778497.

Zhang, J. *et al.* (2022) ‘Text Similarity Calculation Method Based on Optimized Cosine Distance’, *IEEE* [Preprint]. Available at: https://doi.org/10.1109/ICEDCS57360.2022.00015.