JAYANNA, DEEPA, M.S. Machine's conceptual development using FNet. (2022) Directed by Dr. Shan Suthaharan. 53 pp.

Masked language modeling (MLM) is a well-known technique in Natural language processing (NLP) to train a model on randomly masked tokens and use the trained model to predict the masked words. FNet is a recently developed Fourier transform-based transformer that helps solve the MLM problems. It completely eschews the attention computation that has been relatively very famous and replaces it with Fourier transform to perform token mixing. The FNet model reduces the computational complexity of self-attention; however, it compromises with the accuracy scores in contrast to its counterparts. It is well-known that the Fourier transform suffers from the spectral leakage problem caused by the constraint of undersampling of the frequencies from the true infinite frequency domain, as a result; FNet suffers from an aliasing problem that we call text aliasing in our study. The text aliasing, as it resulted from the spectral leakage in Fourier domain, reduces the FNet's ability to predict the correct word for a masked token. In this thesis, we adapted the concept of `learning by exclusion` that is well-established in word learning for children's conceptual development and introduced a new concept of `learning by frequency-exclusion` in the Fourier domain to facilitate word learning for machine's (e.g.FNet's) conceptual development. The idea is to detect the effect of word aliasing through the mutual exclusivity of the narrow-band frequencies, and pass that information to the FNet's encoding mechanism such that the encoder can learn the masked tokens as its vocabulary grows. To validate and evaluate the performance of the proposed approach, we conducted experiments with 15 different sentences as inputs by masking a few words and performing MLM using the pre-trained FNet model parameters. Our finding is that the integration of the proposed `learning by frequency-exclusion` helps FNet to improve its performance.

MACHINE'S CONCEPTUAL DEVELOPMENT USING FNET

by

Deepa Jayanna

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2022

Approved by

Committee Chair

To my husband Nikhil for all the encouragement, support and love.

APPROVAL PAGE

This thesis written by Deepa Jayanna has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair _____
Shan Suthaharan

Committee Members _____
Minjeong Kim

_____
Chunjiang Zhu

_____
Date of Acceptance by Committee

_____
Date of Final Oral Examination

ACKNOWLEDGMENTS

Thanks to my mentor and advisor, Dr. Shan Suthaharan, for encouraging me to pursue this thesis. His guidance and perseverance have made it possible to complete this thesis successfully. The knowledge and humbleness I have gained from him throughout my degree have made me grow as a confident individual.

Thanks to Gunjan Chhablani for the productive discussions on the FNet.

Thanks to my husband for putting up with all my tempers. Thanks to my family for their continuous support. Thanks to my friends who sent me food to keep me alive during my busy schedule.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Transfer learning-based analysis helps improve the interaction between humans and machines. Today human-to-machine interaction is getting more popular because of the latest advancements in technology. Machines can capture the subtle information in the data and respond intelligently, comparable to human cognition. NLP is a branch of Artificial intelligence that aims to teach machines to understand, analyze and comprehend the human language. Appliances like Alexa and Google home have become a vital part of our lives, and they use NLP techniques to learn language representations. A few applications of NLP include text summarization [24, 29] sentiment analysis [21], question answering [19, 23] and language translation [28].

1.1 Machine word learning

Learning from text data for a machine is difficult because of problems like contextual words, homonyms, synonyms, sarcasm, and ambiguity. The same word can have a different meaning according to the context of the sentence. The sentences "I am doing the dishes today" and "That dish was delicious" interpret the word "dish" differently. This problem is categorized as a contextual word problem. Homonyms give rise to a problem for the machine where the spelling of the two words is the same but with different meanings. For instance, "The tree bark" and the "dog barks" are very different in meaning, but the word "bark" has the exact spelling and pronunciation. Text processing in NLP faces tremendous issues due to synonyms. Multiple words can express the same feelings. The terms happy, merry, ecstatic, and joyful convey the same sense, but the intensity of the sentence changes when used in different sentences. For example, when someone says "I am happy" or "I am ecstatic", they don't convey the same meaning. Even though both sentences analyze that the person is happy, the ecstatic word in the second sentence exaggerates the meaning of happy and gives a new dimension to the sentence. Another example could be "The building is excellent" and "The building is magnificent." Here magnificent adds a wow factor that excellent fails to express. Sarcasm is when someone says something but means precisely the opposite. For instance, when the weather is terrible, and someone remarks, "It is

lovely weather today," it actually means the weather is awful but using a positive statement. An ambiguity arises in machine learning when the system cannot interpret the pronouns to what noun they are referring to. For example if we consider the sentence, "The trophy doesn't fit the suitcase because it is too big.", it is easy for us to understand the meaning of the word "it" through a process of word learning and context development. But, for a machine to get the context of "it" is not accessible. There is an ambiguity concerning whether "it" refers to the trophy or the suitcase." These problems make text data learning difficult for a machine learning model [26].

1.2 Masked language modeling

MLM is an NLP modeling technique where the model is trained on a large corpus of data by randomly masking some words. The aim is to predict the masked words by utilizing the unmasked words. The model derives the context of the sentence using the unmasked words to predict the word that best fits into the blank word with the [MASK] token. FNet is a transformer-based architecture that uses MLM to address some of the language problems discussed above. The uniqueness of FNet is that it is attention free, unlike its precursor Bi-directional encoder representation from transformer (BERT) [2]. Self-attention helps evaluate the importance of every word w.r.t to other words in the sequence by assigning a score in order to determine the context of that word. This calculation is computationally expensive.

Hence, FNet replaces the self-attention layer with Fourier Transforms to achieve MLM with limited accuracy. Fourier Transform converts the word representation in the time domain to the frequency domain by performing DFT on the sequence length and the dimension length. A significant drawback of the Fourier Transforms is the spectral leakage problem caused by the under-sampling of the frequencies. As a result, the spectral leakage creates text aliasing problems in the Fourier domain which affects the performance of FNet in selecting the correct token for masked words. We perform frequency-masking for the output generated from Fourier Transform to show the spectral leakage and the text-aliasing problem. This thesis shows that our technique learning by frequency exclusion that the FNet can achieve better results when specific frequencies in the Fourier layer output are excluded. The rest of the thesis is organized as follows: Chapter 2 discusses the literature survey required to understand the related work performed to understand the concept of thesis. Chapter 3 describes the FNet model and the internal working. Chapter 4 explains the methodology incorporated to achieve the results. Chapter 5 delivers the experimentation and results to prove the hypothesis. And at last, chapter 6 concludes the experimentation findings and the scope for future work.

1.3 Terminologies

This thesis explains models like ANN, BERT, LSTM, MLM, NSP, Bi-LSTM, GRUs, CECs, LSTM-CRFs, Transformer, and concepts like self-attention to get acquainted with the previous related work. These concepts are essential because it helps the reader understand the evolution of the model. However, the principal model and the idea that drives this thesis are the FNet model and the Fourier layer. The experimentation is solely based on the Fourier layer to observe and analyze the drawbacks and pave a new thought process to improve the model's performance using the Frequency-exclusion technique. Several new terminologies are introduced and used throughout this thesis, including text-aliasing, learning by exclusion, learning by frequency exclusion, and machine word learning. These terminologies are explained below:

- *Text aliasing*: It is renowned that Fourier Transforms are prone to spectral leakage problems during windowing. This effect causes the word frequencies to shift from their original position; thus, creates an aliasing effect. We call this replacement of word's frequency from its original position to a shifted position in the Fourier layer output as *text aliasing* in the context of word learning.

- *Learning by exclusion*: The concept of learning by exclusion was introduced and experimented on children to learn new vocabulary by excluding whatever they are aware of [8]. In the experimentation, children were shown two images and hear a word. It is up to the child to associate the heard word to any image based on phonolgy and guessing. Let's say the child connects the image 1 with the heard word and labels it. In order to label the other image, they can completely exclude the heard word since it is already associated with the first image 1.

- *Learning by frequency-exclusion*: We borrow the idea of learning by exclusion to learning by frequency-exclusion in FNet. The low frequencies in the Fourier layer output are masked by not letting FNet to see the words in that frequencies and depend on the other unmasked frequencies to infer the words. By doing so, we have successfully seen the FNet predicting the correct masked words as high probability word in some cases or moved them from low priority to higher probability places in the window of 5 predictions.

- *Machine word learning*: The ability of the machine to understand and interpret the words in the sentence concerning all aspects of the language like grammar, context, homonyms, synonyms, ambiguity, sarcasm, etc is termed machine word learning

Chapter 2: Background

This section will discuss the major problems existing in the sequence transduction task of NLP and the new techniques developed to overcome them. We start with Artificial Neural Networks (ANNs) as they are the building blocks of Deep Learning [25]. We familiarize with the ANN-based models, the drawbacks associated with them and the improvements made. Then we discuss about encoder-decoder-based Transformers designed for machine translation using the self-attention and BERT – the first pre-trained model to perform MLM. Finally, we discuss about the FNet model, its advantages, drawbacks as well as literature survey about the Fourier Transforms in deep learning.

2.1 Artificial Neural Networks

Deep learning is a trusted technique for NLP as it provides flexibility in modeling that can achieve state-of-the-art results. The deep learning models help eliminate the scalability issues of big data. Artificial Neural Networks (ANNs) are a class of deep learning algorithms that mimics the human brain-like structure. ANN consists of an input layer, a hidden layer, and an output layer. Every layer has its own set of neurons and is connected to subsequent layers. A simple architecture of an ANN is shown in Figure 2.1. Every node or neuron is associated with an input, weight, bias, and output. Each neuron is assigned a value between 0 and 1, called the activation value. An input text or image is passed through the input layer, giving weights to every layer connection. The input passes through the input layer first and then activates the neurons in the next layer based on the weighted sum of the connections. The weights and biases are initialized to small random numbers. The formula for the weighted sum is [16]:

$$y = \sum_{i=1}^{n} w_i \times x_i + bias \tag{2.1}$$

A weighted sum is calculated to activate a neuron, which can sum up to any value. But for activation, we need values between 0 and 1 because the neurons have activation values between 0 and 1. Hence, the weighted sum is passed through an activation

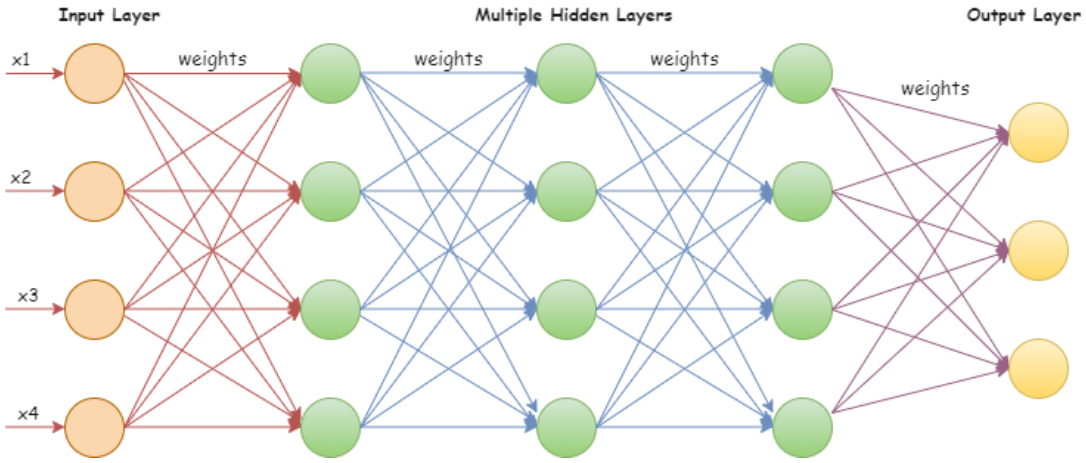


Figure 2.1. Artificial Neural Network. The diagram shows the multiple layers of ANN

function like a sigmoid or logistic curve that yields a value between 0 and 1. Ideally, negative inputs end up close to 0, and positive numbers end up close to 1. The activation neuron is a measure of how positive the weighted sum is. There are cases where the weighted sum is greater than the required value, in which case a bias is added to make the sum 0. The bias is added to the weighted sum before passing through an activation function. The input reaches the output layer by activating multiple neurons in the hidden layer with some output. This one iteration of input data from the input layer to the output layer is called Forward propagation. The results obtained in the forward propagation are not correct yet. The algorithm must evaluate the error and backpropagate to reduce the losses by updating the weights and biases. The weights are updated by taking the difference between old weight and learning rate product and derivative of loss function w.r.t to that weight. The weight updates in the backpropagation is given by [10]:

$$W_{new} = W_{old} - \eta(\frac{\partial L}{\partial W_{old}}), \tag{2.2}$$

where the loss function L is defined by $L = y - \hat{y}$. Weights are updated backward from the output and hidden layers up to the input layers. This iteration of a single forward and backward propagation is called an epoch. A neural network can run many epochs until the losses reach the minimum and the distance between the original and predicted output is reduced. Generally, the learning rate is set to the minimum value to help the model learn optimally by updating the weights minimally. Even though the process can take longer with a lower learning rate, reaching the global minimum in the gradient descent curve becomes easy.

Recurrent Neural Networks (RNNs) are built from recurrent ANNs popularly used for deep learning objectives. They are trained from left to right on sequential data and suitable to preserve the long-term dependencies [20]. But as the length of the text sequence increases, the weight updates in backpropagation get weaker or higher, giving rise to vanishing and exploding gradient problems [12]. As a result, it reduces the training ability of the model and produces inaccurate models. Hence RNNs become unsuitable for preserving long-term dependencies. Long short term memory (LSTM) overcomes the error in backpropagation by maintaining the constant error flow through constant error carrousels (CECs) within special units called cells [13]. Gated recurrent units (GRUs) alleviates the vanishing gradient by introducing a reset gate and an update gate [7]. The gates are vectors that filter the information flow by deciding what information is passed to the output layer. Several versions of LSTM like Bi-directional LSTM (Bi-LSTM), LSTM - Conditional Random Field (LSTM - CRF), and BI-LSTM-CRF have experimented with sequence tagging tasks achieved state-of-the-art results [14]. These versions of LSTMs improved the model's performance in preserving the context of the sequences for a longer time and alleviated the vanishing gradient concerns. Although they circumvent the vanishing gradient problem and help maintain the context of the sentence for a more extended sequence, the definition of the longer sequence is limited to sequences of length 100s or 1000s but not for 10000s and more.

2.2 Transformers - a self-attention model

As the word suggests, a Transformer helps transform or translate a sequence of words from one form to another, like language translation, question answering, and chatbot conversation. The Transformer does not have any Recurrent Neural network to remember the sequential information of words in a sentence. Instead, it has layers of multi-headed attention and feed-forward layers packed inside the stack of encoders and decoders. The attention mechanism notes down keywords or attention words that are important to the semantics of the sentence. This process helps the decoder decipher the meaningful translation from the given input. Other models like GRU and LSTM provide the memory of sequence for a short period, but attention-mechanism retains an infinite window given proper computational resources. Self-attention enables the model to associate each word in the input with other words. Self-attention is the method the Transformer uses to bake the understanding of other relevant words into the one we're currently processing. Remember the text example we considered in the introduction, "The trophy doesn't fit the suitcase because it is too big." Here for a normal human being, it is easy to understand what "it" refers to in the context. But, for a machine to get the context of "it" is not accessible. Self-attention model scans each word (each position in the input sequence) and allows to look at other places in

the input for hints that can help lead to a better encoding for this word. Now the machine will be able to associate "it" with "trophy."

The self-attention calculation is entirely dependent on the Query (Q), Key (K), and Value (V) vectors. Q, K, and V are obtained by multiplying the word embedding with the weight matrix W^Q, W^K, and W^V respectively. These vector dimensions are typically lesser than the dimension of word embeddings. Each word allocates a score to other words in the sentence to determine the importance of that word. The score is calculated by the dot product of Q and K vectors at each location, divided by the square root of query and key vector's dimension (d_k), and then passed through a softmax function. A higher softmax value means that word requires more attention while processing a current word. Multiply the resulting number by a Value vector to keep the essential words for the context and chop off the low-scored words. BERT generates Value vectors for every word while processing a current word, and the value vectors are summed at each position. Instead of using a single attention calculation, multiple attention vectors are calculated by linearly projecting the Query, Key, and Value vectors h times (h-number of heads) using multi-headed attention. In the case of BERT, the h value is 8 and d_k is 64 and d_{model} is 768. Finally, concatenating the attention calculated at each head and multiplying with a weight matrix W^O since the concatenated dimension does not match the original dimension. The attention layer output is then passed to feed-forward layers and the process repeats for n-encoders. The attention matrix calculation is given by the formula [27]:

$$\texttt{Attention}(Q, K, V) = \texttt{softmax}\frac{QK^T}{\sqrt{d_k}} \times V \tag{2.3}$$

Multi-headed attention is calculated by the formula [27]:

$$\texttt{Multi-head}(Q, K, V) = \texttt{Concat}(\texttt{head}_1, ..., \texttt{head}_{12})W^O \tag{2.4}$$

where each head is given by the formula [27]:

$$\texttt{head}_i = \texttt{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.5}$$

2.3 Pre-training and its advantages

Traditionally, a neural network is trained by initializing random weights and then updating the weights in backpropagation. Once the training is complete, the weights are saved to perform analysis on future data. Let's say we have trained the model on text data and want to perform some other classification on new data, but text data. Instead of starting from scratch from weight initialization and going over the process

all over again, what if we can use the old weights to give the model some context about the data beforehand?. Yes, the former model in such a case is the pre-trained model. The trained weights are used as a starting point instead of initializing the weights to random numbers and help the machine to have some knowledge before it is trained for some other related task. Pre-training improves the fine-tuning tasks by reusing the model weights.

2.4 BERT Architecture

BERT is an unsupervised language model pre-trained on a large corpus of unlabelled data trained with an objective of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP)[6]. It is the most reliable and flexible model that performs various NLP tasks like Question Answering, abstract summarization, sentence predictions, etc. The Bi in BERT emphasizes on the bi-directional training capacity. MLM helps the model look to the right and left sides of the masked word to obtain the word's context in the sentence. The context learning happens via the self-attention layer present in the stacked encoders of BERT. The RNNs were only capable of left-to-right sequential training. The main goal of the BERT is to provide bi-directional and parallelization features for training that RNN and Transformer failed to deliver. Also, the pre-training helps the model fine-tune into various NLP applications.

BERT constitutes 12 identical encoders that are stacked and fully connected. Each encoder has a Multliheaded attention layer, a normalization layer, and a feed-forward layer. The first encoder receives the input and passes its output to the successive encoders. At the end of 12 encoders, there is an Output layer for classification tasks. Figure 4 shows a simple diagram of BERT architecture and components of the encoder. To achieve MLM, BERT takes a large corpus of the unlabelled dataset by randomly replacing 15% of the tokens with a unique token called [MASK]. Let's take a simple original sentence - "Machine Learning is an art" and mask the word "art" with [MASK]. The input sequence to the model becomes "Machine Learning is an [MASK]." After pre-training, the model should be able to predict the masked word "art." NSP is a binary classification task and takes two sentences as input separated by a special token [SEP]. It predicts whether the second sentence follows the first sentence by outputting 0 or 1. If the second sentence follows the first sentence logically at any point in the dataset, it outputs 1 and vice versa. [CLS] token marks the beginning of a sentence, and [SEP] determines the end of a sentence. After adding the special tokens, the complete input to the BERT becomes [CLS] Machine Learning is an [MASK]. [SEP] I am very interested in [MASK] about it.[SEP].

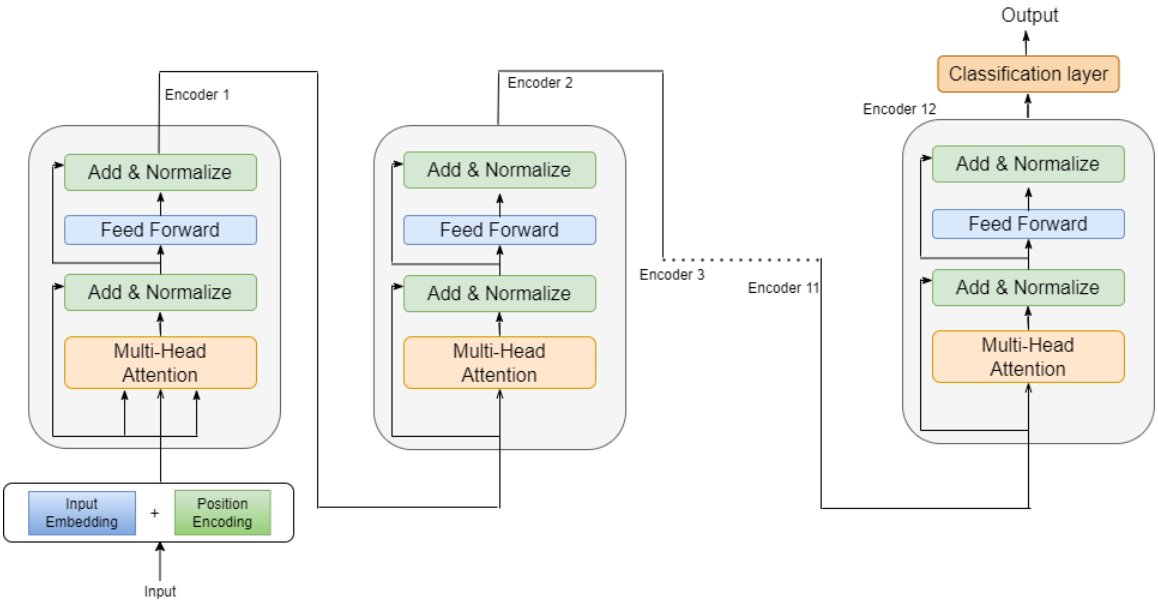


Figure 2.2. BERT. The BERT BASE architecture comprising of 12 encoders

BERT requires a specific format of the input. Every word in the input sequence is converted to tokens and then to a token id that matches the words in the BERT's vocabulary. The token ids are converted into word embeddings and summed with positional embedding to preserve the order of words. The dimension of the word embeddings is [MAX_SEQ_LENGTH, HIDDEN_DIM]. The MAX_SEQ_LENGTH is 512, and HIDDEN_DIM is 768. The first encoder receives an embedding vector of size 512 and a hidden dimension of size 768. The output from the first encoder is given to the subsequent 11 encoders and then projected over a classification layer to get the output. Figure 2.2 shows the components of encoder and 12 encoder connections up to classification layer.

Attention mechanism provides flexibility, resolves one of the significant issues of NLP of context preservation, and allows parallel training. But is it the optimal technique? To find out, let's evaluate the time complexity of the algorithm. Q K^T multiplications over Query and Key vectors of $n \times d$ and $d \times n$ matrices at every position in the sentence result in n^2d complexity. After applying the softmax on the $n \times n$ matrix and multiplying the resulting matrix with the Value vector of $n \times d$ matrix yields n^2d. Since the evaluation of these matrices happens parallelly in multi-headed attention, the end complexity is n^2d. The higher time complexity gives rise to complex and time-consuming evaluations and a genuine question - Do we really need self-attention?

Several novel attempts were made to reduce the Quadratic time and space complexity to linear time by altering the original attention layer. Performers modified the initial attention with Fast Attention via a positive orthogonal random features approach (FAVOR+) [4]. Linear Transformers replaces the actual softmax-based attention with feature map-based dot product attention to achieve linear time complexity[15]. Longformer can process thousands of tokens linearly by drop-in replacement of self-attention combined with local and global windows [1]. It gives attention to each word locally and globally within the specified window, unlike BERT, where the attention is provided locally for a given window length. These methods tweak the attention layer but cannot erase the memory footprints and are still considered memory-bound. FNet replaces the self-attention with a non-parametric Fourier Transform, which eschews any learnable parameters in that layer, allowing a lighter memory footprint.

The use of Fourier transforms in the neural network is not new. In [3], Fourier transforms are used in convolutional neural networks (CNN) to speed up the training process. In [11], the authors further reduce the complexity of convolution in the Fourier domain using the overlap-and-add method. FCNN trains the CNN completely in the Fourier domain by initializing the convolutional kernels in the Fourier domain as in [22]. ANNs are also trained by initializing weights using the Fast Fourier as in [9]. FNet also utilizes the Fourier transformation to achieve a better computational efficiency.

2.5 Drawback of Fourier transforms

Spectral leakage is an inherent problem in the Fourier transforms. FNet performs Fourier transformation on the word embeddings to perform mixing of tokens. As a result, spectral leakage of the word's spectrum passes through subsequent layers and results in wrong predictions sometimes. To show the spectral leakage problem in the Fourier layer, we used the output of sentence 1 test case and generated the magnitude spectrum of the Fourier layer output. The word embeddings are passed through the FFT function, and the mean of 512 points (Maximum sequence length in FNet) over the 768 dimensions is plotted in blue color. The average of windowed 496 points out of 512 is plotted over the 768 dimensions in the red color of Figure 2.3. The original magnitude spectrum of the word embedding is in blue. When the window of 496 values is taken from the actual embedding, spectral leakage can be observed in the red as its magnitude spectrum deviates from the original spectrum. In our case, the words are moving away from their original position where they are supposed to be. And when the masked words are searched in the original frequencies, assuming that it is present, there is no way to extract it because that information has moved to some

other position. In the case of FNet, when the word embeddings that are windowed to the length of 512 words are passed on to the Fourier layer, it generates a spectrum of word frequencies replaced by their original position due to spectral leakage. This leakage creeps into the subsequent layers and iterations, causing the model to predict inaccurate words. We tackle the wrong predictions resulting from the spectral leakage by employing learning by frequency-exclusion.

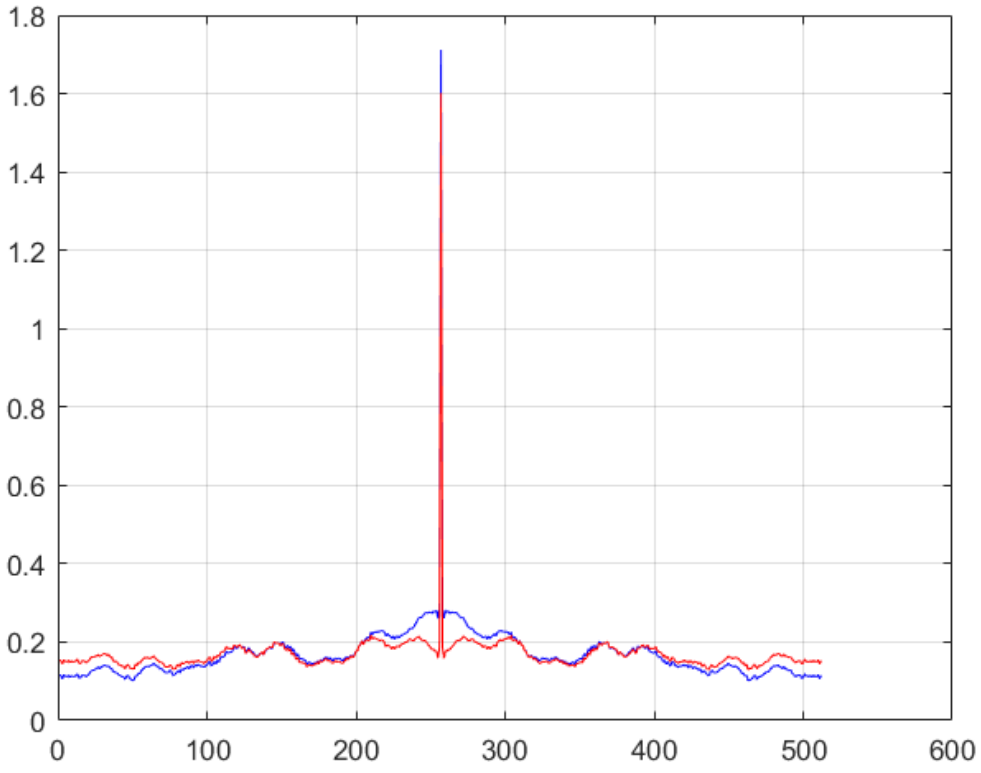


Figure 2.3. Spectral Leakage representation for Sentence 1 experiment. The blue line represents the plot for 512 words over averaged 768 dimensions. The red plot is the plot of 496 points over mean of 768 dimensions. The red line is leaked from it's original spectrum as a result of windowing

Chapter 3: FNet model

FNet is a transformer model that replaces the self-attention layer with a Fourier transform. It addresses the MLM problem by reducing the computations using FFT. FNet performs 80% faster in GPUs and 70% faster on TPUs than its counterpart BERT while performing the same task but with a bit of compromise in the accuracy. Even though self-attention is a powerful tool to provide the language with contextual information using MLM, the computational complexity of self-attention is $\mathcal{O}(n^2)$. The algorithm fails to scale up as the sequence length increases. While the models like Linformer, Longformer have approximated self-attention to speed up the computation, the concept of self-attention persists. FNet eschews the self-attention and expensive dot product cost associated with a non-parametric Fourier Transform layer. Fourier Transform is a mathematical function that converts the time domain x(t) values to the frequency domain $x(\omega)$. The best time complexity available to compute the FFT of a matrix is $\mathcal{O}(n \log n)$ [5]. For any sequence x_n where n ranges between 0 to n-1, the DFT is defined by the formula:[18]

$$\mathtt{X}_k = \sum_{n=0}^{N-1} x_n \times e^{-\frac{2\pi nki}{N}} \tag{3.1}$$

$$= \sum_{n=0}^{N-1} x_n \times [cos(\frac{2\pi nk}{N}) - i \times sin(\frac{2\pi nk}{N})], \tag{3.2}$$

where $i = \sqrt{-1}$ and $0 \leq k \leq N-1$. FNet uses Discrete Fourier Transform (DFT) to perform FFT operations. 2D DFT is performed on the word embeddings - one 1D DFT along the input sequence length (F_{seq}) and one 1D DFT along the hidden dimension (F_h). FNet encoders alternatively operate in frequency and time domain. When FFT is applied in second encoder it converts the output back in to its original "time" domain and the process repeats in subsequent layers [18].

$$y = \Re(F_{seq}(F_h(x))) \tag{3.3}$$

3.1 FNet Architecture

The FNet architecture is same as BERT, pre-trained on a large corpus of text data to perform MLM and NSP. Still, the only difference being self-attention is replaced by Fourier transformation. It has two variants, the "Base" and "Large" models like BERT with 12 and 24 encoders, respectively. In this thesis, we are using the FNet "Base" model. The input text is decomposed into word, position, and type embeddings and summed to get the information as a single embedding vector. The embedding vector is passed to the first encoder, output from the first encoder is fed to the second up to 12 encoders. Each encoder comprises Fourier and Feed Forward layers. The output from the last encoder is passed through a dense layer and then through the output projection to perform the classification task. Figure 3.1 shows a simple FNet architecture with 12 encoders.

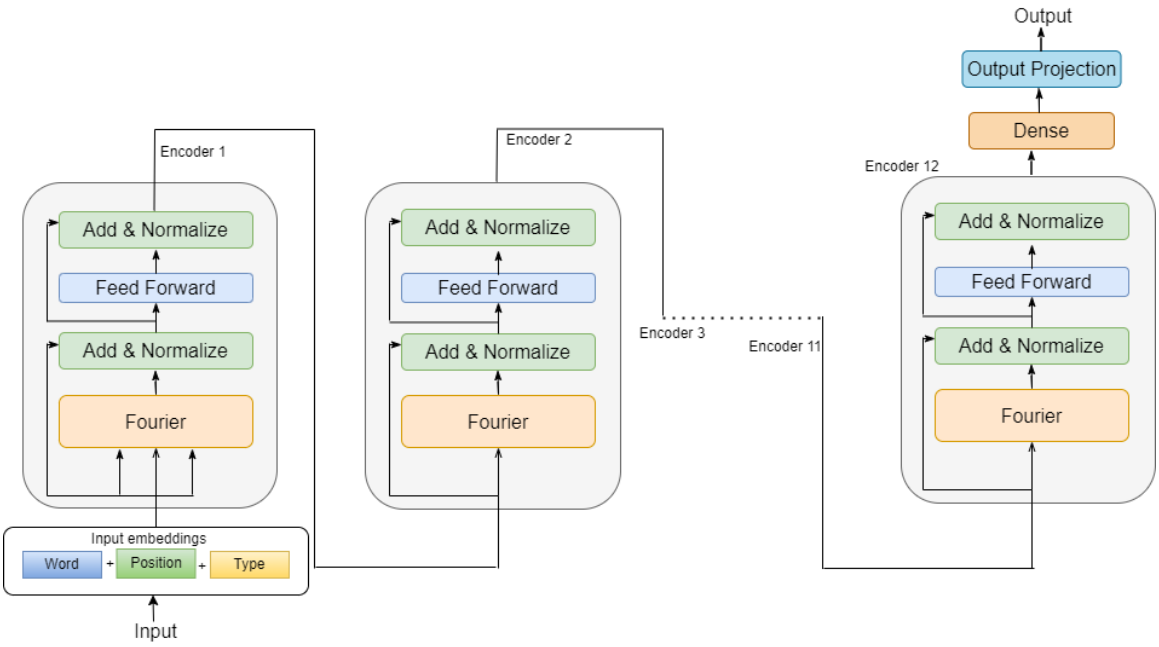


Figure 3.1. FNet. The FNet architecture comprised of 12 encoders

3.2 Dataset and Sentencepiece

The FNet is pre-trained on c4/en dataset [18]. It is a cleaned version of common crawl's web crawl corpus. It has five columns, content-length, content-type, text, timestamp, and URL. The only required column for the training is "text," and can

content-length	content-type	text	timestamp	url
1970	text/plain	Beginners BBQ Class Taking Place in Missoula! Do you want to get better at making delicious BBQ? You will have the opportunity, put this on your calendar now. Thursday, September 22nd join World Class BBQ Champion, Tony Balay from Lonestar Smoke Rangers. He will be teaching a beginner level class for everyone who wants to get better with their culinary skills. He will teach you everything you need to know to compete in a KCBS BBQ competition, including techniques, recipes, timelines, meat selection and trimming, plus smoker and fire information. The cost to be in the class is $35 per person, and for spectators it is free. Included in the cost will be either a t-shirt or apron and you will be tasting samples of each meat that is prepared.....	2019-04-25T12:57:54Z	https://klyq.com/beginners-bbq-class-taking-place-in-missoula/
		Foil plaid lycra and spandex shortall with metallic slinky insets. Attached metallic elastic belt with O-ring. Headband included. Great hip hop or jazz dance costume.		https://awishcometrue.com/Catalogs/Cleara

Figure 3.2. c4 Dataset. The diagram shows the various columns of c4 dataset
Source: `https://www.tensorflow.org/datasets/catalog/c4`

contain varied lengths of words. The Figure 3.2 displays the sample dataset of c4.

The Sentencepiece is a vocabulary model that generates tokenizes and detokenizes the subwords for the given input sentences. It is a language independent model and used to generate vocabulary for specific datasets [17]. Sentencepiece is trained on the c4 dataset to generate all possible tokens for the dataset. It uses Byte Pair Encoding (BPE) and Unigram model to create tokens or vocabulary. The vocabulary has 32000 tokens, each token saved in one line. The line numbers are the token ids for the corresponding tokens. Figure shows a sample arrangement of token ids from 0 to 19 and corresponding tokens in the Sentencepiece vocab file. Similarly there are 32,000 token ids and tokens are available in the model.

The model has few special tokens, [CLS] - marks the beginning of a sentence, [SEP] - a separator that separates two sentences and also exists at the end of a sentence, <pad> - when the default array size cannot be filled because of shorter sequence, in such cases the remaining values are filled with <pad>. The system ignores that token whenever it's read. Table 3.1 represents the special tokens used in FNet model and its usage.

Table 3.1. Sentencepiece model's special tokens

Token	Token id	Usage
[CLS]	4	Used at the indicate the start of a sentence
[SEP]	5	Used to indicate ending of a sentence
[MASK]	6	Used to mask a word
<pad>	3	Used to pad the input arrays if sequence length is less than 512

$< unk >$	$< s >$	$< /s >$	$< pad >$	$[CLS]$	$[SEP]$	$[MASK]$
0	1	2	3	4	5	6
$\_t$	$\_a$	in	he	re	on	$\_the$
7	8	9	10	11	12	13
.	$wrest$	.	.	.	R	
.	15000	.	.	.	31999	

Let's take a simple sentence, "Machine Learning is an art" to understand the tokenization process." Every word in the sentence tries to search for the same word in the Sentencepiece model. If present, the row number of the corresponding word is returned as token id. According to the output in the matrix representation below, we can observe that all the words are present in the vocabulary and returns the token ids. An underscore is always appended at the beginning of a sentence and in-front of every token to indicate the presence of special character space.

$\_Machine$	$\_Learning$	$\_is$	$\_an$	$\_art$	.
8224	7789	65	102	747	16678

If a word is not present, it is split into subwords that can match the words in the vocab and return their token id. This behavior can be seen in below representation, where the word "ambiguous" is not present in the vocab, but splits it into three subwords that are present in the vocab. We can also see in Figure 3.5 that there is no underscore with "ig" and "uous" because "ambiguous" is one word and there is no space present in between.

$\_amb$	ig	$uous$
5271	81	8075

Chapter 4: Methodology

The process followed in the thesis involves two parts. First, we analyze the original FNet model by passing an input sequence to perform MLM. We analyze the Fourier layer output by drawing the magnitude spectrum and shifting the low-frequency content to the middle using fftshift. Secondly, we modify the Fourier layer output by masking the low-frequency region with 0 and predicting the masked words. Figure 4.1 and Figure 4.2 shows a generalized view of the methodolgy.

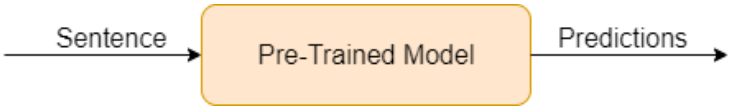


Figure 4.1. Pre-Trained Model. The FNet model pretrained on MLM and NSP

Figure 4.2. Our Modified Encoder. Fourier layer output is masked for various combinati ons of low frequencies and forwarded to subsequent layers

In part one, we will walk through the execution of a simple sentence, "I want to drive. But I am afraid of vehicles on the road". To achieve MLM, two words, "want" and "road," are masked so that the model can predict the original words. The same sentence is passed through the Pre-Trained model in Figure 4.1 and the modified encoder in Figure 4.2.

The first step is to convert the words in the sentence to tokens using the Sentencepiece model. For the machine to understand the start and end of a sentence, the start position is prepended with a special token [CLS], and [SEP] is added after each sentence to mark the end of a sentence. Once the modified input is passed to the Sentencepiece model, it breaks the sentence into tokens. Figure 4.3 shows the generation of tokens

and input ids using the Sentencepiece model. From Figure 4.3, we can see that [CLS] is present in the 4th line, [SEP] in the 5th line, But in the 760th line of the model, and the same holds for all other words. FNet has a constraint for the length of the sequence to be 512. Suppose the sequence length is less than 512. In that case, the remaining values will be padded with <pad> value, i.e., 3. Hence the shape of the input id is [BATCH_SIZE, MAX_SEQUENCE_LENGTH]. The BATCH_SIZE for the experimentation is 1. Figure 4.4 and Figure 4.5 shows the input id and type id matrix for the sentence shown in Figure 4.3

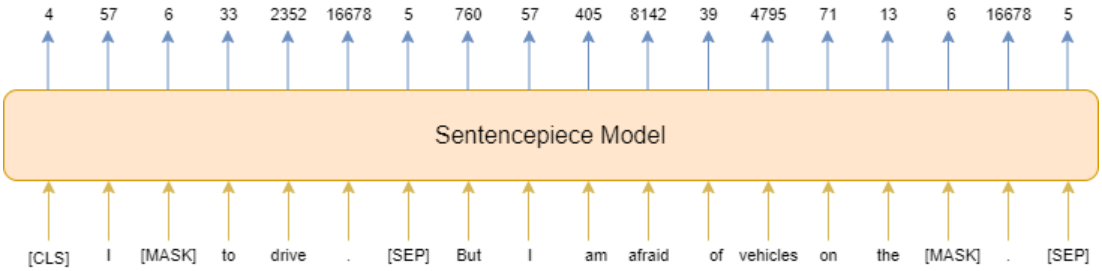


Figure 4.3. Sentencepiece model. The input and output from the Sentencepiece represented with a simple sentence.

The embedding layer takes in two parameters as its input - input id and type id. Type id is the same shape as input id but used to represent the difference between sentence one and sentence 2 for NSP. The length of sentence one is filled with 0, sentence two with one, and the remaining values with 0. The token id and input id are passed into the embedding layer to generate the word, type, and position embeddings. The embedding layer converts the tokens into embeddings of dimension 768. Hence, the output of the embedding layer is of the shape [BATCH_SIZE, MAX_SEQUENCE_LENGTH, EMB_DIMENSION], and in our cases, it is [1,512,768]. Word, type, and position embedding inside the embedding layer are added and represented as a single embedding and returned. Embedding layer can be visualized as in Figure 4.6 and a sample output is shown in Figure 4.7 The Pre-Trained FNet checkpoints are used to test the sentence. The checkpoints are saved for each iteration of the encoder and all the layers inside it. It consists of a dictionary with all the layers of FNet as keys and the associated weights. The keys in the dictionary are embedder, encoders (0 to 11), Feed Forward (0 to 11), and pooler. One can also view the output at each iteration. The embedder does not have iteration because it generates the input ids. For embedding layer ther is only one checkpoint param i.e., embedder. Fourier layer does not have checkpoint parameters because it is non-parametric. Other layers like normalization and feed forward layers takes parameters like encoder_0 and feed_forward_0 respectively for layer 1. Similarly, other layers can be executed individually by giving the respective checkpoints to the layers.

```
input ids [[    4    57     6    33  2352 16678     5   760    57   405  8142    39
   4795    71    13     6 16678     5     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
      3     3     3     3     3     3     3     3     3     3     3     3
```

Figure 4.4. Input ids. Input tokens of shape 512 - empty values are padded with 3

```
type ids [[0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0]]
```

Figure 4.5. Type ids. Type ids of shape 512. Type ids are used to represent sentence 1 length with 0s and sentence 2 with 1s and is helpful in NSP task

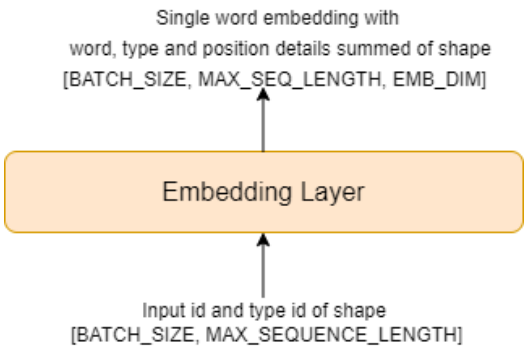


Figure 4.6. Embedding Layer. Embedding layer inputs and outputs shapes

```
embedding shape (1, 512, 768)
embedding layer output [[[-27.18083     -8.900375    -7.768763   ...  -4.4743447   -7.8549266
    -9.025658  ]
  [ -4.037872     1.0234393    0.6311152  ...   1.0761861    0.44298568
    -0.11926322]
  [  0.7403611    0.25609332  -0.23931466 ...  -0.937243    -0.0757433
    -0.05176774]
  ...
  [ -0.56888473  -0.05925487   0.09879221 ...  -0.5823097    0.05607019
     0.07224365]
  [ -8.472882    -0.9528905    0.38689992 ...  -0.7531265    0.14556694
     0.59730315]
  [-12.686011     1.2225777    1.2916728  ...   0.78397226  -0.26144025
     0.17299823]]]
```

Figure 4.7. Embedding Layer. A sample output of embeddings for sentence 1 experimentation

The embedding layer output next goes through the Fourier Transform as in Figure 4.8. 2D DFT is applied across the length of the sequence and the hidden dimension, 1D DFT across the sequence dimension, F_{seq}, and 1D DFT along the hidden dimension, F_h, and returns the real part of Fourier output. A sample fourier output can be seen in Figure 4.9. The equation for 2D Fourier transformations is given as below:

$$y = \Re(F_{seq}(F_h(x))) \tag{4.1}$$

Fourier output is passed to subsequent Feed Forward layers in the next step, as shown in Figure 3.1. Our area of interest is to analyze the importance of frequencies in the Fourier layer. The model predicts the masked words as in Figure 4.10. A sequence of possible words are generated that fit into the blank position by rearranging the

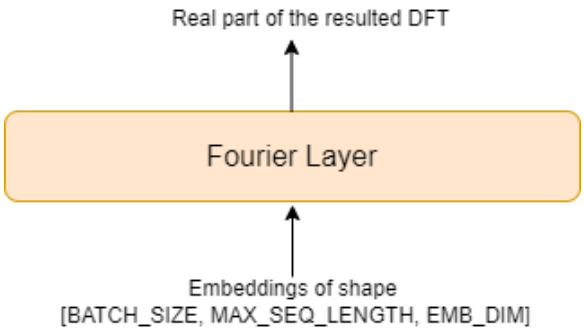


Figure 4.8. Fourier Layer. Produces the real part of resulting DFT

```
Fourier shape (1, 512, 768)
Fourier layer output [[[ 1.3145126e+03  5.5310132e+02  7.1310956e+02 ...  8.9085944e+02
    7.1310950e+02  5.5310120e+02]
  [ 2.0992787e+02  1.9775467e+01  6.6910172e+01 ... -1.7911469e+02
    2.7971078e+02 -5.2122826e+01]
  [ 1.9719070e+02 -3.5260956e+01 -1.0525770e+00 ... -1.2515081e+02
    1.3623657e+02 -4.4801567e+01]
  ...
  [ 2.1526050e+02 -4.0675575e+01  1.2817499e+02 ...  1.0251999e-01
    6.2817154e+00 -8.0670120e+01]
  [ 1.9719073e+02 -4.4801575e+01  1.3623657e+02 ... -2.7920063e+01
   -1.0525780e+00 -3.5260948e+01]
  [ 2.0992786e+02 -5.2122791e+01  2.7971082e+02 ... -3.9743176e+01
    6.6910141e+01  1.9775436e+01]]]
```

Figure 4.9. Fourier layer output. The real part of the output produced by applying DFT on embedding layer output

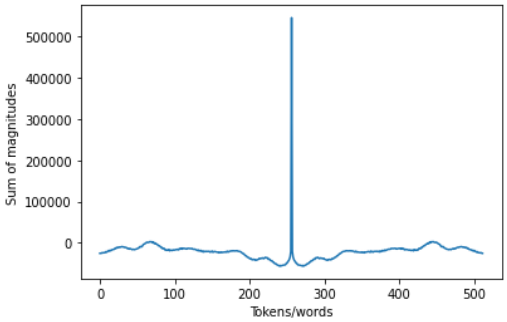
Sentencepiece model. In Figure 4.10, we can see the first five potential values that fit into the masked position, and according to the original sentence, the first value in the array fits perfectly in the blank.

In part two, we will discuss the modifications performed to the Fourier Transform layer for the same sentence considered above. The original flow of the model remains the same, but we tweak the Fourier layer by masking the low frequencies to find the impact in predictions. The dimension of the Fourier Layer output is (1,512,768) and is reshaped to ((1*512), 768) to obtain the magnitude spectrum in 2 dimensional form. We sum the frequencies at each word location and perform a fftshift operation on the fourier layer output to produce the magnitude spectrum of the sentence. The fftshift function rearranges the spectrum by moving the zero-frequency content to the middle of the array. The dimension of the input embedding is (1,512,768) and is reshaped to (BATCH_SIZE*512), 768 to get the magnitude spectrum in 2D form.

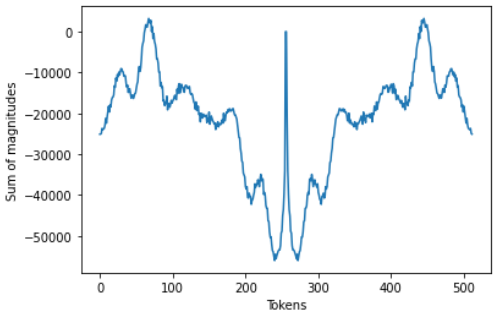
```
Tokens:  [CLS] I[MASK] to drive.[SEP] But I am afraid of vehicles on the[MASK].[SEP]
Top-5 Answers
Mask 0
['want', 'have', 'need', 'like', 'love']
Mask 1
['road', 'roads', 'street', 'highway', 'streets']
```

Figure 4.10. Masked words prediction. The masked words with the [MASK] keyword are predicted

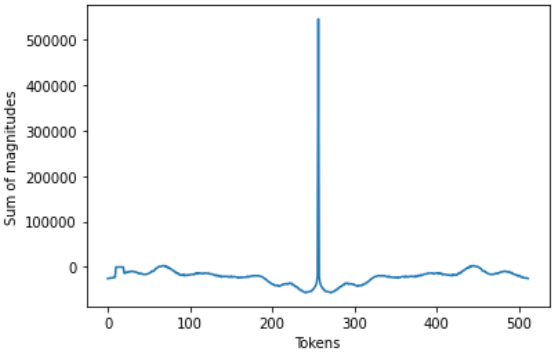
For the experimentation, we will be using BATCH_SIZE = 1. Since the length of the input id is 512, the mid-frequency region is available at 256^{th} and 257^{th} locations. We mask the zero-frequency part and areas around the zero-frequency, both LHS and RHS components, by setting the values to 0. The masking technique is followed for multiple regions in the magnitude spectrum. The fftshift is inversed after making the required frequencies zero by performing inverse FFT shift. The word-frequencies are masked at various locations of low frequency regions and few samples are shown in Figure 4.11 The modified input is converted to its original dimension and passed on to the Feed Forward layers. In Figure 4.11 (a), the original spectrum of sentence 1 experimentation is plotted without any masking of frequencies. The zero frequency component is brought to the middle. In Figure 4.11 (b), we can observe the behaviour of plot when the zero-frequency region is masked. Similarly, the masking region 1 to 9, 230 to 239, 254 to 259, and 259 to 263 plots can be observed from Figure 4.11 (c), (d), (e), and (f) respectively. These are the sample masking techniques discussed, but there are also many other regions that are masked for the experimentation and will be explained in detail in experimentation section.



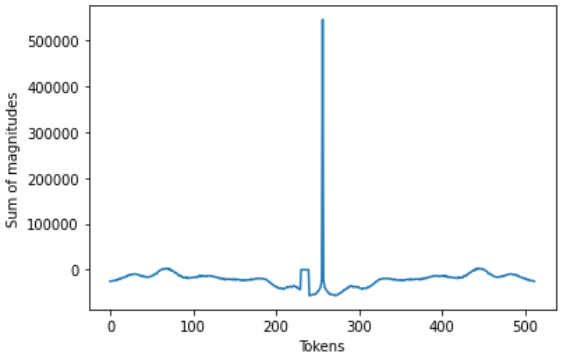
(a) magnitude spectrum: original



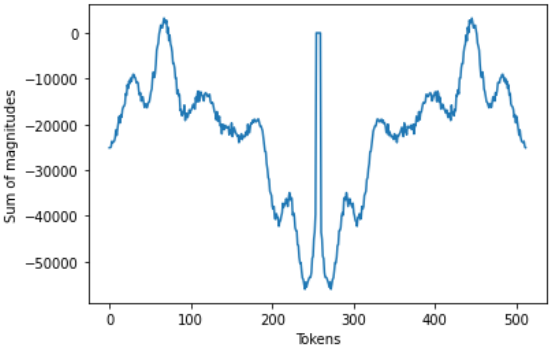
(b) magnitude spectrum : masked position 256 and 257



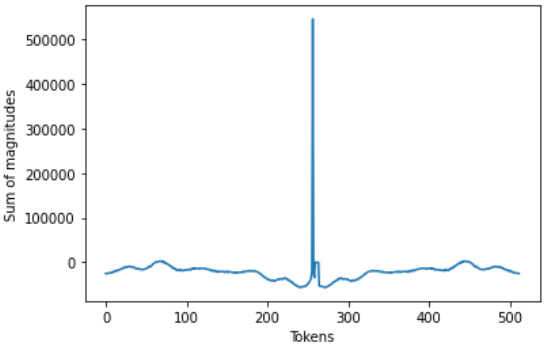
(c) magnitude spectrum : masked position 1 to 9



(d) magnitude spectrum : masked position 230 to 239



(e) magnitude spectrum : masked position 254 to 259



(f) magnitude spectrum : masked position 259 to 263

Figure 4.11. Summed magnitude spectrum of Fourier layer output to visualize the masked positions

Chapter 5: Results

This section will discuss the experimentation conducted and the results obtained. The results are divided into three sections: ace-case scenario, average-case scenario and adverse-case scenario. FNet's performance on the learning by frequency-exclusion are discussed for these three cases. Also, the improvement in the predictions of masked words which are giving better results than FNet original predictions are also discussed. The experiment is conducted on 15 different sentences of various lengths. Some of the words in the sentences are masked using the special token [MASK] and passed through the original FNet encoder to get the masked words. The same sentences are again experimented with by masking various low-frequency regions. The low-frequency region is achieved by performing fftshift on the Fourier output of the word embeddings of size [512, 768].

5.1 Ace-case scenario

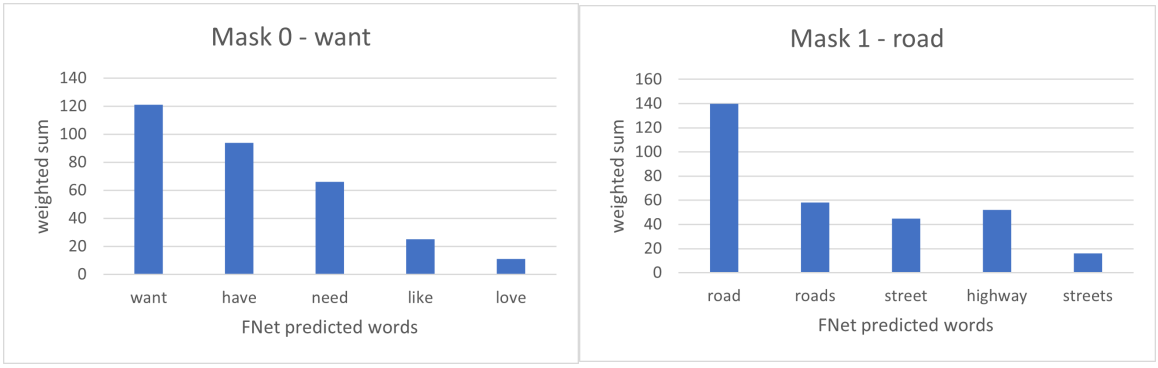
Sentences 1, 4, and 7 performed equivalent to FNet in terms of predictions in multiple frequency-exclusion scenarios. A very simple sentence is considered for the first experimentation and the sentence was able to predict the correct words for masked tokens in original encoder as well as the refined Fourier Transform. The sentence is "I [MASK] to drive. [SEP]But I am of afraid of vehicles in the [MASK]." The masked words being mask0 - want and mask1 - road. The FNet original encoder predicts the desired words and our model also predicts the correct words. FNet predicts the range of words that best fits the masked word. Ideally, it re-organizes the sentencepiece model with the highest probable word to the least probable word. For the experimentation we only consider top five predictions.

In Figure 5.1, the ignored or windowed frequencies are made to 0 so that they are not considered during prediction. It should be noted that since the experimentation was performed in python, the selection [256:258] means the frequency band 256 and 257 were masked. It can be observed that the "want" is predicted as the first choice in 19 cases and road is predicted with high probability in 25 scenarios. With some

masked window like [250:254], [254:356], the expected words move farther from the expected position. It can also be noted that in frequency-exclusion window [257:259], [257:261], [258:260], [254:260], [259:265], [259:270] and [280:310] the required word does not appear in the prediction window signifying the importance of the frequencies in the prediction task. We evaluated these results by assigning weights to the FNet predictions. 1st was assigned a weight of 5, 2nd position with 4, 3rd position with 3, 4nd position with 2 and 5th position with 1. At every masked window, we count the occurrence of all the original predictions of FNet at each position and multiply with assigned weights. Figure 5.2 shows the histogram for the weighted results. It is evident that for mask 0, "want" has the maximum occurrence and for mask 1, "road" has the maximum occurrence making our model behaving equivalent to FNet's prediction but using lesser frequencies.

			Mask 0				Mask 1			
	Word1	Word2	Word3	Word4	Word5	Word1	Word2	Word3	Word4	Word5
Original	want	have	need	like	love	road	roads	street	highway	streets
[256:258]	want	like	have	desire	love	left	road	other	go	right
[254:260]	in	by	x	of	x	service	USA	condition	book	day
[250:256]	vy	am	have	need	want	road	highway	car	roads	street
[250:255]	have	below	get	remember	do	road	ground	car	future	highway
[250:254]	am	want	need	have	mean	road	highway	street	roads	track
[250:253]	want	need	have	like	am	road	street	roads	highway	track
[250:252]	want	have	like	need	hate	road	roads	street	highway	streets
[250:251]	want	have	need	like	love	road	roads	highway	street	streets
[246:248]	want	need	like	have	love	road	highway	roads	street	streets
[244:250]	want	have	need	like	am	road	car	left	ground	right
[243:250]	want	need	have	am	wanted	road	car	world	right	other
[242:250]	want	need	am	have	hope	car	road	computer	world	past
[230:240]	want	have	drive	need	like	drive	like	roads	car	driving
[220:230]	want	need	have	am	plan	road	roads	car	street	highway
[220:240]	hope	want	drive	have	wish	road	car	other	vehicle	street
[210:240]	hope	try	want	learn	wish	road	car	vehicle	roads	highway
[240:250]	want	need	have	am	love	be	one	that	or	to
[255:259]	in	by	x	of	x	service	market	system	year	middle
[255:257]	want	like	have	love	need	go	road	left	rise	right
254:256	have	need	refuse	want	not	road	roads	streets	street	highway
[250:252] [256:258]	need	love	have	want	aid	inside	opposite	course	same	public
[250:251] [256:258]	have	want	need	learn	not	day	top	year	can	time
[257:259]	in	by	x	of	x	road	roads	streets	street	highway
[257:261]	in	by	x	of	x	road	street	streets	roads	highway
[258:260]	in	by	x	of	x	road	roads	street	highway	streets
[259:261]	want	have	need	like	love	road	roads	streets	highway	street
[259:262]	want	am	have	need	like	road	highway	roads	street	ground
[259:263]	am	got	want	have	drive	road	highway	street	roads	ground
[259:264]	want	am	have	drive	vy	road	highway	street	ground	roads
[259:265]	su	ng	ago	have	be	road	street	highway	map	vehicle
[259:270]	am	have	want	do	am	road	website	site	car	head
[270:280]	want	need	have	am	like	by	:	and	of	in
[280:290]	want	need	have	am	.	road	car	highway	roads	street
[280:300]	want	have	plan	got	wanted	road	car	vehicle	roads	highway
[280:310]	posted	go	do	how	do	car	vehicle	road	other	automobile
[330:340]	want	need	have	wanted	used	road	highway	roads	street	like

Figure 5.1. Masked words predictions for original encoder in FNet and our modified encoder for various frequency masks.



(a) mask 0 words

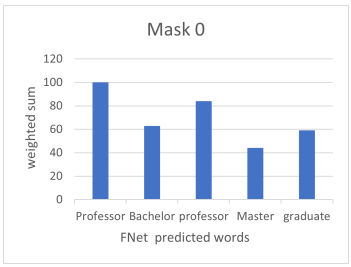
(b) mask 1 words

Figure 5.2. Histogram for Test 1 sentence - mask 0 and mask 1

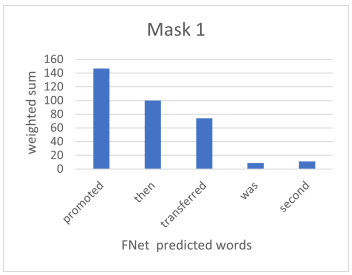
Sentence 7 is a bigger sentence than sentence 1 and the count of masked words are 9. A similar experiments were conducted as sentence 1 and the results are shown from Figure 5.3, Figure 5.5 to Figure 5.7. The weighted sum v/s FNet predicted words are also represented for the corresponding masks in Figure 5.5 to Figure 5.7. The first row mentions the original words that are masked. The second row contains the original FNet predictions without any frequency mask. FNet predicts the correct words as the first option for all masks except mask 7. For mask 7, the correct word "prestigious" is predicted as the 5^{th} priority. But it can be seen that all other words predicted from 1^{st} to 4^{th} position are synonyms of the word "prestigious". FNet fails to predict the correct word as first option due to the presence of synonyms and inability to extract the context from the unmasked words. From 5.7 it can be noted that the word "prestigious" is moved from the last place to the first place in some of the frequency-masks proving that the model can learn or predict the accurate words by frequency-exclusion. From the histograms, it is evident that frequency-exclusion technique performs equivalent to original FNet. Frequency-exclusion technique helps the model to look into available frequencies and picking up the correct word at first position as in case of mask 7. Sentence 4 is experimented by masking 8 words and 6 out of 8 words are predicted as the first choice both by original FNet and by frequency-exclusion technique. The results can be viewed in Figure 5.9.

	Mask0	Professor				Mask1	promoted				Mask2	Director			
none	Professor	Bachelor	professor	Master	graduate	promoted	then	transferre	was	second	Director	Dean	Professor	Chair	Head
256-258	Departme	Division	Master	mother	family	transferre	then	offered	moved	returned	part	Dean	Director	Head	Chair
254-260	the	simple	proven	kind	unique	the	The	another	a	third	part	one	use	half	ative
254-257	collection	piece	day	family	mother	returned	then	transferre	moved	offered	part	Dean	Head	Director	one
254-256	Professor	professor	Bachelor	Master	Doctor	then	promoted	transferre	later	subseque	Professor	Director	Dean	Head	Chair
253-256	Professor	professor	University	Bachelor	Master	then	promoted	transferre	moved	subseque	Professor	Director	Head	Dean	Chair
252-256	Bachelor	Master	University	Professor	professor	then	promoted	transferre	subseque	later	Professor	Director	Head	Chair	Dean
251-256	Bachelor	Master	University	Professor	Masters	then	promoted	transferre	subseque	later	author	Professor	Director	part	Head
250-256	Bachelor	Master	Masters	University	Professor	then	promoted	transferre	later	continued	Professor	author	Head	professor	Director
249-256	Bachelor	Master	Masters	University	Professor	promoted	then	later	transferre	continued	Professor	Head	author	professor	Director
248-256	Bachelor	Master	Masters	Professor	University	promoted	then	consultan	Secretary	later	Professor	Head	Director	head	part
245-248	Professor	PhD	professor	University	S	promoted	then	second	Assistant	Second	Director	Dean	Chair	Professor	Head
245-247	Professor	professor	graduate	Bachelor	Master	promoted	then	second	admitted	Second	Director	Dean	Chair	Professor	Head
244-247	professor	graduate	Professor	student	member	promoted	then	Assistant	second	Post	Director	Dean	Chair	Professor	Head
243-247	professor	graduate	Professor	member	student	promoted	then	admitted	transferre	Assistant	Director	Dean	Chair	Professor	Head
242-247	professor	graduate	Professor	member	student	promoted	then	admitted	transferre	went	Director	Dean	Chair	Professor	Head
230-240	graduate	professor	member	Professor	student	promoted	then	subseque	Assistant	appointed	Professor	Dean	Director	Chair	Head
1:10	Professor	Bachelor	professor	Master	graduate	promoted	then	was	Dean	went	Professor	Dean	Director	Chair	Head
10:20	Professor	professor	Bachelor	Master	graduate	promoted	then	transferre	went	admitted	Director	Professor	Dean	Head	Chair
20:30	Professor	graduate	professor	Bachelor	Master	promoted	then	went	transferre	was	Director	Professor	Dean	Head	Chair
257-259	Professor	professor	Bachelor	Master	graduate	then	promoted	transferre	later	subseque	Director	Dean	Head	Professor	Chair
257-260	professor	Professor	graduate	University	Bachelor	then	promoted	transferre	appointed	subseque	Director	Dean	Head	Chair	Professor
257-261	University	Master	graduate	Bachelor	professor	then	promoted	subseque	assistant	was	Chair	Head	Director	Dean	part
262-268	by	of	know	page	visit	promoted	then	transferre	went	elected	Director	Chair	Dean	Head	Professor
263-268	by	of	know	page	visit	promoted	then	transferre	went	elected	Director	Chair	Dean	Head	Professor
264-268	by	of	know	page	visit	promoted	then	transferre	went	elected	Director	Chair	Dean	Head	Professor
265-268	by	of	know	page	visit	promoted	then	transferre	went	second	Director	Chair	Dean	Professor	Head
266-267	Professor	professor	graduate	Bachelor	Master	promoted	then	transferre	second	was	Director	Dean	Professor	Chair	Head
267-268	Professor	professor	Bachelor	graduate	Master	promoted	then	transferre	went	elected	Director	Dean	Professor	Chair	Head
267-270	graduate	Professor	professor	student	Graduate	promoted	then	transferre	admitted	went	Dean	Director	Chair	Head	Professor
267-271	graduate	professor	Professor	student	Graduate	promoted	then	transferre	admitted	went	Dean	Director	Chair	Head	Professor
272-274	Professor	professor	Bachelor	graduate	Master	promoted	then	transferre	elected	accepted	Director	Dean	Professor	Head	Chair
272-275	Professor	professor	graduate	Bachelor	Master	promoted	then	transferre	elected	accepted	Dean	Professor	Director	Head	Chair
280-285	Professor	graduate	professor	Bachelor	Master	promoted	then	transferre	accepted	elected	Professor	Dean	Director	Chair	Head
500-511	Professor	Bachelor	professor	graduate	Master	promoted	then	was	went	transferre	Dean	Director	Head	Chair	Professor

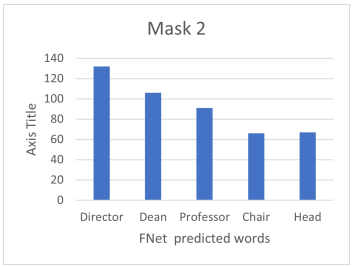
Figure 5.3. Ace case scenario. The sentence 7 was able to predict 8/9 masks. Mask 0, Mask 1 and Mask 2 results shown



(a) mask 0 words



(b) mask 1 words

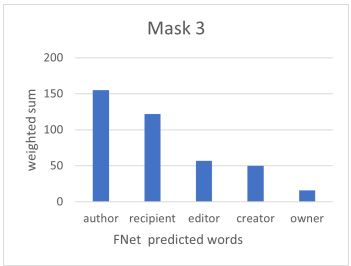


(c) mask 2 words

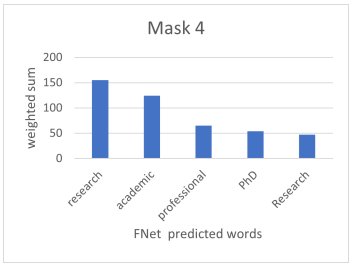
Figure 5.4. Histogram for Test 7 sentence - mask 0, mask 1 and mask2

Mask3	author				Mask4	research				Mask5	machine			
author	recipient	editor	creator	owner	research	academic	professior	PhD	Research	machine	computer	data	statistical	student
use	source	manufact	example	result	personal	best	TRUE	greatest	future	data	machine	computer	,	new
S	s	s	end	change	and	low	,	or	high	the	new	and	good	of
use	result	choice	advantage	source	personal	future	TRUE	full	greatest	data	web	collaborat	grid	visual
author	recipient	architect	champion	creator	research	academic	career	Research	professior	machine	computer	data	statistical	collaborat
author	recipient	master	architect	creator	research	academic	Research	PhD	career	machine	computer	data	statistical	collaborat
author	recipient	master	winner	creator	research	academic	Research	career	professior	computer	machine	data	collaborat	digital
author	recipient	winner	master	creator	research	academic	Research	main	PhD	computer	collaborative	student	machine	digital
author	recipient	winner	holder	owner	research	academic	professior	Research	career	computer	student	collaborat	academic	electronic
author	recipient	winner	champion	owner	research	academic	professior	Research	PhD	student	collaborative	computer	academic	university
author	recipient	winner	owner	editor	research	academic	Research	professior	PhD	student	collaborative	computer	academic	university
author	recipient	creator	editor	architect	research	academic	professior	PhD	Research	machine	computer	data	statistical	process
author	recipient	editor	creator	owner	research	academic	professior	PhD	Research	machine	computer	data	statistical	collaborat
author	recipient	editor	creator	owner	research	academic	professior	PhD	Research	machine	computer	data	statistical	collaborat
author	recipient	editor	creator	owner	research	academic	professior	PhD	Research	machine	computer	data	statistical	collaborat
author	recipient	creator	editor	architect	research	academic	PhD	Research	professior	machine	computer	data	statistical	collaborat
author	recipient	master	creator	leader	research	academic	Research	best	professior	machine	computer	data	statistical	student
author	recipient	editor	creator	owner	research	academic	PhD	professior	Research	machine	computer	data	statistical	software
author	recipient	editor	owner	creator	research	academic	professior	PhD	Research	machine	computer	student	statistical	collaborat
author	recipient	editor	creator	architect	research	academic	professior	PhD	career	machine	computer	data	statistical	language
author	recipient	creator	architect	editor	research	academic	Research	PhD	career	machine	computer	data	statistical	software
author	recipient	master	architect	creator	research	academic	Research	PhD	career	machine	computer	data	statistical	student
author	recipient	master	creator	architect	research	academic	Research	main	career	computer	machine	digital	data	learning
author	recipient	editor	architect	winner	research	academic	professior	PhD	Research	machine	computer	student	academic	collaborat
author	recipient	architect	editor	creator	research	academic	professior	PhD	Research	machine	computer	data	statistical	collaborat
author	recipient	creator	architect	editor	research	academic	professior	PhD	teaching	machine	computer	data	statistical	collaborat
author	recipient	creator	architect	editor	research	academic	professior	PhD	Research	machine	computer	data	statistical	collaborat
author	recipient	editor	creator	owner	research	academic	professior	PhD	career	machine	computer	data	statistical	collaborat
author	recipient	editor	creator	owner	research	academic	professior	PhD	Research	machine	computer	data	statistical	process
author	editor	recipient	creator	owner	research	academic	PhD	professior	Research	machine	computer	data	statistical	student
author	editor	recipient	creator	architect	research	academic	PhD	professior	Research	machine	computer	data	statistical	student
author	recipient	editor	creator	architect	research	academic	professior	PhD	Research	machine	computer	data	process	statistical
author	recipient	editor	architect	creator	research	academic	professior	PhD	Research	machine	computer	data	student	process
author	recipient	editor	creator	owner	research	academic	professior	PhD	Research	machine	computer	data	statistical	software
author	recipient	editor	owner	creator	research	academic	PhD	professior	teaching	machine	computer	data	statistical	Machine

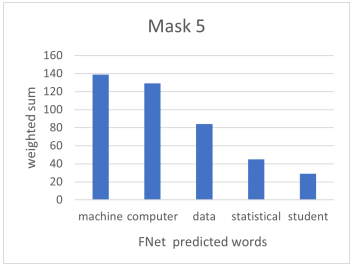
Figure 5.5. Ace case scenario. The sentence 7 was able to predict 8/9 masks. Mask 3, Mask 4 and Mask 5 results shown



(a) mask 3 words



(b) mask 4 words

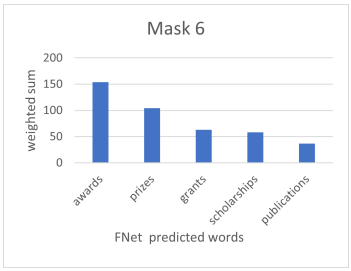


(c) mask 5 words

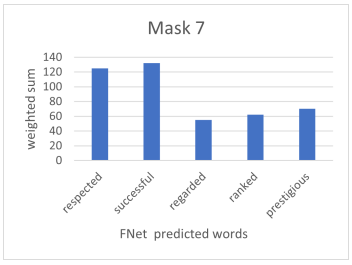
Figure 5.6. Histogram for Test 7 sentence - mask 3, mask 4 and mask 5

Mask6	awards				Mask7	prestigious				Mask8	research			
awards	prizes	grants	scholarshi	publicatio	respected	successful	regarded	ranked	prestigiou	research	work	projects	studies	efforts
others	scholarshi	awards	prizes	offers	popular	successful	reputable	desirable	develope	work	research	projects	education	works
ing	services	users	work	post	and	for	the	a	from	technique	methods	site	technolog	,
offers	others	books	more	prizes	popular	selected	develope	successful	reputable	solutions	content	research	work	results
awards	scholarshi	prizes	grants	positions	successful	respected	prestigiou	ranked	popular	research	work	projects	efforts	learning
awards	prizes	scholarshi	papers	grants	successful	prestigiou	respected	popular	ranked	research	work	projects	learning	tasks
awards	scholarshi	prizes	positions	grants	successful	popular	prestigiou	competiti	respected	research	work	data	tasks	projects
awards	prizes	scholarshi	positions	grants	prestigiou	successful	respected	popular	competiti	research	work	projects	tasks	data
awards	publicatio	prizes	papers	positions	prestigiou	successful	respected	popular	ranked	research	work	projects	studies	efforts
awards	publicatio	papers	articles	prizes	prestigiou	successful	respected	popular	ranked	research	work	projects	studies	training
publicatio	papers	awards	articles	prizes	prestigiou	respected	successful	popular	ranked	research	work	studies	projects	learning
awards	prizes	grants	scholarshi	papers	successful	respected	ranked	regarded	prestigiou	research	work	projects	studies	efforts
awards	prizes	grants	scholarshi	publicatio	respected	successful	ranked	regarded	prestigiou	research	work	projects	studies	efforts
awards	prizes	grants	scholarshi	publicatio	respected	successful	ranked	regarded	prestigiou	research	work	projects	studies	efforts
awards	grants	prizes	scholarshi	publicatio	successful	respected	regarded	ranked	prestigiou	research	work	projects	studies	efforts
awards	prizes	grants	scholarshi	publicatio	successful	respected	ranked	regarded	prestigiou	research	work	projects	studies	learning
awards	prizes	grants	publicatio	scholarshi	successful	respected	ranked	prestigiou	regarded	research	work	studies	projects	learning
awards	prizes	papers	publicatio	grants	respected	successful	prestigiou	ranked	regarded	research	work	projects	learning	studies
PhD	that	in	scientist	to	successful	respected	regarded	prestigiou	ranked	research	work	learning	projects	studies
awards	publicatio	papers	conferenc	prizes	respected	successful	prestigiou	regarded	ranked	research	work	projects	studies	learning
awards	scholarshi	prizes	grants	positions	successful	respected	prestigiou	ranked	regarded	research	work	projects	learning	studies
awards	scholarshi	prizes	grants	publicatio	successful	prestigiou	respected	competiti	popular	research	work	projects	studies	learning
awards	positions	scholarshi	publicatio	grants	successful	prestigiou	competiti	popular	diverse	research	data	work	solutions	algorithms
awards	prizes	grants	scholarshi	papers	respected	regarded	successful	ranked	prestigiou	research	work	projects	studies	experiments
awards	prizes	grants	papers	scholarshi	respected	regarded	ranked	successful	prestigiou	research	work	projects	studies	experiments
awards	prizes	grants	scholarshi	papers	regarded	respected	successful	ranked	prestigiou	research	work	projects	studies	experiments
awards	prizes	grants	scholarshi	papers	respected	regarded	ranked	successful	prestigiou	research	work	projects	studies	efforts
awards	prizes	grants	scholarshi	publicatio	respected	successful	regarded	ranked	prestigiou	research	work	projects	studies	learning
awards	prizes	grants	scholarshi	publicatio	respected	ranked	regarded	successful	prestigiou	research	work	projects	studies	efforts
awards	prizes	grants	publicatio	scholarshi	respected	ranked	successful	regarded	prestigiou	research	work	projects	studies	efforts
awards	prizes	grants	scholarshi	publicatio	respected	ranked	successful	regarded	prestigiou	research	work	projects	studies	efforts
awards	prizes	grants	scholarshi	publicatio	respected	successful	ranked	regarded	prestigiou	research	work	projects	learning	efforts
awards	prizes	grants	scholarshi	papers	respected	successful	ranked	regarded	prestigiou	research	work	projects	learning	efforts
awards	prizes	papers	publicatio	grants	successful	respected	regarded	ranked	prestigiou	research	work	projects	studies	efforts
awards	prizes	papers	grants	publicatio	successful	respected	prestigiou	regarded	ranked	research	work	studies	projects	experiments

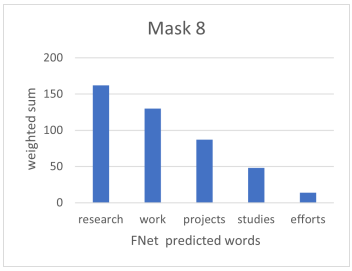
Figure 5.7. Ace case scenario. The sentence 7 was able to predict 8/9 masks. Mask 6, Mask 7 and Mask 8 results shown



(a) mask 6 words

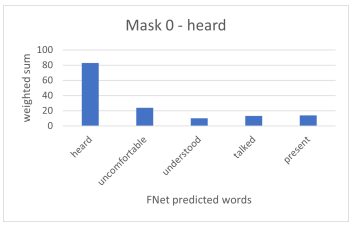


(b) mask 7 words

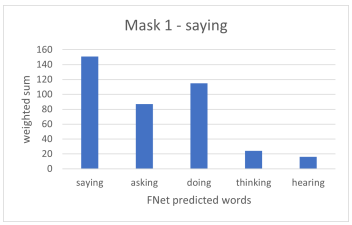


(c) mask 8 words

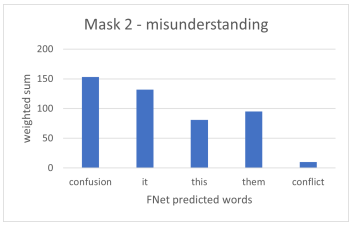
Figure 5.8. Histogram for Test 7 sentence - mask 6, mask 7 and mask 8



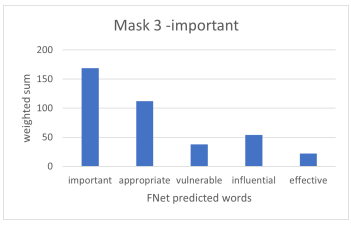
(a) mask 0 words



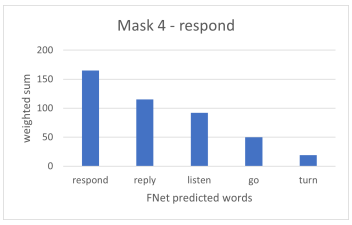
(b) mask 1 words



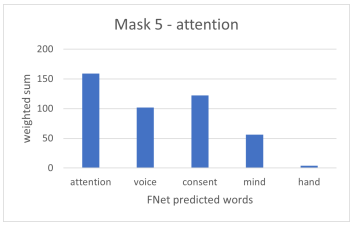
(c) mask 2 words



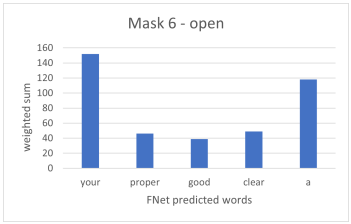
(d) mask 3 words



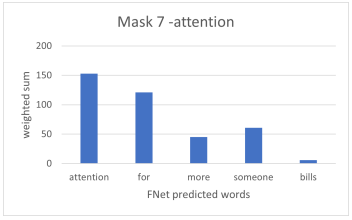
(e) mask 4 words



(f) mask 5 words



(g) mask 6 words

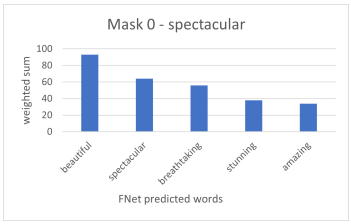


(h) mask 7 words

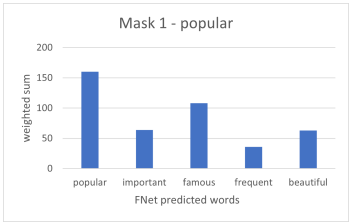
Figure 5.9. Histogram for Test 4 sentence - mask 0 to mask 7

5.2 Average-case scenario

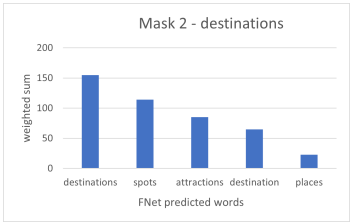
Sentences 6, 9, 11, and 15 are categorized as average performing sentences as they did not predict all the masks correctly but at least half of them. The weighted sum v/s FNet predicted words histogram is shown for all the sentences from Figure histograms of the results are presented for the sentences in Figure 5.10 and Figure 5.11. Sentence 11 is masked with 11 words and the frequency-exclusion technique correctly predicts 6 out of 11 words as high probability word similar to original FNet. Sentence 6 has 7 masked words among which 3 are predicted correctly by our modified encoder. Sentence 9 has been masked with 4 words and correctly predicts 2 of them. Sentence 25 has 7 masked words and the frequency-exclusion predicts 3 words at the first position.



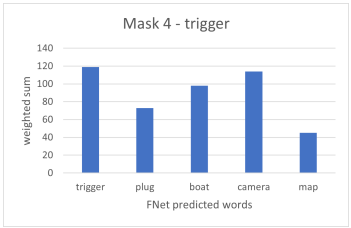
(a) Test 11 - mask 0



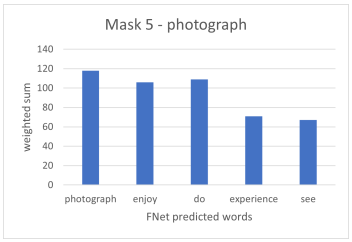
(b) Test 11 - mask 1



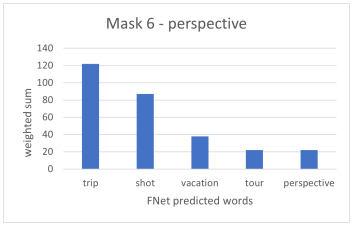
(c) Test 11 - mask 2



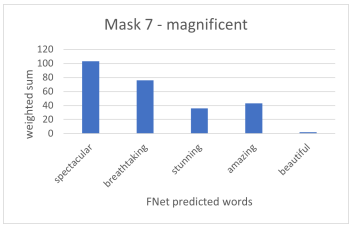
(d) Test 11 - mask 3



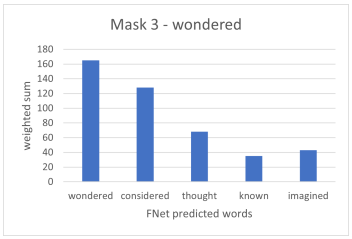
(e) Test 11 - mask 4



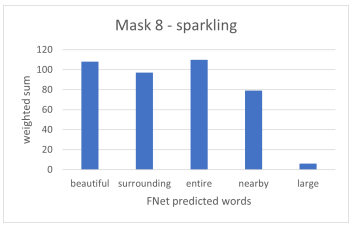
(f) Test 11 - mask 5



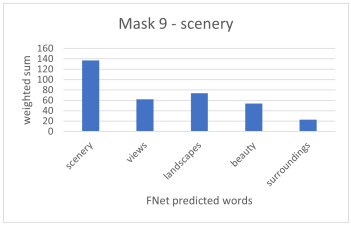
(g) Test 11 - mask 6



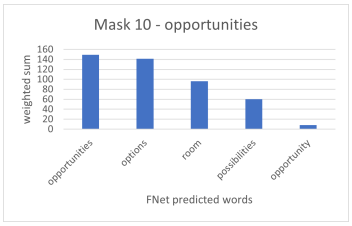
(h) Test 11 - mask 7



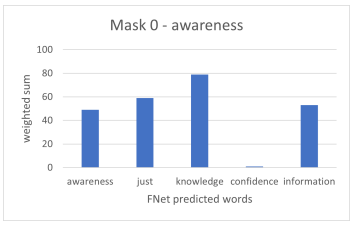
(i) Test 11 - mask 8



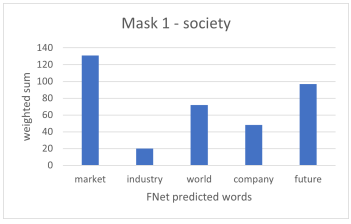
(j) Test 11 - mask 9



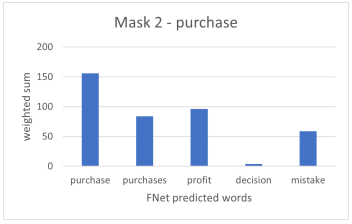
(k) Test 11 - mask 10



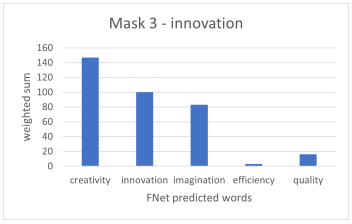
(l) Test 6 - mask 0



(m) Test 6 - mask 1

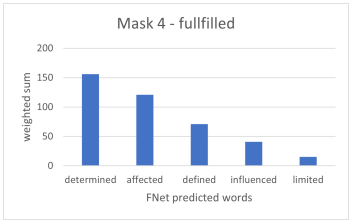


(n) Test 6 - mask 2

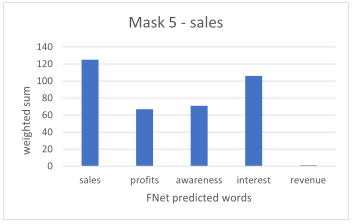


(o) Test 6 - mask 3

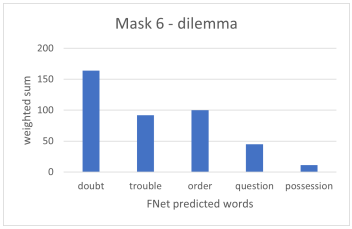
Figure 5.10. Histogram for Test 11 sentence - mask 0 to mask 10, Test 6 sentence - mask 0 to mask 3



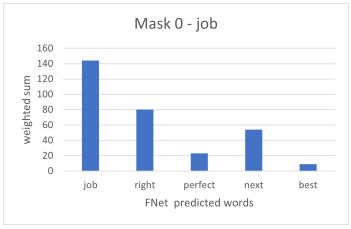
(a) Test 6 - mask 4



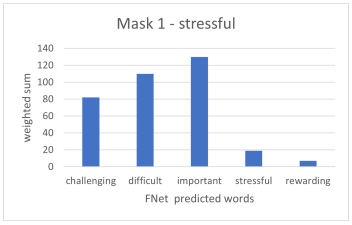
(b) Test 6 - mask 5



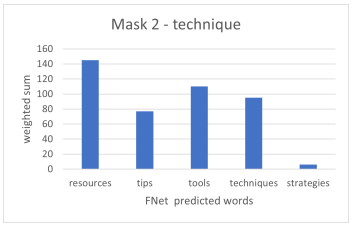
(c) Test 6 - mask 6



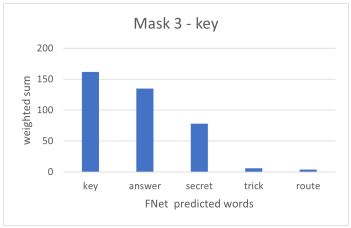
(d) Test 9 - mask 0



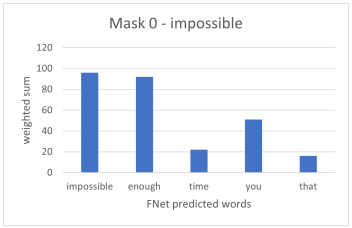
(e) Test 9 - mask 1



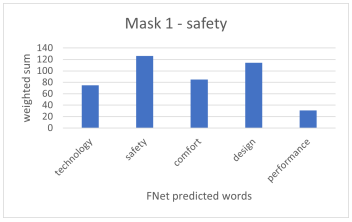
(f) Test 9 - mask 2



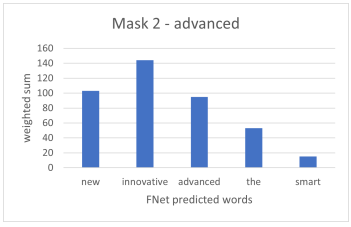
(g) Test 9 - mask 3



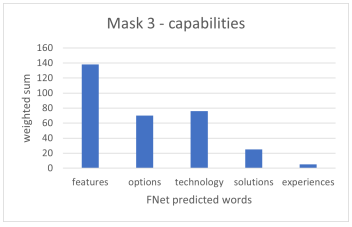
(h) Test 15 - mask 0



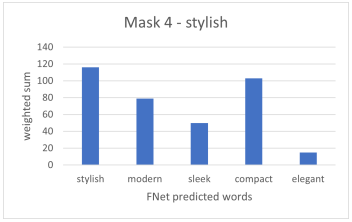
(i) Test 15 - mask 1



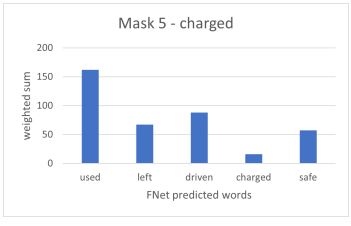
(j) Test 15 - mask 2



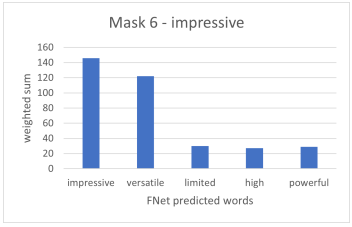
(k) Test 15 - mask 3



(l) Test 15 - mask 4



(m) Test 15 - mask 5



(n) Test 15 - mask 6

Figure 5.11. Histogram of weighted sum for Test 6 sentence - mask 4 to mask 6, Test 9 sentence - mask 0 to mask 3, Test 15 sentence - mask 0 to mask 6

5.3 Adverse-case scenario

Sentences 2, 3, 5, 8, 10, 12, 13, and 14 are categorized as adverse-case scenario as the model was able to predict the correct words to a maximum of 2 words. Sentence 2 predicts 1 out of 5 , sentence 3 predicts 1 out of 5, sentence 5 predicts 1 out of 5, sentence 8 predicts 1 out of 6, sentence 10 predicts 1 out of 8, sentence 12 predicts 1 out of 8, sentence 13 predicts 2 out of 12, and predicts 2 out of 10 masked words. The histogram of weighted sum v/s FNet masked words is shown from Figure 5.12 to Figure 5.15.

5.4 Improvement in FNet due to learning-by-exclusion

This section discusses the improvement in the predictions when the model was allowed to predict the words without seeing a specific range of frequencies which we call learning by frequency-exclusion. Along with correctly predicting the words as high probability words similar to FNet, the frequency-exclusion technique improves the FNet's ability to bring the words from a farther position in the prediction window to a nearer place. The table 5.1 shows the improvements made in 5 masked words. The masked word "scheduling" has shifted from 3^{rd} position to 2^{nd}, "care" from 4^{th} to 1^{st}, "performance" from 3^{rd} to 1^{st}, "preferences" originally not predicted in the window show up in 5^{th} position, and "critical" moved from 4^{th} place to 2^{nd} place.

Table 5.1. Learning-by-exclusion improvements in Sentence 2 experiments

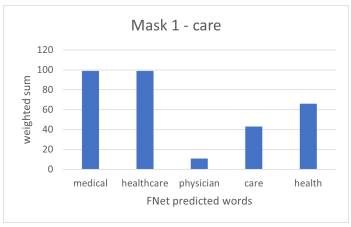
Mask	word	Original position	Improved position	Excluded frequencies
Mask 0	scheduling	3	2	[262:268]
Mask 1	care	4	1	[280:285]
Mask 2	performance	3	1	[280:285]
Mask 3	preferences	not present	5	[254:256]
Mask 4	critical	4	2	[242:247]

Sentence 3 predictions at mask 0 and mask 1 has improved from its original position as shown in table 5.2. Similarly sentence 5, sentence 6, sentence 7, sentence 8, sentence 9, sentence 10, sentence 11, sentence 12, sentence 13, sentence 14, and sentence 15 have proved to perform better than original FNet by employing frequency-exclusion technique. The results can be viewed from table 5.2.

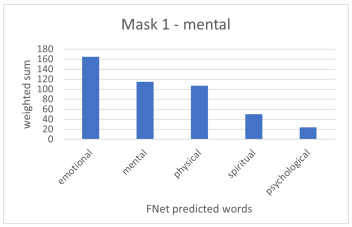
Table 5.2. Learning-by-exclusion improvements in Sentence 3 to 15 experiments

Note: The Excluded frequency column mentions only one frequency masking. But the same results are obtained in multiple frequency-masking as well

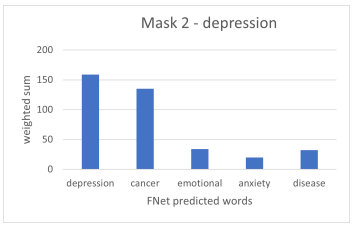
Exp No	Masks	word	Original pos	Improved pos	frequency excluded
sentence 3	Mask 0	worried	3	2	267:271
sentence 3	Mask 1	depression	4	1	267:271
sentence 5	Mask 0	essential	3	1	267:270
sentence 5	Mask 4	presence	4	1	267:270
sentence 6	Mask 3	innovation	2	1	257:259
sentence 7	Mask 7	prestigious	5	1	248:256
sentence 8	Mask 2	struggle	3	1	257:260
sentence 9	Mask 2	techniques	4	2	242:247
sentence 10	Mask 0	necessity	not present	2	256:258
sentence 10	Mask 2	still	2	1	243:247
sentence 10	Mask 4	cooking	3	2	272:274
sentence 10	Mask 5	relationship	2	1	243:247
sentence 10	Mask 6	memories	not present	4	257:259
sentence 11	Mask 0	spectacular	2	1	267:271
sentence 11	Mask 6	perspective	5	2	262:268
sentence 12	Mask 1	significant	4	2	267:270
sentence 12	Mask 6	forgotten	3	2	245:248
sentence 12	Mask 8	widespread	not present	5	267:270
sentence 13	Mask 2	tough	4	2	251:256
sentence 13	Mask 4	values	4	3	500:511
sentence 13	Mask 6	emerge	5	4	245:248
sentence 13	Mask 11	positive	3	1	251:256
sentence 14	Mask 0	greatly	not present	5	245:248
sentence 14	Mask 4	massive	not present	2	230:240
sentence 14	Mask 5	launched	4	3	245:248
sentence 14	Mask 9	culinary	5	4	265:268
sentence 15	Mask 1	safety	2	1	262:268
sentence 15	Mask 2	advanced	3	1	257:261
sentence 15	Mask 3	capabilities	not present	4	262:268
sentence 15	Mask 5	charged	4	2	230:240



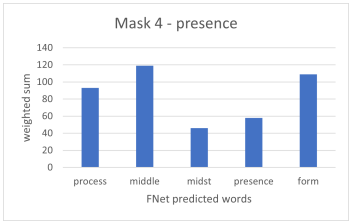
(a) Test 2 - mask 0



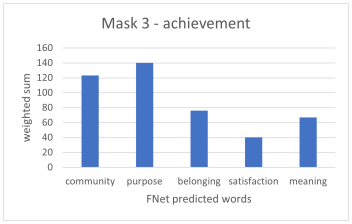
(b) Test 2 - mask 1



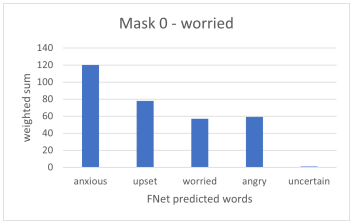
(c) Test 2 - mask 2



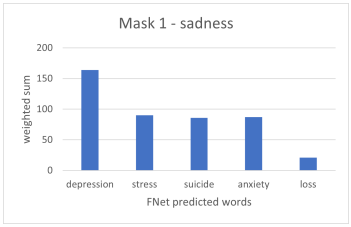
(d) Test 2 - mask 3



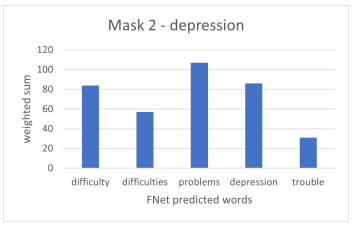
(e) Test 2 - mask 4



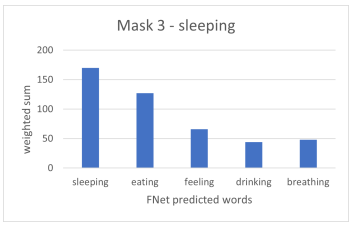
(f) Test 3 - mask 0



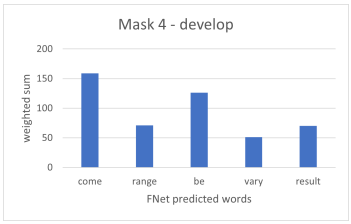
(g) Test 3 - mask 1



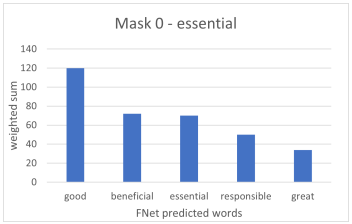
(h) Test 3 - mask 2



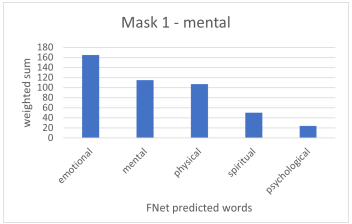
(i) Test 3 - mask 3



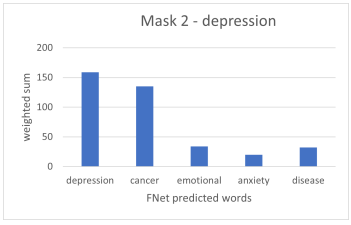
(j) Test 3 - mask 4



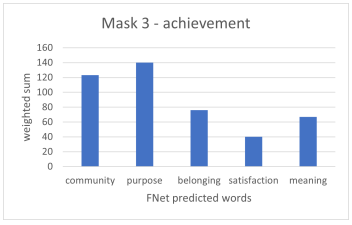
(k) Test 5 - mask 0



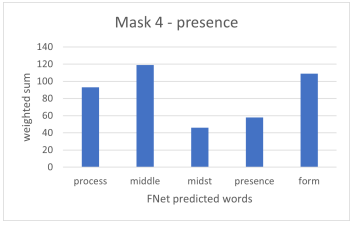
(l) Test 5 - mask 1



(m) Test 5 - mask 2

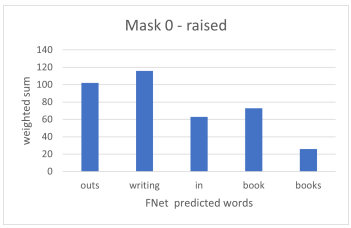


(n) Test 5 - mask 3

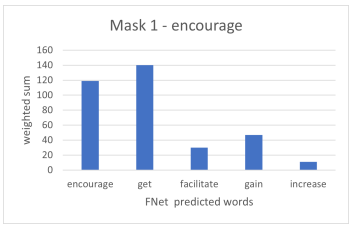


(o) Test 5 - mask 4

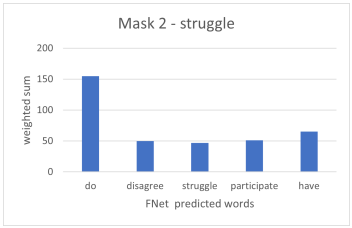
Figure 5.12. Histogram of weighted sum for Test 2 sentence - mask 0 to mask 4, Test 3 sentence - mask 0 to mask 4, Test 5 sentence - mask 0 to mask 4



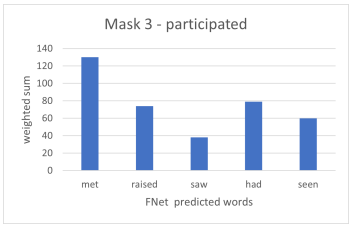
(a) Test 8 - mask 0



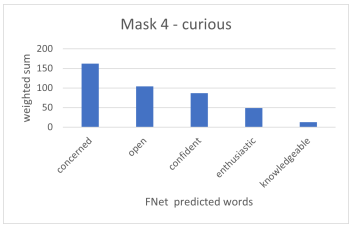
(b) Test 8 - mask 1



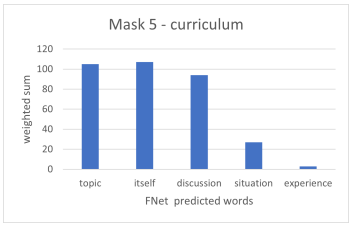
(c) Test 8 - mask 2



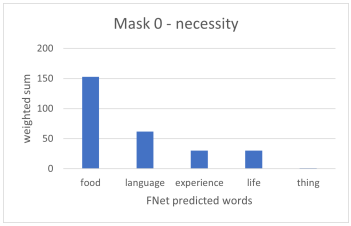
(d) Test 8 - mask 3



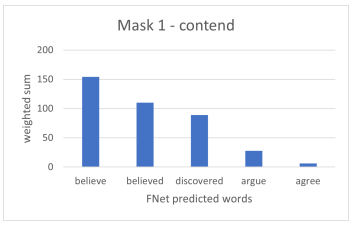
(e) Test 8 - mask 4



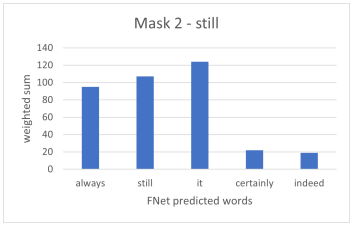
(f) Test 8 - mask 5



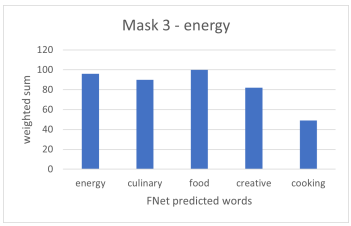
(g) Test 10 - mask 0



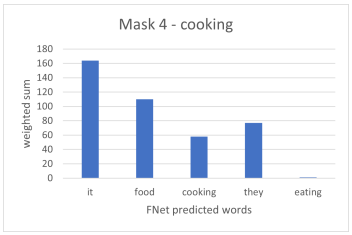
(h) Test 10 - mask 1



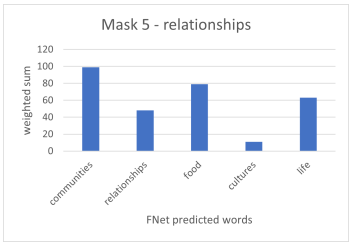
(i) Test 10 - mask 2



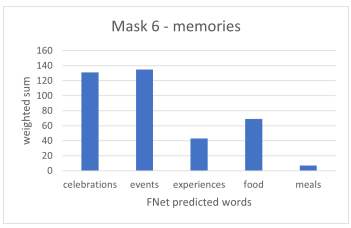
(j) Test 10 - mask 3



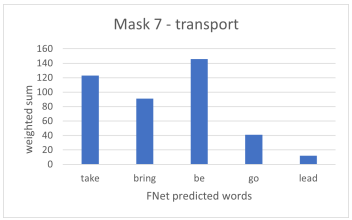
(k) Test 10 - mask 4



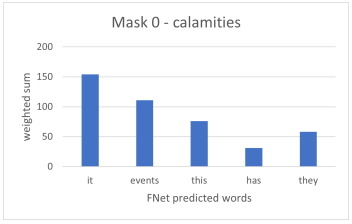
(l) Test 10 - mask 5



(m) Test 10 - mask 6

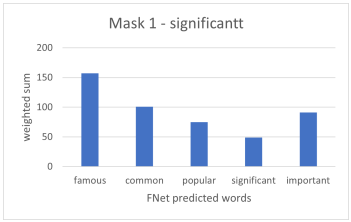


(n) Test 10 - mask 7

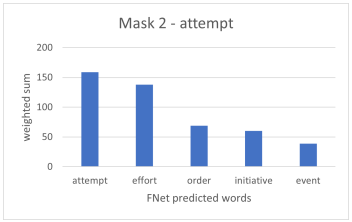


(o) Test 12 - mask 0

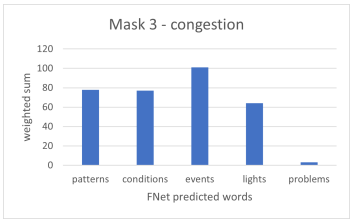
Figure 5.13. Histogram of weighted sum for Test 8 sentence - mask 0 to mask 5, Test 10 sentence - mask 7 to mask 4, Test 12 sentence - mask 1



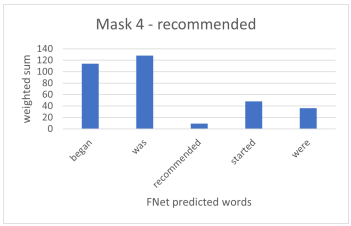
(a) Test 12 - mask 1



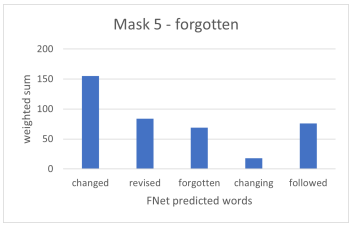
(b) Test 12 - mask 2



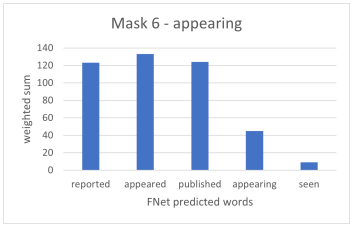
(c) Test 12 - mask 3



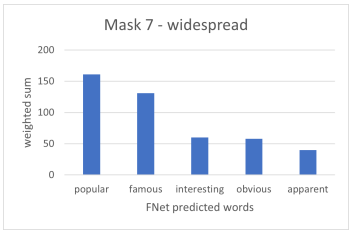
(d) Test 12 - mask 4



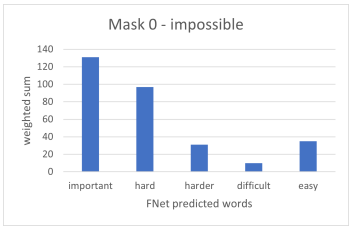
(e) Test 12 - mask 5



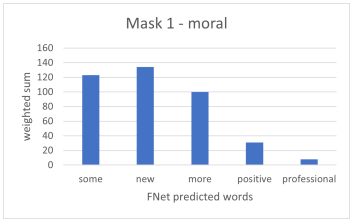
(f) Test 12 - mask 6



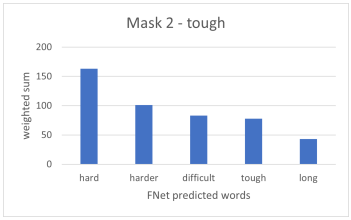
(g) Test 12 - mask 7



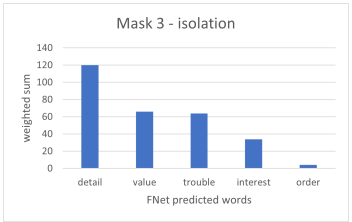
(h) Test 13 - mask 0



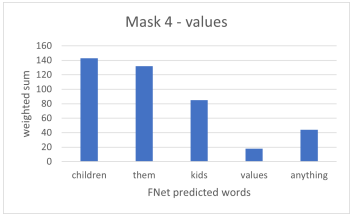
(i) Test 13 - mask 1



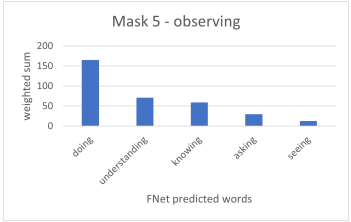
(j) Test 13 - mask 2



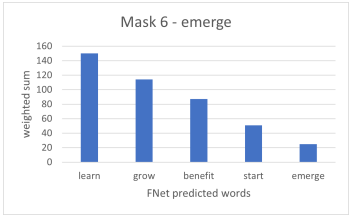
(k) Test 13 - mask 3



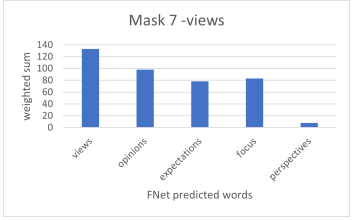
(l) Test 13 - mask 4



(m) Test 13 - mask 5

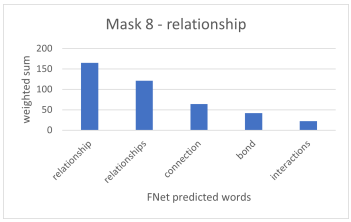


(n) Test 13 - mask 6

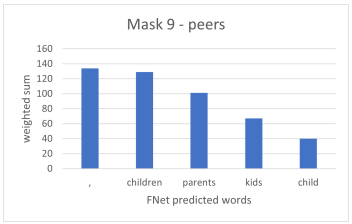


(o) Test 13 - mask 7

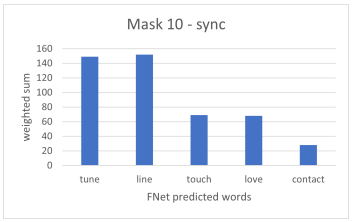
Figure 5.14. Histogram of weighted sum for Test 12 sentence - mask 1 to mask 6, Test 13 sentence - mask 0 to mask 7



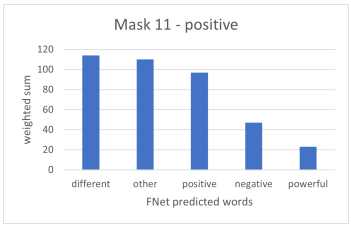
(a) Test 13 - mask 8



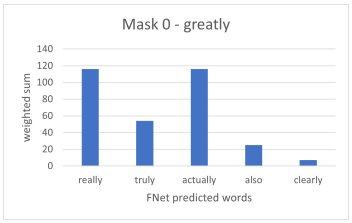
(b) Test 13 - mask 9



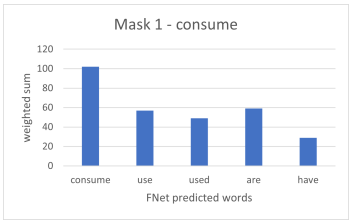
(c) Test 13 - mask 10



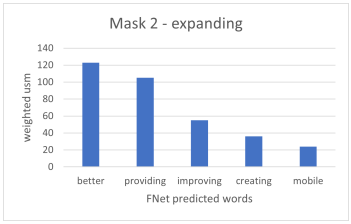
(d) Test 13 - mask 11



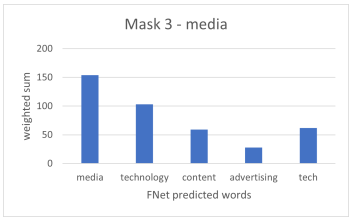
(e) Test 14 - mask 0



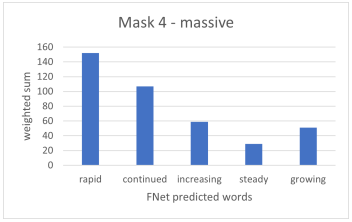
(f) Test 14 - mask 1



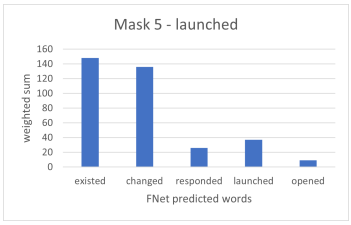
(g) Test 14 - mask 2



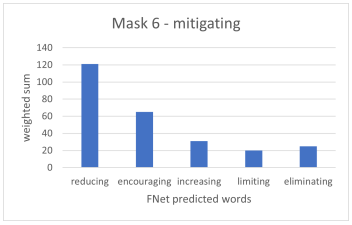
(h) Test 14 - mask 3



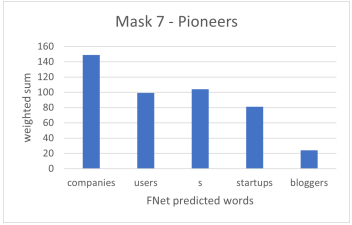
(i) Test 14 - mask 4



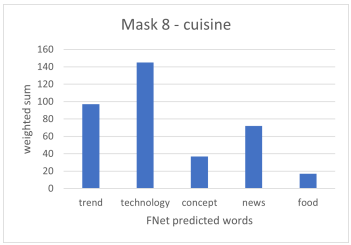
(j) Test 14 - mask 5



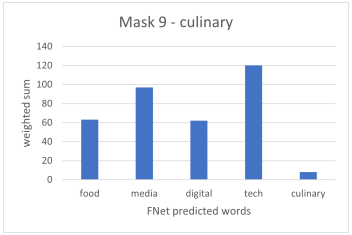
(k) Test 14 - mask 6



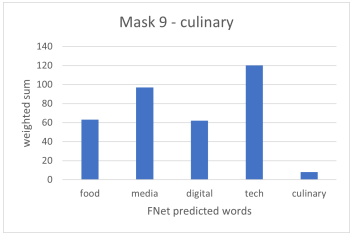
(l) Test 14 - mask 7



(m) Test 14 - mask 8



(n) Test 14 - mask 9



(o) dummy

Figure 5.15. Histogram of weighted sum for Test 12 sentence - mask to mask 6, Test 13 sentence - mask 0 to mask 7

Chapter 6: Discussion

FNet reduces the computational time of training the large corpus of text data to perform MLM and NSP using the Fourier Transforms. It is empirically shown that training is 80% faster in GPUs and 70% faster in TPUs when compared to BERT [18]. However, the inherent problem with the Fourier Transforms was not taken into account while FNet was built. In this thesis, we have shown the existence of spectral leakage with our dataset.

Figure 2.3 shows the spectral leakage problem in the red plot. The original frequency domain is infinite, and when the undersampling of data or windowing is performed over such domain, the information leaks from the original spectrum and spreads across. We consider 512 points as the infinite domain to show the spectral leakage problem. An original magnitude spectrum is generated over 512 points by averaging the 768 features in blue. Then, a windowed frequency of 496 points averaged at 768 features is laid on the same graph in red to observe the effect of spectral leakage. As we observe from Figure 2.3 when windowing is performed for 496 points, the magnitude of the frequency in red moves away from the original spectrum. As a result of spectral leakage, the words shift from their original position. When the model looks at that frequency in search of the word, it cannot find it because of "text-aliasing." The words would have already leaked to other positions due to leakage in the original spectrum. The model's ability to capture the correct words for the masks in MLM is reduced due to spectral leakage. This thesis shows how the model struggles to choose the masked word as it shifts to another position from the original position in the frequency spectrum of Fourier layer output. In the experimentation of 15 sentences, it can be observed some of the words struggle with synonyms problem and fail to appear as the first choice even though it is present in the prediction window. For example, in sentence 7, mask 7, the predicted words are "respected", "successful", "regarded", "ranked", and "prestigious". The correct word for mask 7 is "prestigious" and is predicted as the 5^{th} option in the window. Similar observations can be made in sentence 15 as well. Mask 1, 2, 3, and 5 were originally predicted as 2^{nd}, 3^{rd}, not present in the prediction window, 4^{th} position but improved the predictions to 1^{st}, 1^{st}, 4^{th}, and 2^{nd} positions respectively.

Similar observations are made in various experiments performed in the thesis. These results prove that FNet is struggling with the synonyms problem and fails to choose the correct word matching to the context of the sentence. Our approach of learning by frequency-exclusion is motivated by learning by word-exclusion [8], which has shown impressive results in improving the prediction quality of the FNet for MLM. The sentences are categorized as ace-case scenarios based on FNet's ability to predict the masked words as the first choice in the maximum number of cases. Our model also behaves similarly to the original FNet but uses fewer frequencies. The average-case scenario sentences could predict at least half of the masked words as high probability words. And in adverse-case cases scenarios, the original model and the frequency-exclusion technique could only predict lesser than half of the masked words.

Along with matching the FNet's performance, our model predicted the masked words in a high probability position in most of the sentences when FNet failed to do so. The complete proof of improvements achieved can be viewed in Table 5.1 and 5.2. The adverse-case scenario does not mean our approach was unable to predict words. It symbolizes FNet's inability to predict at least half of the words as the first choice, but in the given window, the word still existed or did not even occur. Our model performed equivalent to FNet in adverse-case scenarios and, in some cases, boosted the correct masked word from low probability position to high probability position. The results can be verified in Table 5.2. For example, sentence ten is categorized as an adverse-case scenario that only predicts one out of eight masked words at the first position. But the improvements our model has made in bringing the words closer to the correct position are visible in Table 5.2. In original FNet, Sentence 10 only predicts mask 3 correctly, masks 0, 1, 6, and 7 are not predicted with correct words in any place, and masks 2, 4, and 5 have the words predicted somewhere in the prediction window, not as the first choice. The frequency-exclusion technique shifts the correct word from originally predicted position 2 to 1 in mask 2, position 3 to position 2 in mask 4, and position 2 to 1 in mask 5. Also, the unpredicted words appear in a specific frequency-masking window in mask 0 and mask 6. We also expand the solution space of FNet to two domains, spatial and frequency domains, wherein the original FNet only produced results in the spatial domain. The original FNet only allowed to deliver a one-dimensional array with five options. But as we experiment with the multiple frequency-exclusion techniques, we generate a two-dimension solution space and give the user to select from various frequency-mask options.

We have performed multiple experiments on 15 sentences by following similar masking techniques. The results demonstrate that there are frequencies that FNet does not have to look at to make correct predictions which proves the correctness of our approach to learning by frequency-exclusion. We have observed that different sentences perform uniquely to the same masking techniques. This is because of the varying lengths of the sequences. As the length of the sequence varies, the magnitude spectrum resulting from the Fourier layer output also varies, and the words get distributed uniquely for each sentence. As a result, no single masking window works for all sequences, meaning our technique has no generalized model. But, the masking technique we used has given very promising results. This is one area in which further analysis can be driven to find a generalized model using frequency-exclusion to improve the overall performance of the FNet.

At this point, the performance of the our model is not evaluated using any measure like accuracy or precision. We solely rely on the observation that has been made during the experimentation and the results obtained. A histogram using weight assignment to the original predictions are drawn by counting the number of times the words appear in the frequency-exclusion technique and multiplying them with the weights. The histogram determines the correct word by the maximum value of a masked word that appear in the prediction window. Further research could be performed to expand the current work to find a qualitative measure to prove the correctness of the predictions made using frequency-exclusion technique.

References

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[2] Gunjan Chhablani, Abheesht Sharma, Harshit Pandey, Yash Bhartia, and Shan Suthaharan. NLRG at Semeval-2021 Task 5: Toxic spans detection leveraging bert-based token classification and span prediction techniques. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 233–242, 2021.

[3] Kamran Chitsaz, Mohsen Hajabdollahi, Nader Karimi, Shadrokh Samavi, and Shahram Shirani. Acceleration of convolutional neural network using fft-based split convolutions. *arXiv preprint arXiv:2003.12621*, 2020.

[4] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[5] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017.

[8] Stanka A Fitneva, Morten H Christiansen, and Padraic Monaghan. From sound to syntax: Phonological constraints on children's lexical categorization of new words. *Journal of child language*, 36(5):967–997, 2009.

[9] Michael S Gashler and Stephen C Ashmore. Training deep fourier neural networks to fit time-series data. In *International Conference on Intelligent Computing*, pages 48–55. Springer, 2014.

[10] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[11] Tyler Highlander and Andres Rodriguez. Very efficient training of convolutional neural networks using fast fourier transform and overlap-and-add. *arXiv preprint arXiv:1601.06815*, 2016.

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[15] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[16] Anders Krogh. What are artificial neural networks? *Nature biotechnology*, 26(2):195–197, 2008.

[17] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[18] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

[19] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A deep learning approach to visual question answering. *International Journal of Computer Vision*, 125(1):110–135, 2017.

[20] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

[21] MS Neethu and R Rajasree. Sentiment analysis in twitter using machine learning techniques. In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*, pages 1–5. IEEE, 2013.

[22] Harry Pratt, Bryan Williams, Frans Coenen, and Yalin Zheng. Fcnn: Fourier convolutional neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 786–798. Springer, 2017.

[23] Yashvardhan Sharma and Sahil Gupta. Deep learning approaches for question answering system. *Procedia computer science*, 132:785–794, 2018.

[24] Shengli Song, Haitao Huang, and Tongxiao Ruan. Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications*, 78(1):857–875, 2019.

[25] Shan Suthaharan. Deep learning models. In *Machine Learning Models and Algorithms for Big Data Classification*, pages 289–307. Springer, 2016.

[26] Shan Suthaharan. Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst*, 36:1–359, 2016.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. A survey of deep learning techniques for neural machine translation. *arXiv preprint arXiv:2002.07526*, 2020.

[29] Mahmood Yousefi-Azar and Len Hamey. Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68:93–105, 2017.

Chapter A: Appendix

Experiment	Sentence	Masks
1	I[MASK[to drive. But I am afraid of the vehicles on the[MASK[.	Mask 0 - want Mask 1 - road
2	2 [CLS] Appointment[MASK] systems are used by primary and specialty[MASK] clinics to manage access to service providers, as well as by hospitals to schedule elective surgeries.[SEP] Many factors affect the[MASK] of appointment systems including arrival and servicetime variability, patient and provider[MASK], available information technology and the experience level of the scheduling staff. In addition, a[MASK] bottleneck lies in the application of Industrial Engineering and Operations Research (IE/OR) techniques.[SEP] **Source**: `https://experts.umn.edu/en/publications/`	Mask 0 - scheduling Mask 1 - care Mask 2 - performance Mask 3 -preferences Mask 4 - critical
3	As we know, movies have the power to make or[MASK] the world. Movies can act as a[MASK] medium for bringing significant changes in society. Nowadays, people[MASK] watching movies than reading[MASK]. Visualization leaves a great[MASK] on the minds of people than imagination. So, the best use of this should be made such that our young[MASK] absorbs all the positive things from it. Movies can act as a great source of[MASK] as well. Many autobiographies have been made, through which we can learn a lot about our culture and history. But not everything shown in the movies must be absorbed like fights, violence, vulgarity, smoking, taking drugs, etc. It all depends upon our perception that how we take it. Youth is a stage of life when we are full of[MASK] and very enthusiastic about learning new things. So, if the positive things are taken into consideration, movies can act as a source of social awareness as well as motivation for achieving goals. **Source**: `https://mbakarlo.com/impact-of-movies-on-youth/`	Mask 0 - worried Mask 1 - sadness Mask 2 - depression Mask 3 - sleeping Mask 4 - develop

4	People want to know that they are being[MASK]. Really listen to what the other person is[MASK], instead of formulating your response. Ask for clarification to avoid[MASK]. At that moment, the person speaking to you should be the most[MASK] person in your life. Another important point is to have one conversation at a time. This means that if you are speaking to someone on the phone, do not[MASK] to an email, or send a text at the same time. The other person will know that she doesn't have your undivided[MASK]. Body language is important for face-to-face meetings and video conferencing. Make sure that you appear accessible, so have[MASK] body language. This means that you should not cross your arms. And keep eye contact so that the other person knows that you are paying[MASK]. **Source**: `https://atlascorps.org`	Mask 0 -heard Mask 1 - saying Mask 2 - misunderstanding Mask 3 - important Mask 4 - respond Mask 5 -attention Mask 6 - open Mask 7 - attention
5	Art is good for you. From beautifying the surroundings to helping to alleviate stress and discomforts, art is[MASK] for physical, emotional, and[MASK] wellbeing. Music, for example, is increasingly used by people to battle[MASK] and other mental illnesses. For art creators, including musicians, dancers, painters, and writers, creating or performing a piece of art is a cathartic experience that also provides a sense of[MASK]. There are numerous studies showing how people feel better whether by creating art or by consuming it or by simply being in the[MASK] of something art-related. **Source**: `https://mylenebesancon.medium.com`	Mask 0 - essential Mask 1 - mental Mask 2 - depression Mask 4 - achievement Mask 5 - presence
6	Advertising has brought in an advanced manner of building[MASK] about any product or a service in the[MASK]. It has enabled the consumers to have knowledge about the service or the product before making any[MASK]. Advertising has grown on the levels of creativity and[MASK]. The types of ads being produced these days have shown great influence on the minds of people by persuading them through attractive advertising tactics. Today the human needs are[MASK] by the source of advertisements.	Mask 0 - awareness Mask 1 - society Mask 2 - purchase Mask 3 - innovation Mask 4 - fullfilled Mask 5 - sales Mask 6 - dilemma

	Almost every product today is advertised in order to reach larger group of people. This benefits the company with increased[MASK]. For instance, if a person wishes to buy a car and is in[MASK] whether to make a purchase for it or not. In this case his target would set and he would be eager to learn more about it from the advertisements. The advertisement will in a way provoke him to buy. In this way advertisements control the mind of the interested person by fulfilling their want. **Source**: `https://www.easypresswire.com`	
7	Dr. Suthaharan is a[MASK] of Computer Science at the University of North Carolina at Greensboro (UNCG). He joined UNCG in 2001 as an Assistant Professor, and[MASK] to Associate Professor in 2005, then promoted to Professor in 2014. He also served as[MASK] of Undergraduate Studies for more than 10 years and as Interim Head in Fall 2015 at UNCG. He played a major role in leading the committee and maintaining ABET accreditation of the undergraduate program successfully. Dr. Suthaharan is also the[MASK] of the high impact and high quality textbook on the state-of-the-art topics of big data analytics and machine learning. Notably this book was reviewed by ACM Computing Reviews and received a "Reviewer Recommended" rating. Dr. Suthaharan's[MASK] interests mainly fall under the state-of-the-art themes of big data and machine learning. He studies advanced mathematical, statistical, and computational techniques to formulate smart[MASK] learning models and algorithms that can help accomplish secure big data analytics research in interdisciplinary settings. Dr. Suthaharan is a recipient of several[MASK], including the research awards from University of Pittsburgh Medical Center, Emory University, UC-Irvine, UC-Berkeley, University of Sydney-Australia, and University of Melbourne-Australia, and visited these highly[MASK] universities to perform collaborative[MASK]. **Source**: `https://sites.google.com/uncg.edu/shan-suthaharan/home`	Mask 0 - Professor Mask 1 - promoted Mask 2 - Director Mask 3 - author Mask 4 - research Mask 5 - machine Mask 6 - award Mask 7 - prestigious Mask 8 - research

8	In my class, I ask open-ended questions—sometimes I even call on students who don't have their hand[MASK]—and I put students in groups and pairs to[MASK] participation. But sometimes my students still[MASK]. One student in particular, Anise, never[MASK] or raised her hand in class, and I was quick to slot her as a student who just wanted to make it to June. But one day, halfway through the school year, Anise came to see me during her lunch break. When no one else was around, she seemed much more[MASK] about the subject matter. She asked questions, she listened to my answers, and we had a thorough conversation related to the class[MASK]. **Source**: `https://www.wgu.edu/heyteach`	Mask 0 - raised Mask 1 - encourage Mask 2 - struggle Mask 3 - participated Mask 4 - curious Mask 5 - curriculum
9	Preparing for the[MASK] interview can be one of the most[MASK] parts of the job search. Luckily, there are a number of tools and[MASK] to make this task easier. The[MASK] to any interview is being well prepared. It is important to write down, in advance, the answer to questions you are likely to be asked. This will help you focus your thoughts. **Source**: `https://www.shawnee.edu/career-resources`	Mask 0 - job Mask 1 - stressful Mask 2 - techniques Mask 3 - key
10	Food is a universal[MASK]. But it is only human beings who endeavour to transform food into something more. Several archaeologists and evolutionary biologists[MASK] that cooking was, and[MASK] is, crucial to our evolution. Not only did it simulate mastery over fire and necessitated innovation in tool making, but by eating cooked food, we are able to increase our[MASK] output for other creative pursuits. As primitive hunter-gatherer societies developed into more sedentary ones,[MASK] became a social activity. The preparing ans sharing of food came to define[MASK] within families as well as in the larger community. Food became central to community celebrations. Food is inextricably linked to occasions and[MASK]. Even a simple meal of dal and rice can[MASK] one to the sandy beaches of Sri Lanka if that's what you now associate it with. **Source**: `https://timesofindia.indiatimes.com`	Mask 0 - necessity Mask 1 - contend Mask 2 - still Mask 3 - energy Mask 4 - cooking Mask 5 - relationships Mask 6 - memories Mask 7 - transport

11	Lake Tahoe, known for its[MASK] scenery and year-round outdoor recreation, is one of the most[MASK] tourist[MASK] in both California and Nevada. But have you ever[MASK] what it would be like to live there full time? In August of 2019, I decided to pull the[MASK] and move from Los Angeles to Lake Tahoe. As an outdoor photographer, I fell in love with Tahoe's jaw-dropping beauty and wanted to be able to[MASK] it every day. After living in Tahoe for roughly 9 months, I decided to put together a list of pros and cons based on what my personal experience has been like. I hope this list is helpful to anyone who is contemplating a move to Lake Tahoe and wants an inside[MASK]. The scenery in Lake Tahoe is nothing short of[MASK]. In fact, it's the number one reason why I moved here. The phenomenal clarity and color of the lake is something you have to see to believe. Adding to the beauty are the snow-capped peaks and alpine forests that surround the[MASK] lake. Being able to wake up every day and enjoy such stunning[MASK] is a huge perk of living in Lake Tahoe. With its granite peaks, towering forests, and turquoise beaches, Lake Tahoe is an outdoor photographer's dream. In addition to the beautiful scenery, there's ample wildlife to photograph such as black bears, eagles, and coyotes to name a few. If action/adventure photography is your thing, there's plenty of[MASK] for photographing skiers, rock climbers, wakeboarders, paddleboarders, etc. **Source**: `https://www.gabriellaviola.com/post/living-in-lake-tahoe-pros-and-cons`	Mask 0 - spectacular Mask 1 - popular Mask 2 - destinations Mask 3 - wondered Mask 4 - trigger Mask 5 - photograph Mask 6 - perspective Mask 7 - magnificent Mask 8 - sparkling Mask 9 - scenery Mask 10 - opportunity
12	For centuries, the adjective "black" has been applied to days upon which[MASK] occurred. Many events have been described as "Black Friday", although the most[MASK] such event in American history was the Panic of 1869, which occurred when financiers Jay Gould and James Fisk took advantage of their connections with the Grant Administration in an[MASK] to corner the gold market.	Mask 0 - spectacular Mask 1 - calamities Mask 2 - significant Mask 3 - attempt Mask 4 - congestion Mask 5 - recommended Mask 6 - forgotten Mask 7 - appearing Mask 8 - widespread

	When President Grant learned of this manipulation, he ordered the Treasury to release a large supply of gold, which halted the run and caused prices to drop by 18%. Fortunes were made and lost in a single day, and the president's own brother-in-law, Abel Corbin, was ruined. The earliest known use of "Black Friday" to refer to the day after Thanksgiving occurred in the journal, Factory Management and Maintenance, for November 1951, and again in 1952. Here it referred to the practice of workers calling in sick on the day after Thanksgiving, in order to have a four-day week-end. However, this use does not appear to have caught on. Around the same time, the terms "Black Friday" and "Black Saturday" came to be used by the police in Philadelphia and Rochester to describe the crowds and traffic[MASK] accompanying the start of the Christmas shopping season. In 1961, the city and merchants of Philadelphia attempted to improve conditions, and a public relations expert[MASK] rebranding the days "Big Friday" and "Big Saturday"; but these terms were quickly[MASK]. The use of the phrase spread slowly, first[MASK] in The New York Times on November 29, 1975, in which it still refers specifically to "the busiest shopping and traffic day of the year" in Philadelphia. Although it soon became more[MASK], The Philadelphia Inquirer reported in 1985 that retailers in Cincinnati and Los Angeles were still unaware of the term. **Source**: `https://en.wikipedia.org`	
13	Some psychologists think values are[MASK] to teach, and it is certainly true that telling kids to be more honest, or diligent, or considerate, doesn't work any better than telling adults to be. But if values are impossible to teach, they are too important to leave to chance. In recent years, some schools have tried to add[MASK] development to their curriculum. But schools have a[MASK] time teaching kids values because they intervene too late, not to mention in too much[MASK] from the rest of the child's life. Worse yet, they are often at odds with what the child is learning at home about values.	

	Because the truth, of course, is that we do teach values to kids, daily, every minute of their lives. The question isn't whether to teach[MASK], only WHAT we are teaching. The way children learn values, simply put, is by[MASK] what you do, and drawing conclusions about what you think is important in life. Regardless of what you consciously teach them, your children will[MASK] from childhood with clear[MASK] on what their parents really value, and with a well developed value system of their own. Of course, parents are not the only source from which children learn values, and peers certainly influence your kids, especially as teenagers. And of course it's healthy for young people to think for themselves and develop their own world view, as much as we may want to influence our children. But research shows that the stronger your[MASK] with your child, the more her world, including the opinions of her[MASK] is filtered through the values she's picked up from you. Not to mention that if she has good self-esteem and a warm home life, she is more likely to pick friends who are more in[MASK] with your values. TV is an effective teacher. While some TV, especially public TV has many[MASK] social messages for young children, most TV, especially commercial television with advertising - teaches values antithetical to what most parents want for their kids. **Source**: `https://www.thefyi.org/15-ways-raise-child-great-values`	Mask 0 - impossible Mask 1 - moral Mask 2 - tough Mask 3 - isolation Mask 4 - values Mask 5 - observing Mask 6 - emerge Mask 7 - views Mask 8 - relationship Mask 9 - peers Mask 10 - sync Mask 11 - positive
14	The past 20 years have[MASK] shaped how and where we[MASK] media. In the early 2000s, many tech firms were still focused on[MASK] communication for work through advanced bandwidth for video streaming and other media consumption that is common today. Others followed the path of expanding media options beyond traditional outlets. Early Tech Pioneers such as PlanetOut did this by providing an outlet and alternative media source for LGBTQIA communities as more people got online. Following on from these first new[MASK] options, new communities and alternative media came the[MASK] growth of social media.	Mask 0 - greatly Mask 1 - consume Mask 2 - expanding Mask 3 - media Mask 4 - massive Mask 5 - launched Mask 6 - mitigating Mask 7 - Pioneers Mask 8 - cuisine Mask 9 - culinary

	In 2004, fewer than 1 million people were on Myspace; Facebook had not even[MASK]. By 2018, Facebook had more 2.26 billion users with other sites also growing to hundreds of millions of users. While these new online communities and communication channels have offered great spaces for alternative voices, their increased use has also brought issues of increased disinformation and polarization. Today, many tech start-ups are focused on preserving these online media spaces while also[MASK] the disinformation which can come with them. Recently, some Tech[MASK] have also approached this issue, including TruePic – which focuses on photo identification – and Two Hat, which is developing AI-powered content moderation for social media. Many scientists today are looking to technology to lead us towards a carbon-neutral world. Though renewed attention is being given to climate change today, these efforts to find a solution through technology is not new. In 2001, green tech offered a new investment opportunity for tech investors after the crash, leading to a boom of investing in renewable energy start-ups including Bloom Energy, a Technology Pioneer in 2010. In the past two decades, tech start-ups have only expanded their climate focus. Many today are focuses on initiatives far beyond clean energy to slow the impact of climate change. When we think of American classics our minds jump to the comforting standbys we grew up with: hot dogs, fried chicken and chocolate chip cookies. Over the years, this[MASK] has made such a mark on us that it seems like at some point or the other, everyone has a little love affair with it. With the splash of American grubs all over, it has transformed from delicious to glorious in India too.	

	And this 4th of July weekend, an all-American feast is just what you need to celebrate the nation's endless[MASK] creativities. We present our list of their 10 most delicious food items of all time. This selection covers the gamut from summertime staples to comfort food favourites. **Source**: `https://www.weforum.org,https://food.ndtv.com`	
15	When it comes to electric vehicles, it is near[MASK] to avoid talking about Tesla. Since the first Tesla Roadster was created in 2008, the company has since become synonymous with high-quality, luxury electric vehicles. Always pushing the envelope of modern vehicle[MASK] and convenience, Tesla is known for offering[MASK] engineering features built into their vehicles. For example, Tesla's full self-driving subscription options have changed the way people view their morning commute to work by offering innate self-driving[MASK] designed to make the road a safer place for every driver. From the Tesla FSD subscription to the sleek and[MASK] body of the Model X, Tesla truly has a product for nearly every type of consumer. In most instances of city or commute driving, Tesla vehicles have no problem maintaining their battery life. Capable of being[MASK] at home overnight, there is rarely an issue with battery range — especially when driving short distances. But what about long drives or road trips? As a leader in battery range amongst competitor electric cars, the battery range of Tesla vehicles is incredibly[MASK]. With some Tesla models capable of achieving between 393-525 km on a single charge, Tesla vehicles are highly dependable on longer drives too. **Source**: `https://www.loopit.co/blog/the-top-5-benefits-of-switching-to-tesla`	Mask 0 - impossible Mask 1 - safety Mask 2 - advanced Mask 3 - capabilities Mask 4 - stylish Mask 5 - charged Mask 6 - impressive

Table A.1. Sentences used for experimentation along with correct masked words