Indian Institute of Technology

Jodhpur

Project Report

CSL7340 - Natural Language Processing

Topic: Abstractive Text Summarization

Submitted by:

Kopal Rastogi [P20CS205]
Mayank Gupta [M20MA005]

May, 2021

Table of Contents

Part A: Literature Review

PART A: Literature Review

1. Introduction

In today's era, information plays a vital role in one's daily routine, of which maximum times it is in textual format. Dealing with it requires an enormous amount of textual corpus and efficient systems. Therefore, it is convenient to summarize the work efficiently so that its context remains untouched and saves time. In this scenario, designing automatic summarization techniques comes into the foreground to meet the demands and needs of users.

Therefore, in this framework, we discuss the current trends and the scientific literature that target the above-mentioned problem. In particular, we selected the recent research that aims at neural network-based abstractive summarization. We define a qualitative analysis using a concept matrix, depicting the trends in neural abstractive summarization techniques in vogue. However, the methods involving encoder-decoder-based transformers established a new milestone in natural language text processing. Grounded on the inference made from the in-depth survey, using pre-trained language models and deep neural networks is explicitly preferred for the abstractive text summarization tasks.

2. Text Summarization: A Natural Language Processing Task

In the current scenario, there is a significant surge in the amount of textual data. Moreover, the maximum of the data we produce is in digital format, and it is multiplying continuously. Automatic summarization methods deliver a convenient way to manage lengthy textual data effectively and efficiently in a time-efficient manner. Text summarization targets to generate summaries from lengthy texts that are concise and comprehensive, and at the same time, they retain the context in the topic.

Some applications of text summarization are:

1. Generate news headlines from the news articles that help in the search process and browsing.
2. Summarize the research articles to focus on their research rather than reading an irrelevant paper entirely.
3. Lawsuit abstraction
4. Clinical and Biomedical data abstraction

The automatic text summarization is done in two flavors, viz.,

1. Extractive summarization
2. Abstractive Summarization

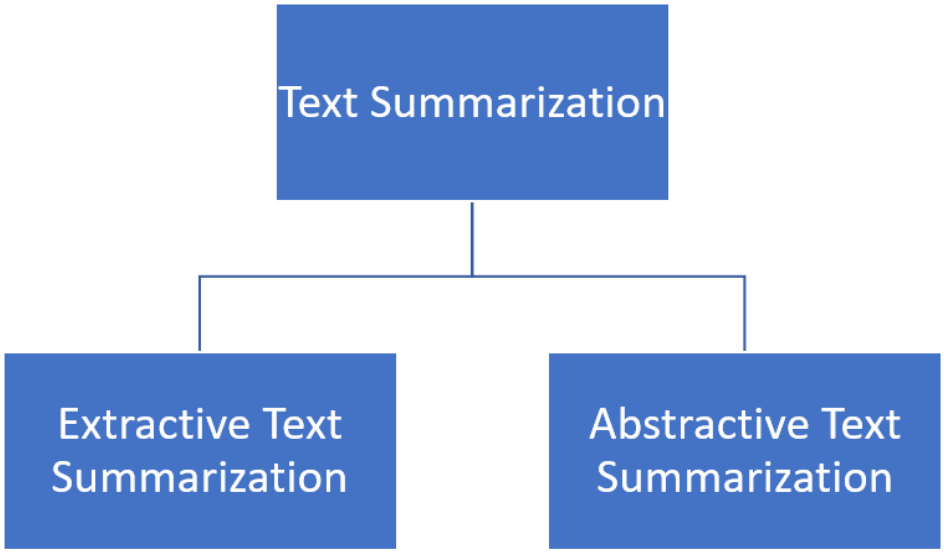

Fig. 1: Taxonomy of Automatic Text Summarization

2.1 Extractive Summarization

When modeled using extractive techniques, the main sections of the text are extracted based on some scoring criteria and then concatenated to produce the summary.

2.2 Abstractive Summarization

Abstractive techniques are a bit more complex and challenging to work with because the text is paraphrased to generate a summary having words different from the original text.

2.3 Extractive v/s Abstractive: Comparing and Contrasting

All summarization methods and models, whether extractive or abstractive, share the common purpose of generating summaries that are fluent, non-redundant, and coherent. Both approaches can be employed to generate summaries across either a single source document or multiple source documents. In the case of one source document, the summarization system produces a short outline of the document. When a summary is generated across multiple documents, readers can familiarize themselves with information through several documents related to the same topic in relatively less time.

3. Theoretical Background

Some of the very early approaches to automatic text summarization leveraged statistical models to select and copy the most important words from the text. However, these models were not capable of generating new text or paraphrasing text because these models were not capable of understanding the context or the meaning of these words. Most of the earlier research focussed on extractive summarization, and then the trends gradually shifted towards abstractive methods.

Later on, with the increasing popularity of deep learning-based approaches and their applicability in a wide range of domains, the text summarization method focussed on artificial neural networks and their variants, such as,

1. Artificial neural networks
2. Convolutional neural networks
3. Language models
4. Sequence to sequence models

4. Recent Trends in Abstractive Text Summarization

4.1 Abstractive text summarization based on deep learning and semantic content generalization (Kouris et al., ACL-2019)

4.1.1 Introduction and Summary

The paper defines a novel framework to solve the inherent challenges in abstractive text summarization grounded on the appropriate amount of combination of the deep learning methods and semantic data transformations. This work proposes an elegant theoretical model for generalizing the text semantically, focusing on the encoder-decoder-based method to generate a concise and clear summary. In conjunction with this, a concrete method for transforming these summaries into a human-readable format is proposed. The essential characteristic of this method is that it retains the inherent context and the original information piece while dealing with the challenges associated with out-of-vocabulary (OOV) words or unknowns.

4.1.2 Model Architecture

The model consists of the following components:

4.1.2.1 Text generalization

Text generalization defines that the taxonomy of concepts is well-defined and can easily be mined from the text. The key ingredients behind the text generalization are:

1. Concepts; and
2. Hypernyms defined in a hierarchical fashion

Moreover, it is evident that the extraction of ordered sequences from the taxonomy tree leads to text generalization. This idea is depicted in the figure below:

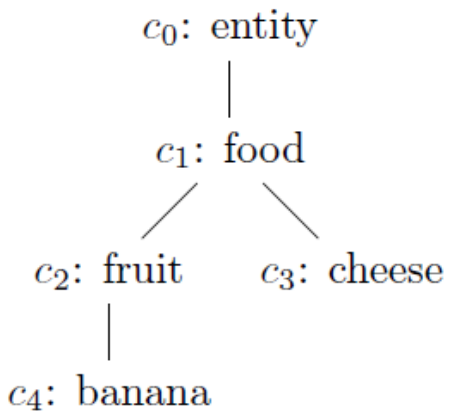

Fig. 2: Taxonomy Path

4.1.2.2 Text generalization strategies

The basic idea about text generalization is to view the frequent patterns in the concept within the source text. This emerges from the fact that machine learning models require an ample amount of examples for training for making accurate predictions. Therefore, high-frequency hypernyms are better choices for consideration than low-frequency terms as they convey the text's fundamental meaning.

4.1.2.2.1 Named Entities-driven Generalization (NEG)

As the name signifies, NEG generalizes the named entities such as person, place, organization, or any other proper noun. For example,

“Trump has been in Chicago” -> (generalization) -> “_person_ has been in _location_”.

The key factors in named -entities recognition are:

1. Input text;
2. Taxonomy of the concepts, denoted by T;
3. Extracted concepts C and taxonomy paths;
4. Set of named entities E; and
5. Threshold for defining the frequency of the concepts

4.1.2.2.2 Level-driven Generalization (LG)

In a similar spirit, the level-driven generalizations define the generalization rules depending upon the level of the generalization. Also, for LG, the key factors are:

1. Input text;
2. Taxonomy of the concepts, denoted by T;
3. Extracted concepts C and taxonomy paths;
4. Threshold for defining the frequency of the concepts; and
5. Level of generalization, denoted by d

The set of concepts C is regularly updated by merging the c_i's along with its hypernym with every step of generalization thus, creating an extensive network of named entities.

4.1.2.3 Predicting Summaries

The text generalization defines the pre-processing task that occurs just before the model is trained. The model for training is based on an encoder-decoder framework whereby the encoder consists of the Bi-directional LSTM, and the decoder consists of a unidirectional LSTM with multi-head attention. However, the words follow the popular word embeddings using the Word2Vec approach. The training is done in a supervised fashion using article-summary pair.

4.1.3 Why did the proposed architecture work?

1. The deep learning models have achieved state-of-the-art status, and since the results of the abovementioned architecture are generalized, the specific sense out of the words is easy to specify. Moreover, the use of hypernyms for each candidate token helps in the replacement of the generalized concepts.
2. Since the model incorporates both the named entity generalization and level-driven generalizations, it fits the real-world problems efficiently. In the case of LG, checking for the taxonomy path for a token is unimportant, leading to the sense of replacement. In the same spirit, for NEG, the tokens are replaced using the person, location, or organization relationship using the taxonomy path.
3. Finally, a significant surge in the model performance is observed due to the similarity function. Since it is a hyperparameter, one can select from a wide range of similarity functions ranging from simpler ones (Jaccard or Cosine) to complex ones (Word mover distance or Levenshtein distance measures).

4.1.4 Results and Analysis

1. The method is tested against the two famous datasets Gigaword and DUC 2004. The data is preprocessed, and duplicates are removed. Further, punctuations and longer and shorter summary elements are also removed.
2. The dataset is normalized by expanding the abbreviations such as **I’ve = I have.**
3. The method is tested against 3 million article-summary pairs with around 10 million words.
4. The average article and summary length is 28:9 and 8:3 words, respectively.
5. 4k pairs are randomly selected for testing the performance of the model.

6. Also, the method is compared and contrasted with ABS+, RAS-Elman, words-lvt5k-1sent, and GLEAM for DUC 2004 data.

7. The performance of the approach is evaluated using ROUGE metrics (ROUGE-1, ROUGE-2, ROUGE-L).

8. In particular, F-measure for Gigaword and Recall for DUC 2004 dataset is defined.

9. For LG strategy, the best performance is achieved while generalizing the words having at most 100 occurrences. Similarly, for the NER strategy, the best performance is achieved with f=500 for Gigaword and f=1000 for DUC 2004.

10. LG strategy is more efficient in improvising the potential of the deep learning models while generalizing the low-frequency words.

11. The NER strategy is more fit in the case of the named entities because, in general, the words describing the named entities have a specific function within the text. Moreover, their reduction leads to accuracy.

4.1.5 Limitations and further improvement

1. Intuitively, certain features of the projected method can be extended. In the current scenario, the generalization of nouns is kept in focus. All other parts of speech, such as verbs, can be included.

2. Ambiguity in word sense and context is a significant challenge in natural processing tasks; addressing and managing it using the taxonomical paths for text can be pretty interesting.

3. Lastly, the performance of deep learning models can be increased using separate semantic representations for each token/word.

4.1.6 Conclusion

Although deep learning-based abstractive text summarization techniques are quite prevalent in the current scenario, the study involving the combination of structure-based and semantic-based strategies is in demand. In this direction, the proposed method discusses a novel approach to address the above-said issue. It combines the semantic-based content methodologies along with the deep learning techniques to generate the human-readable abstractive summaries. Also, some of the experiments are performed to test and evaluate the potential and generalizability of the framework.

4.2. BiSET: Bi-directional Selective Encoding with Template for Abstractive Summarization (Wang et al., ACL - 2019)

4.2.1 Introduction and Summary

In Abstractive summarization, we preserve the context and the idea by rewriting the sentence in a shorter form. Sequence-to-sequence models have achieved state-of-the-art facility in text summarization. However, they fail to distinguish the article information from the noisy text, and thus these models generate repeated words frequently. Usually, template-based summarization works well, where templates have to be selected manually, but the manual selection is tedious work. The

authors have proposed a novel technique that consists of a bidirectional layer of two gates, which select essential information from templates and articles to support in text summarization. Moreover, a multi-stage process (Fast Rerank method) has been proposed for retrieving templates automatically. The authors have divided their approach into three modules viz, Retrieve, Fast Rerank, and BiSET. Retrieve returns selected templates from the training dataset. Fast Rerank identifies the best templates from the selected ones. Finally. Bi-SET returns the vital information from the input article, which finally gives a compelling summary of the given article. These three modules are a combination of Convolutional neural networks(CNN) and encoder-decoder layers. After the model is created, training was performed with two different loss functions. The Fast Rerank method uses ROUGE-1 score and cross-entropy loss function whereas, Bi-SET uses negative log-likelihood between predicted and actual summary.

4.2.2 Model Architecture

The model architecture consists of the above three described modules:

4.2.2.1 Retrieve

In this module, first, the non-alphabetical characters are removed, followed by searching the corpus with the source to discover (5 to 30) related articles. Their summary is treated as selected templates.

4.2.2.2 Fast Rerank

This method maintains the semantic relationship with the source article from the selected templates. This method consists of an Encoder Block of Convolution, a matrix for similarity, and a layer for pooling.

4.2.2.3 Convolution Encoder Block

The authors have focused on semantic relevance in the convolutional encoder block and have used a novel block that includes a layer for word embedding, convolution network(1D) with a nonlinear function, and skip connection. The word embedding is allowed to pass through the convolution layer to generate the n-gram feature and GLU(gated linear unit) activation function to identify how much information has to be passed. Finally, skip connections are applied to retain original information.

4.2.2.4 Similarity matrix

The performance of the Euclidean distance matrix is quite better than the available similarity functions. This function has been used to calculate the similarity matrix for article representation.

4.2.2.5 Pooling layer

This has been used to remove the unnecessary information in the similarity matrix-like, repeated words considered to be one. After pooling, a neural network is used to find out the score of similarity for the article.

4.2.2.6 BiSET

This helps in assisting article representation and summary generation. It consists of two gates: Template to Article(T2A) and Article to Template (A2T). T2A filters the source article representation. At the same time, A2T controls the amount of final output template, which has to be present in the article representation. The weighted sum of source representation and output representation is considered to be the final article representation. The RNN decoder layer produces a new hidden state for the current step. Attention is computed between the current hidden state and final article representation to find the current context vector. Finally, the resulting context vector and current hidden state are concatenated, which gets input to the softmax layer to generate target word distribution.

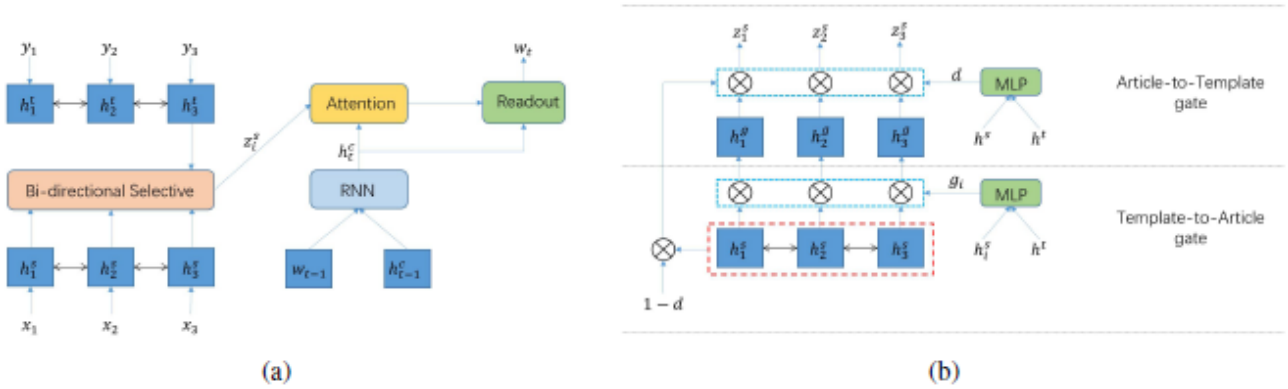

Fig 3: Model architecture: a) BiSET model and b) bi-directional selective layer

4.2.3 Why did the proposed architecture work?

1. As already mentioned, sequence to sequence models does not perform well and produce repeated words quite often. The novel architecture solved this problem with two gates to extract critical information from the templates and articles by omitting repeated words.
2. Template-based summarization is an already existing technique with the problem that templates have to be selected manually. This was also solved by the Fast Rerank module, which automatically selected the best template from the selected templates.

4.2.4 Results and Analysis

1. **Retrieve:** This module was evaluated with three layers of templates viz, Random, which randomly selected the corpus. Retrieve-top, which selects the highest-ranked summary. N-optimal selects the top N results with the high rouge score. The experiments suggest that N-optimal gives better results than the other two.

2. **Fast Rerank:** The ranking quality is measured under different ranges. For each template, ROGUE-2 score is calculated, and it provides enhanced ranking compared to the original Retrieve ranking.

3. **BiSET:** During the experiment, BiSET was tested with two other templates: randomly and the best template selected by Fast Rerank. In both the cases, the model performed well with satisfactory results.

4. **Speed Comparison:** The proposed model performs not only accurately but efficiently also. The Fast Rerank module consumes around 30 minutes. At the same time, the BiSET model consumes around 8 hours. The property of parallelizability for CNN helped in achieving fast performance.

5. **Ablation study:** This was done to study the effectiveness of bi-directional layers and their gates. The model's performance was poor on replacing the selective layer with an article's template. The authors also found that T2A is the most important gate as it can alone control the article's information.

4.2.5 Limitations and further improvement

1. Though the model performs very well, improvement can still be made. In the Fast Rerank architecture, the Similarity matrix can be used to combine other similarity functions. Randomly selected functions might produce good results.
2. In BiSET architecture, in the decoder layer, authors have used RNN to control the flow of information. LSTM or Bi-LSTM could be used to have better control of long and short-term information.
3. Although the authors have evaluated the model on Annotated English Gigaword, evaluation could be done on some other language datasets too. This would have helped in understanding where the model fails and does not perform well.
4. In the bidirectional selective layer, T2A and A2T gate can be combined, and the results in both of them are taking the same input, and a filter gate can be used which output is to be passed in the two layers.

4.2.6 Conclusion

1. The authors have presented a novel technique for abstractive summarization, which is the Bi-SET model.

2. Because of the issues with sequence to sequence models and lack of training corpus, summaries have been generated as templates to help with the source article representations.

3. Quality templates have been achieved by Retrieval and Fast Rerank modules.

4. Evaluation with the English Gigaword dataset shows that the model performs efficiently and accurately and produces high-quality templates. From the results it can be concluded that the novel method performs best as compared to existing baseline models.

5. Ablation study also validates the effectiveness of bi-directional selective layers.

4.3 MAST: Multimodal Abstractive Summarization with Trimodal Hierarchical Attention (Khullar et al., EMNLP - 2020)

4.3.1 Introduction and summary

In the present era, retrieving information from videos is simple with the help of speech-to-text transcripts, but the information obtained is lengthy. In such cases, multimodal text summarization is quite an effective technique. It is a way of compressing the information from the interacting modalities into an output summary. Abstractive multimodal summarization cannot capture the audio effects; hence authors have sorted out this issue with the inclusion of audio modality. Authors have

proposed a novel model with trimodal hierarchical attention (MAST). The model consists of three components: Modality Encoder, Trimodal Hierarchical Attention Layer, and Trimodal Decoder. This model is a sequence to sequence model that combines the important information from text, audio, and video modalities. The encoded information is allowed to pass through these components. During the process, the information is affected by two modalities(Audio-Text and Video-Text) followed by having modality in each pair. Different combinations of modality helps decoder to find the summary.

4.3.2 Model Architecture

There exist multiple modalities which human uses for communication. It is evident that by nature, we humans use natural languages as a prime medium for communication. The major forms of natural language inputs include audio, text, and video. In particular, MAST incorporates all these three modalities into one.

1. Audio: 40-D Kaldi filter banks in the range of 16kHzaudio clips have been extracted. The time window is set to 25ms with a frameshift of 10ms.

2. Video: 2048-D features per 16 frames are taken from the dataset using ResNeXt-101.

3. Text: The video transcripts are used for textual data inputs. The data is converted into lower case and normalized before it is fed into the model.

MAST is a seq-2-seq model that incorporates the features of all three modalities. An encoder-decoder-based method is proposed; namely, Trimodal Hierarchical Attention Layer to cater to the different needs of the dataset forms. The essential components of the architecture are described as under:

4.3.2.1 Modality Encoder

A bidirectional GRU-based encoder is used for textual data, while a bidirectional LSTM is used to handle the audio and video data.

4.3.2.2 Trimodal Hierarchical Attention Layer

This module is responsible for combining all three modalities altogether. For each kth-modality, the attention distribution and context vector are computed individualistically at each decoder step. There are two different strategies for merging the multiple modalities, viz., TrimodalH2 and MAST. In particular, TrimodalH2 is just an extension of the hierarchical attention, while MAST is employed for merging and binding the multiple modalities into a single bundle.
The MAST model incorporates the audio-text and text-video contexts via the second layer of hierarchical attention. However, the third hierarchical attention is used to merge them into a single layer.

4.3.2.3 Trimodal Decoder

This is simply a GRU-based conditional decoder that is used to produce the ultimate vocabulary distribution. Moreover, it is evident that the decoder has cumulative information gathered from all the resources at each stage. The prime focus of the trimodal decoder is on the cumulative modalities rather than the individual ones.

4.3.3 Why did the proposed architecture work?

The process of merging multiple modalities into one single framework comes with its own challenges. However, the proposed method targets them efficiently.

1. Major issue in multimodal architectures is associated with the audio data representations. Since Mel-frequency spectral coefficients (MFSC) perform well with deep architectures compared to Mel-frequency cepstral coefficients (MFCC), usage of filter banks is preferred.

2. By their inherent nature, audio data are complex and require many parameters for handling data efficiently. Also, this is more time-consuming. To address these issues, the authors club the audio features into the bins. This results in comparable audio video parameters per timestep.

4.3.4 Results and Analysis

The Trimodal Hierarchical Attention and TrimodalH2 are trained on How2 dataset using all three modalities. Moreover, Hierarchical Attention models. The experiments are performed by embedding the text using an embedding layer of 256 via a bidirectional GRU encoder. However, the audio and video inputs are encoded using bidirectional LSTM having 128 sized hidden layers. Also, a 128 layered GRU-based decoder is used along with two linear layers to generate the ultimate vocabulary.
The improvement in the generalization is achieved through drop-out layers. The number of dropout layers in the text encoder is kept 2 while one in the decoder, which is quite reasonable. Early stopping is used to avoid overfitting criteria.
The model is evaluated against the ROUGE metrics and Content F1 metrics.

4.3.5 Limitations and further improvement

1. One of the future directions toward expanding the concept lies in merging other modalities and representations such as neural network-based audio feature extraction.

2. Other refinement techniques involve the exploration of a transformer-based end-to-end attention-based framework.

3. One other usage of MAST could be seen in other natural language tasks such as machine translations.

4.3.6 Conclusion

MAST is a seq-2-seq model that combines three different modalities of expression viz, text, audio, and video to produce the abstractive multimodal text summaries. In short,

1. It introduces a novel method to incorporate audio into summarization task

2. It also introduces a deep learning-based multimodal framework to incorporate multiple modalities solely for abstractive text summarization.

5. Comparative Concept Matrix

The concept matrix below summarizes the core ideas from the reviewed papers:

Paper Title	Venue, Year	Model Architecture	Method	Training Parameters and Optimizer	Datasets	Evaluation Metrics
Abstractive text summariza-tion based on deep learning and semantic content generaliza-tion	ACL'19	Bi-direction al LSTM based encoder-de coder with attention	Semantic content generaliza-tion using NER	Word2vec embeddings cosine similarity Adam	Gigaword DUC204	ROUGE-1 ROUGE-2 ROUGE-L
BiSET: Bi-directional Selective Encoding with Template for Abstractive Summariza-tion	ACL'19	Bi-direction al LSTM	Bi-directional selective encoding with template	Adam	Gigaword	ROUGE-1 ROUGE-2 ROUGE-L
MAST: Multimodal Abstractive Summarizati on with Trimodal Hierarchical Attention	EMNLP'20	Multimodal encoder-de coder with attention	Tri-modal hierarchical attention	NLL Loss, Adam	300h version of How2	ROUGE-1 ROUGE-2 ROUGE-L Content F1

PART B: Implementation Details

1. Dataset

For the task of abstractive text summarization, the dataset chosen is News Dataset.

The dataset consists of two CSV files, one CSV consisting of **4515** examples describing the author name, headlines, Links of the article, short text, and complete article. The other CSV file consists of ground truth headlines and their corresponding text **(around 98k)**. From the first CSV file, headlines and text have been extracted.

Thus, after merging the two CSVs, the dataset description is:

	headlines	text
count	102915	102797
unique	99916	102756
top	Ganguly called his shirt-waving celebration a ...	Former Windies' captain Brian Lara named his f...
freq	3	2

Clearly, we have more headlines than texts. These must be dropped as we cannot have a summary without an article.

So, we check for all the null values:

```
1 # Checking for null values
2 data.isnull().sum()
```

```
headlines      0
text         118
dtype: int64
```

Now, after dropping all the records with null values, we have:

```
# Drop the rows with Null
data.dropna(axis=0,how ='any',inplace=True)
data.shape
```

```
(102797, 2)
```

So, In total, we have 102797 data samples to work on, and the final CSV file in pandas dataframe has around 1 lakh headlines for the text. Some samples from the dataset are as follows:

	Unnamed: 0	headlines	text
0	0	Daman & Diu revokes mandatory Rakshabandhan in...	The Administration of Union Territory Daman an...
1	1	Malaika slams user who trolled her for 'divorc...	Malaika Arora slammed an Instagram user who tr...
2	2	'Virgin' now corrected to 'Unmarried' in IGIMS...	The Indira Gandhi Institute of Medical Science...
3	3	Aaj aapne pakad liya: LeT man Dujana before be...	Lashkar-e-Taiba's Kashmir commander Abu Dujana...
4	4	Hotel staff to get training to spot signs of s...	Hotels in Maharashtra will train their staff t...

2. Preprocessing (Specific to the Transformer)

1. We intentionally introduced padding using <go> and <stop> tokens with the separation of the start and end delimiters of a sentence.
2. The tokenizer is used to filter unnecessary information, including punctuation marks. This is also used to convert sentences into tokens and to convert text to lower case.
3. Padding/truncating is done to have a fixed-length output which is subsequently fed into the model.
4. Positional embeddings are used to provide the order to the input words.
5. Masking is used in two forms: padding mask for overlooking extra padding with a length shorter than the assigned length. Look ahead mask is used for overlooking words that come after a given current word.

3. Model Architecture

The model we have used is an Encoder-Decoder-based Transformer with multi-head attention. Unlike BERT, which uses a masked language model, we have used two types of masks in the transformer. The model architecture consists of the following parts:

1. **Scaled dot-product:** This is used for attention weight calculation using query, key, and values.
2. **Feedforward network:** It is a two-layered feed-forward network used after every sub-layer. The activation function used is Relu.
3. **Multihead attention:** In this, the input is split into multiple heads(8). This means to have more than one Queries, key, and values triplets. The value of Queries, key and values can

be obtained just like that of single head attention, and these are concatenated to form a single output.

4. **Transformer unit- Encoder and Decoder blocks:** In Encoder, layer normalization and dropout(p=0.1) are being added for 2 layers. The final output of the encoder is the sum of the individual output of layer normalization and dropouts. The decoder part is more or less the same; the only difference is that normalization and dropouts are being done 3 times. This consists of Multi-head attention.

5. **Encoder and Decoder multilayers:** Encoder and Decoder blocks are integrated into these layers. Here, input embeddings are first obtained and added to the positional embeddings.

6. **Custom learning rate:** This type of learning rate changes adaptively and helps in faster convergence.

7. **Transformer:** The transformer class consists of the input of encoder and decoder units, and it finally returns the decoder output and attention weights.

4. Training

1. **Hyperparameters**

    - Number of layers = 4
    - Number of attention heads = 8
    - Epochs = 22 (20 for comparative study), (50 for bonus question)
    - Optimizer = Adam with custom learning rate
    - d_model = 128
    - dff = 512
    - Buffer size = 31000
    - Batch size = 52
    - Encoder max length = 220
    - Decoder max length = 56
    - encoder_vocab_size, decoder_vocab_size = 85232, 33178.

2. **Selection of hyperparameters**

    Earlier, the hyperparameters were supposed to be selected based on grid search, but due to the large size of the dataset, colab crashed. Thus selection was based on randomly selecting the parameters from their standard range(standard range refers to the range found in research papers and blogs). By doing this 2-3 times, we finally selected the most suitable parameters for our model.

3. **Training**

In training, the model is trained on the train set (50k samples). The predictions are computed from the transformer function; the training loss and validation loss are simultaneously computed. The training for the last five epochs(Epoch is starting from 1 because we are reloading from the previous checkpoint) can be summarized as below:

```
Epoch 1 train_Loss 4.5589
Epoch 1 val_Loss 4.4546
Time taken for 1 epoch: 544.6295955181122 secs

Epoch 2 train_Loss 4.5081
Epoch 2 val_Loss 4.4056
Time taken for 1 epoch: 525.1484453678131 secs

Epoch 3 train_Loss 4.4619
Epoch 3 val_Loss 4.3595
Time taken for 1 epoch: 524.1073451042175 secs

Epoch 4 train_Loss 4.4201
Epoch 4 val_Loss 4.3162
Time taken for 1 epoch: 524.2665183544159 secs

Saving checkpoint for epoch 5 at checkpoints/ckpt-7
Epoch 5 train_Loss 4.3781
Epoch 5 val_Loss 4.2749
Time taken for 1 epoch: 523.396210193634 secs
```

4. **Loss Function**

   Training is being done with sparse categorical cross-entropy loss. This needs a target as the one-hot encoded version of each word and thus resulting in a huge array.

5. Validation and Splits

1. As already mentioned, the dataframe has around **1 lakh samples.** This has been split into train, validation, and test set.
2. First, the 1 lakh samples are split into train and test in **70:30 ratio**.
3. The train set is further split in the final train and validation set in a 70:30 ratio.
4. The final samples in train, validation and test are **50359, 21583, 30833,** respectively.

6. Evaluation Metrics

1. **ROUGE:** ROUGE stands for Recall Oriented Understudy for Gisting Evaluation. There are two types of ROUGE metrics: ROUGE-N and ROUGE-L. ROUGE-N measures the number of matching between the original summary and the predicted summary. N can be unigram or bigram etc. After deciding, N, Recall, Precision, and F1 scores are calculated. In the code part, we have implemented ROUGE-L. This we selected to have a count of maximum length of common subsequence. For this, a rogue function has been implemented, called when calculating, resulting in the score.

2. **BLEU:** Bilingual Evaluation Understudy (BLEU) was developed for translation; it is also used for text summarization. A perfect match gives the 1-score, whereas a perfect mismatch gives 0. It counts the matching n-grams in the predicted summary to n-grams in the original summary. 1-gram, results can be treated as a single token and 2-gram as a pair.

3. Because of the size constraint of the array, only **1000** test and validation samples are taken. Also, both the test set and validation sets are evaluated on both the metric.

4. The scores for validation and test set can be summarized below:

	Blue score	**Average Rouge score (Rouge-L)**		
		Precision	**Recall**	**F-score**
Validation set	**0.722**	**0.685**	**0.658**	**0.668**
Test set	**0.723**	**0.691**	**0.657**	**0.671**

From the above table, it can be concluded that both the validation and test set perform nearly the same. From the bleu and rouge-l score, it can be concluded that the model performs nearly average over the test set. The score is not up to the mark because of a lack of hyperparameter tuning or training on fewer epochs.

7. Examples (from test set)

7.1 Good summaries

Actual text_1: "Salman Khan, Jacqueline Fernandez and Anil Kapoor starrer 'Race 3', which released today, is "high on style and low on substance," wrote Bollywood Hungama. The film "is a visual treat but lacks in every other department," wrote Times Now. NDTV called the film an "unimaginatively scripted heist thriller". It has been rated 2/5 (Bollywood Hungama, Times Now) and 1.5/5 (NDTV)."

Original summary_1: "Salman Khan starrer 'Race 3' hits the theatres".

Predicted summary_1: "salman khan starrer 'race 3' hits the theatres".

ROUGE-L score:

```
Precision is: 0.9347826086956522

Recall is: 0.9148936170212766

F Score is: 0.9247321328014827
```

Actual text_2: "US commercial startup Rocket Lab has launched its Electron rocket into orbit and deployed three satellites for the first time. The Electron can carry up to about 226 kilograms of payload and the startup

offers a mission for about $5 million. Rocket Lab has also signed contracts with NASA and Seattle-based Spaceflight for small-satellite launches."

Original summary_2: "US startup launches Electron rocket into orbit for 1st time"

Predicted summary_2: "spacex launches self driving aircraft that orbit for 1st time"

ROUGE-L score:

Precision is: 0.7868852459016393

Recall is: 0.8135593220338984

F Score is: 0.800000950013892

7.2 Bad summaries

Actual text_1: "Bihar defeated Arunachal Pradesh by an innings and 870 runs after bowling them out for 54 runs in the U-16 Vijay Merchant Trophy on Monday. Bihar had bowled out Arunachal for 83 in the first innings before posting 1,007 runs in their first innings. This comes exactly 53 years after Pakistan Railways registered first-class cricket's biggest ever margin of victory."

Original summary_1: "Bihar U16 team hits 1,007, bowls out Arunachal for 83 and 54"

Predicted summary_1: "windies beat windies in odi squad for 7th consecutive t20i"

ROUGE-L score:

Precision is: 0.603448275862069

Recall is: 0.5833333333333334

F Score is: 0.5932212889974188

Actual text_2: "Singer Adnan Sami has slammed those who claim to be staunch Muslims on Twitter, but are seen stalking the Twitter timelines of female users during the holy month of Ramzan. He further went on to call such people 'tharkee', which roughly translates to lustful. Responding to Adnan's tweet, a user wrote that 'kattar Muslims' should not be using social media."

Original summary_2: "Adnan slams Muslims stalking girls online during Ramzan"

Predicted summary_2: "twitter reacts to Canadian cop who tried to enter the un"

ROUGE-L score:

```
Precision is: 048214285714285715

Recall is: 0.4909090909090909

F Score is: 0.48648743649054976
```

8. Results and Discussion

1. One word is predicted at a time at decoder, and output is obtained by appending it. The complete sequence is fed to the decoder and repeated until stop keywords appear.

2. Good and bad summaries have been shown in the previous section, along with the corresponding rouge score.

9. Comparative Study

We compare our work in two aspects, viz., the theoretical comparison and the implementational comparison.

9.1 Theoretical Comparison

1. A theoretical comparison has been made between the BiSET paper's result and our code implementation. One more article has been compared (with code) in the upcoming sections **(point 9.2)** .

2. The architectures of both models have some similarities, like attention-based models with the same evaluation metric (ROUGE).

3. The authors have used a family of ROUGE metrics, that is, ROUGE-1, ROUGE-2, and ROUGE-L, for evaluating the model, whereas we have used only **ROUGE-L.**

4. The ROUGE-L F1 score for the BiSET model for various combinations lies between **35-40%**, whereas, for our model, it lies between **65-70%**.

9.2 Implementational Comparison

1. We have exploited the concept used in the paper of converting sentences into their respective entities.

2. This has been done with the Spacy NER. 10,000 samples are taken and replaced with their corresponding entities and finally fed into transformer architecture.

3. Size of train, validation and test set are: **4900, 2100, 3000 (70:30)**.

4. We train the model for 20 epochs and the last five epochs can be shown as:

```
Epoch 1 train_Loss 1.3479
Epoch 1 val_Loss 1.1523
Time taken for 1 epoch: 53.53255867958069 secs

Epoch 2 train_Loss 1.1873
Epoch 2 val_Loss 0.9885
Time taken for 1 epoch: 34.03365683555603 secs

Epoch 3 train_Loss 1.0211
Epoch 3 val_Loss 0.8348
Time taken for 1 epoch: 34.13114142417908 secs

Epoch 4 train_Loss 0.8682
Epoch 4 val_Loss 0.6984
Time taken for 1 epoch: 34.083160638809204 secs

Saving checkpoint for epoch 5 at checkpoints/ckpt-8
Epoch 5 train_Loss 0.7337
Epoch 5 val_Loss 0.5835
Time taken for 1 epoch: 34.38087582588196 secs
```

5. **Examples from test set are:**
    1. **Good summaries:**

Actual text_1: "A GPE court has cleared extradition of 50-year-old bookie PERSON, a key accused in the DATE match-fixing scandal involving late GPE captain PERSON, to GPE. The accused had moved to the GPE in DATE and obtained a NORP passport in DATE. This comes after the GPE high court accepted assurances given by GPE regarding PERSON condition."

Original summary_1: "GPE court clears extradition of accused in DATE match-fixing scandal".

Predicted summary_1: "org scam accused asks court for separate cell in gpe".

ROUGE-L score:

```
Precision is :0.8269230769230769

Recall is :0.6323529411764706

F Score is :0.7166676175555589
```

Actual text_2: "PERSON has tweeted a picture of himself from the sets of his son PERSON debut film PERSON Ke Paas in PERSON. PERSON captioned the picture, "WORK_OF_ART!" as he is directing the film. PERSON's debut film is being produced by the PERSON family's production house PERSON, which also launched his father in 'PERSON' in DATE. GPE, Mar CARDINAL (ORG) Actor-filmmaker PERSON, who is working hard to give his son PERSON a perfect launch, is shooting for his debut film in the beautiful mountains of ORG. Titled "PERSON" the movie is a romantic drama."

Original summary_2: "Sunny tweets pic from sets of his son PERSON's debut film"

Predicted summary_2: "org's pic of sets of org surfaces online"

ROUGE-L score:

```
Precision is :0.8

Recall is :0.5614035087719298

F Score is :0.6597947659687569
```

2.Bad summaries:

Actual text_1: "ORG (DRDO) has successfully test-fired PERSON TIME (Long Range Surface-to-Air Missile) from naval warship INS Chennai off the coast of GPE. "The missile (directly) hit a low flying aerial target," ORG Minister PERSON tweeted. The PERSON CARDINAL LRSAM missile defence system has been jointly developed by ORG and ORG."

Original summary_1: "DRDO successfully test-fires PERSON TIME missile"

Predicted summary_1: "gpe 1st to voluntarily contribute at org tax fund"

ROUGE-L score:

```
Precision is :0.40816326530612246

Recall is :0.4166666666666667

F Score is :0.41237208402593867
```

6. The scores for validation and test set can be summarized below:

	Blue score	Average Rouge score (Rouge-L)		
		Precision	Recall	F-score
Validation set	**0.7145**	**0.6162**	**0.5864**	**0.5935**
Test set	**0.7143**	**0.6153**	**0.5833**	**0.5908**

7. From the above table it can be concluded that when the sentences are replaced with entities, the performance degrades. This could be due to variation of entities generated by Spacy NER in the train and test set. Had it been Stanford or any other effective NER, the results would be much much better.

10. Limitations and Future Direction

1. As already mentioned, our model is traditional encoder-decoder transformer-based, and such models are prone to overfitting.
2. Our model did not generalize well because we used fewer **epochs (22),** and hyperparameters are also appropriately selected.
3. Transformer-based models are resource hungry and are computationally inefficient. Hence it becomes necessary to select hyperparameters properly.
4. The Attention used in the model deals with constant length strings. So when the text is split into tokens, the context of the sentences changes.

Colab-file links:

Pre-processing :

https://colab.research.google.com/drive/185ar78NjuvE_vJ5mfeIpPoJmZt6UZl1H?usp=sharing

Main Model:

https://colab.research.google.com/drive/1ndjq-oSW0FHyNaR-bt_TdpV_9x1cfvKV?usp=sharing

Part C: Bonus

Text Summarization on Amazon Food Reviews

1. **499** samples of Amazon food reviews were selected (subset of the dataset).

2. Train test and validation splitting has been done by 5 fold cross validation(**train samples:400 and validation:99**).

3. First the pre-processing is done on the dataset and the cleaned dataset is fed as input to the transformer model. Following is the head of the processed:

	Unnamed: 0	Id	Summary	Text
0	0	1	good quality dog food	i have bought several of the vitality canned d...
1	1	2	not as advertised	product arrived labeled as jumbo salted peanut...
2	2	3	delight says it all	this is a confection that has been around a fe...
3	3	4	cough medicine	if you are looking for the secret ingredient i...
4	4	5	great taffy	great taffy at a great price there was a wid...

4. The training is done on **50 epochs**. Following are the training and validation losses for last 5 epochs:

```
Epoch 46 train_Loss 2.1295
Epoch 46 val_Loss 2.0843
Time taken for 1 epoch: 5.265980958938599 secs

Epoch 47 train_Loss 2.0402
Epoch 47 val_Loss 2.0186
Time taken for 1 epoch: 5.24295711517334 secs

Epoch 48 train_Loss 1.9448
Epoch 48 val_Loss 1.9236
Time taken for 1 epoch: 5.2502148151397705 secs

Epoch 49 train_Loss 1.8556
Epoch 49 val_Loss 1.8291
Time taken for 1 epoch: 5.242142915725708 secs

Saving checkpoint for epoch 50 at checkpoints/ckpt-11
Epoch 50 train_Loss 1.7452
Epoch 50 val_Loss 1.7143
Time taken for 1 epoch: 5.58104681968689 secs
```

5. Randomly **20 samples** are selected from the validation set and the blue score and rouge score is shown below:

```
blue score for every 20 samples 0.18926305296998985

blue score for every 20 samples 0.23926305296998987

blue score for every 20 samples 0.2892630529699899

blue score for every 20 samples 0.3392630529699899

blue score for every 20 samples 0.38926305296998986
```

BLUE score

```
for every 20 samples:
 Average Precision is :0.11036350345560872
AverageRecall is :0.08328947368421052
Average F Score is :0.09203778622194574


for every 20 samples:
 Average Precision is :0.11588167862838916
AverageRecall is :0.08745394736842106
Average F Score is :0.09663967553304302


for every 20 samples:
 Average Precision is :0.11615758738702817
AverageRecall is :0.08766217105263158
Average F Score is :0.0968697699985979


for every 20 samples:
 Average Precision is :0.11617138282496013
AverageRecall is :0.0876725822368421
Average F Score is :0.09688127472187562


for every 20 samples:
 Average Precision is :0.11617207259685672
AverageRecall is :0.08767310279605264
Average F Score is :0.09688184995803953
```

Rouge-L score

Average blue score for 99(validation set) samples is 0.791
Average Rouge-L score for 99(validation set) samples is 0.459

6. Some examples from validation set are:

Actual text_1: ”my year old basenji jack russell mix loves this dog food he s been noticeably healthier and more energetic since i switched him over from the standard dog foods earlier this year despite the higher cost of natural dog foods i find that he eats significantly less of the natural balance dog foods and still stays happy and full on the normal dog foods he d eat up to cups of dog food a day the recommended serving for his size whereas he only eats about cup to cup of the natural balance dog food a day when you take this into account you re actually getting more bang for your buck with the natural dog foods since you don t have to buy as much to last just as long as the normal dog foods and a healthier happier dog to boot add in the fact that you can get free day shipping with amazon prime i m sold”

Original summary_1: “great dog food”.

Predicted summary_1: “great for my dog food”.

ROUGE-L score:

Precision is :0.6666666666666666

Recall is :1.0

F Score is :0.8000009520000029

Actual text_2: “not what i was expecting in terms of the company s reputation for excellent home delivery products ”

Original summary_2: “disappointing”

Predicted summary_2: “the best ever”

ROUGE-L score:

Precision is :0.15384615384615385

Recall is :0.15384615384615385

F Score is :0.15384710384617012

7. **What worked:** It can be said that few examples are there where the F score is above 0.8 and summary produced also acceptable. The correct summary samples are very less and this could possibly because of wrong hyperparameters for this dataset.

8. **What didn’t work:** Most of the predictions are wrong as can be seen from average Rouge-L score. The overall F score and Blue score is very less for every 20 samples but this might be due to less training samples. But the blue score is higher than the F score. That's why we selected two metrics to understand how the model behaves.

References

1. Vaswani, Ashish, et al. "Attention is all you need." arXiv preprint arXiv:1706.03762 (2017).

2. Kouris, Panagiotis, Georgios Alexandridis, and Andreas Stafylopatis. "Abstractive text summarization based on deep learning and semantic content generalization." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

3. Wang, Kai, Xiaojun Quan, and Rui Wang. "BiSET: Bi-directional Selective Encoding with Template for Abstractive Summarization." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

4. Khullar, Aman, and Udit Arora. "MAST: Multimodal Abstractive Summarization with Trimodal Hierarchical Attention." Proceedings of the First International Workshop on Natural Language Processing Beyond Text. 2020.

5. Nallapati, Ramesh, et al. "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond." Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning. 2016.

6. Shi, Tian, et al. "Neural abstractive text summarization with sequence-to-sequence models." ACM Transactions on Data Science 2.1 (2021): 1-37.

7. https://towardsdatascience.com/abstractive-summarization-using-pytorch-f5063e67510 [Accessed : 05-05-2021].

8. Torres-Moreno, Juan-Manuel, ed. Automatic text summarization. John Wiley & Sons, 2014.