CONTEXTUAL TOPIC MODELLING USING BERT AS AN EMBEDDING LAYER

Trishul Chowdhury
Student ID: 931140

A thesis submitted in fulfilment of the
requirements for the award of the degree of
MASTER OF SCIENCE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Liverpool John Moores University

February 2021

Dedication

I dedicate this thesis to my beloved parents

Acknowledgement

I would like to thank my thesis supervisor, Mr Suvajit Mukhopadhyay, for his valuable and timely review and feedback for the completion of the thesis. I would also like to thank DR. Manoj Jayabalan from Liverpool John Moores University (LJMU) for his continued support and guidance through weekly and one-on-one sessions throughout the duration of the program.

I am grateful to my family for supporting and encouraging me throughout the duration of my study and writing this thesis.

Regards,
Trishul Chowdhury

Abstract

E-commerce (Electronic Commerce) is a kind of industry where the buying, selling, and interaction between buyers, manufacturers, retailers of different products, and services are conducted over electronic systems such as the Internet.
In the e-Commerce ecosystem, effective growth mostly relies on great customer care service. It involves getting to know your customers (sellers and buyers) so well that the platform owner can anticipate their needs and exceed their expectations.
Product reviews one of the major keys for the consumers to get an idea about the product and that can be leveraged to a robust recommender system as well.
Topic Modelling system of customer's reviews using the traditional NLP techniques based on bag-of-words (BOW), TF-IDF and hierarchical Bayesian models rarely adopt contextual information. Hence, these techniques do not consider an enormous amount of serviceable (Contextual and semantic) knowledge of a document, due to the sparsity of the review data.
To solve this gap of data, the researchers have developed different advanced pre-trained models by training the general-purpose language representation model using an enormous amount of unannotated data on the web.
In this study, we will focus on a new state of the art model for the contextual topic identification problem incorporating a new open-sourced NLP technique called **B**idirectional **E**ncoder **R**epresentations from **T**ransformers, or **BERT** (Developed by Google Brain) as a sentence embedding layer with the probabilistic topic assignment vector by LDA.

Keywords: Amazon, Text Mining, LDA, Review Analysis, E-commerce, Topic Modelling, BERT, Sentence Embedding, Vector Space, Transformers, Self-Attention, Contextual Topic Modelling, Auto Encoder

Table of Contents:

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Expansion
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long-Short-Term-Memory
PCA	Principal component analysis
HBN	Hierarchical Bayesian Network
SVD	Support vector machine
TM	Topic Modelling
LDA	Latent Dirichlet Allocation
MCME	Markov chain monte carlo
VI	Variational inference
DTM	Document term matrix
TF-IDF	Term Frequency- Inverse Document Frequency
BOW	Bag-of-Words
PLSA	Probabilistic Latent Semantic Analysis
NPMI	Normalized Pointwise mutual information
NLTK	Natural Language Toolkit
SciBERT	Scientific BERT
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
TPU	Tensor Processing Unit
AE	Auto Encoder
GPU	Graphics Processing Unit

CHAPTER 1: INTRODUCTION

1.1 Background of the study

Amazon is known for its disruption of well-established industries through technological innovation and mass scale (Wikipedia, 2020a).
Product reviews are a very essential tool for Amazon's customers to decide whether to buy the product.
Natural language processing (NLP) is a subfield of explainable AI associated with linguistics and computer science. It focuses on the interactions between computers and human language. Typically it is a research area to study how computers learn to process and analyse large amounts of natural language (Wikipedia, 2020b)
Topic Modelling is one such NLP task which is largely used in real-life business problem like sentiment analysis of customers, analysing the trending topics among them about the newly launched product, as a solution of explainable AI.
Benefits of Topic modelling in E-Commerce:
With the help of task like Topic Modelling, machines can pick the phrases and words are generally used by human beings while they do a generic search or review a particular product. It customizes the searches result for users using a search engine. The system finds the user's search result by using its cognition of the user's language and the structure of the sentence used. It also helps the manufacturer to focus on the key points raised in customer reviews about the performance of the products.

The modern world's E-Commerce retailers can use Text mining techniques in different ways:
Sentiment Analysis: AI makes it easier to analyse the emotional and temporal behaviour of customers and can classify it into different classes as, negative, positive, and neutral outcomes.
Text Recognition: AI converts the text and character as numeric data to make the computer understand it. For example, search engine uses this kind of language model to convert the search text of customer into a machine coded text. It provides the answer to the question instead of showing the search results.
Semantics: Humans can efficiently understand the context of the written or spoken words. However, it is difficult to make computers learn the latent context. Machines have been fed a lot of incoherent and unstructured raw data. With the help of effective mathematical algorithms, it understands the semantic of the text.
Customer service centre dynamics: Call centres agent and their interaction with customers can be handled by the AI efficiently. The business houses can be benefited through real-time dynamics. A single server can handle an enormous number of calls by searching the queries in a flash and give the best possible response to the customer using the embedded intelligence like DEEP NLP.
In traditional natural language understanding (NLU) tasks, there is a hierarchy of lenses through which we can extract meaning — from words to sentences to paragraphs to documents. At the document level, one of the most useful ways to understand text is by analysing its topics. The process of learning, recognizing, and extracting these topics across a collection of documents is called topic modelling.
However, most of the traditional Topic modelling techniques are based on hierarchical Bayesian techniques which fails to capture the contextual semantic topic of the documents.

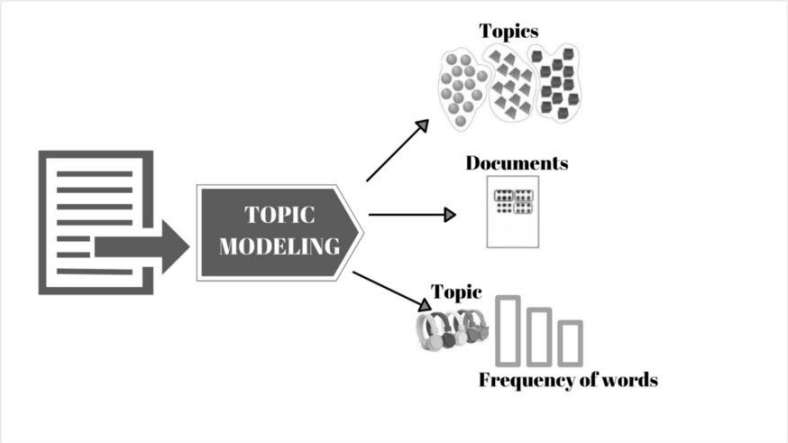


Figure 1.1: Hierarchical lenses of Topic modelling

In this research, we will be focusing on a new state of the art for topic modelling incorporating BERT as a sentence embedding layer.
Unlike recent language representation models (Clark et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabelled text which is being trained with information from both the left and the right side of a token's with the idea of Masked Language Model(MLM) and multilayer Self-Attention mechanism. It is modelled in such a generic way so that it can be used in different NLP related tasks (Transfer Learning).
We will focus on combining the BERT sentence embedded vector with the traditional Bayesian technique LDA to understand the context of the customer reviews more precisely.

1.2 Problem Statement

This section will start with an overview of related works in the field of Deep NLP and finally will explain the problem statement that the study will focus on.
LDA of TF-IDF is sufficient for identifying topics in the coherent texts when they are able to find the most frequent words.
However, when the choice of the words and the meaning of the sentences are incoherent, extra contextual information is required to represent the idea of the texts(Shao, n.d.).
Adding more contextual knowledge to the model improves the coherence.
The essence of building a robust Topic modelling technique lies in the advanced methods of Word embedding which is highly leveraged by the latest development of Deep Learning. The recent development of topic models based on neural networks are gaining huge attention, while BERT-based models are pushing the general neural models as state of the art(Bianchi et al., 2020).
The most popular embedding techniques are described as follows:

1.2.1 Word Embedding

In word embedding, we would generate an embedding for each word in the set. The simplest method would be just like one-hot encoding for the sequence of words. Word embedding not only converts the word but also identifies the semantics and syntaxes of the word to build a vector representation of the information. However, we miss the entire context of the sentence and the document.

1.2.2 Sentence Embedding

The new state of the art for understanding sequential text is Sentence embedding.

Sentence embedding techniques represent entire sentences and their semantic information as vectors. This helps the machine in understanding the context, intention, and other nuances in the entire text.
BERT representations have been largely used by the research community in a diverse set of NLP applications (Rogers et al., 2020).
Currently, BERT is the most effective solution for sentence embedding.

1.2.3 Doc2Vec Embedding

Doc2Vec embedding is an extension of the Word2Vec model incorporating 'paragraph vector' at the document level. It is an unsupervised model.
There are two main ways of applying this technique:

- **PV-DM (Distributed Memory version of Paragraph Vector):** Here, the model predicts the next word given a set of words. The word vectors are being shared among all the sentences and a paragraph vector to the sentences. Then the paragraph and word vectors are combined to get the final sentence representation.

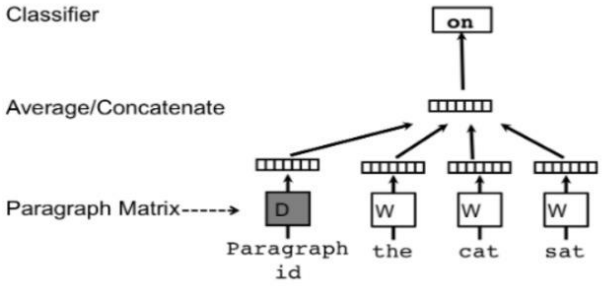


Figure 1.2: PV-DM framework (Le and Mikolov, 2015)

- **PV-DBOW (Distributed Bag of Words version of Paragraph Vector):** This is based on the skip-gram model. Here the model predicts the source(sentence) of the word by sampling random words across the documents.

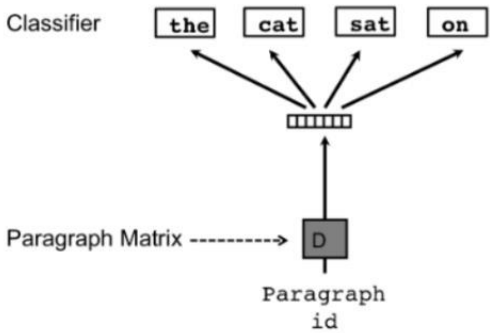


Figure 1.3: PV-DBOW version of paragraph vectors(Le and Mikolov, 2015)

1.2.4 BERT Embedding

BERT is basically a pre-trained deep bidirectional representation from unlabelled text by training from both directions of the text. BERT can be tuned incorporating different output layer and leverage to a new state of the art models for various NLP related use cases. There is no need to build any specific model from scratch to achieve any particular task (Devlin et al., 2019).
BERT is based on the OpenAI GPT (Hugging Face AI, 2019) in the stacking of transformers, but uses bidirectional ones instead of just left-to-right. The figures below demonstrate this distinction:

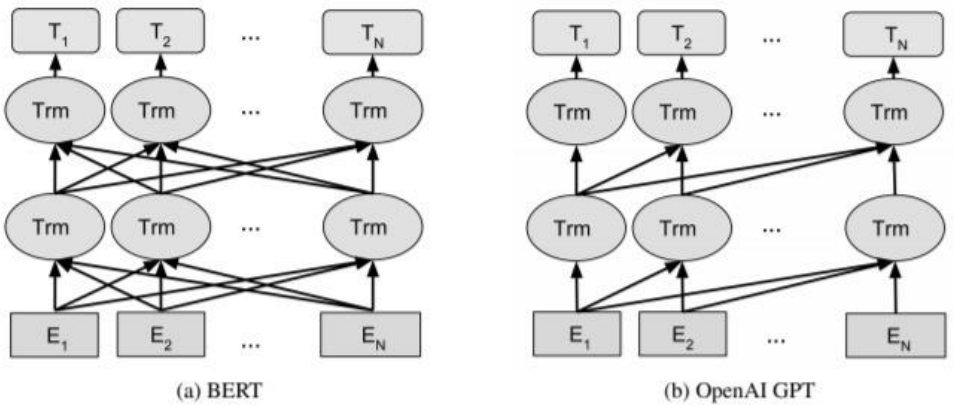


Figure 1.4: BERT uses bidirectional transformers, as opposed to GPT which uses only left-to-right(Devlin et al., 2019)

In a study (Wang and Ranganathan, 2019), demonstrated the use of three variations of attention models like BERT viz., vanilla BERT, CNN + BERT, and BERT + Linear and compared it with LSTM+GRU models. Few good insights were that BERT did not require any embedding as it does the embedding on its own and still performed better than the LSTM + GRU model.
The study also gives some reference examples wherein the data labelled is incorrect whereas BERT predicts the correct class but considering the target and predicted does not match the prediction score decreases. One of the reasons for BERT to perform well can also be linked to the fact that it tends to underfit and generalize which makes it a good choice for relatively noisy data.
Amazon receives thousands of orders daily, and in-turns receives thousands of customers reviews every day.
Considering their digital product, customers generally talk about so many different topics in the same review. Customers can have different concerns like server issues, hacker issues, processing speed, product aesthetics, budget, display and screen size, customized App performance in those products, issues related to integrated hardware like Bluetooth, memory drive, etc.
In this research work to understand the underlying context-based concerns of customers or their appreciation about the product, document level embedment using a Pre-trained model along with classical TM technique will be experimented to enhance the efficiency of latent topic identification task.

1.3 Aim and Objectives

The aim of this study is to propose an approach to enhance the contextual topic identification capabilities of Bayesian method-based models on a large incoherent corpus of documents.
The goal of this study is to identify the hidden topic of customer reviews of Amazon's digital products.
Considering the aim of the study the research objectives are articulated as follows:

- To analyse the topic identification capability of traditional Bayesian methods when customer review data is hugely sparsed.
- To investigate the potential of document embedding techniques and pre-trained NLP model BERT to understand the underlying idea of the review.
- To develop an approach that combines the sentence embedding and Bayesian techniques to get a more accurate context-based topic discussed among the customers.

1.4 Research Questions

A comprehensive literature review indicates there have been multiple research works done in the field of Text mining, specifically in Topic Modelling. However, most of the Topic modelling techniques are developed using Probabilistic Latent Semantic Analysis like PLSA or its Bayesian version like LDA.
There is no research work found on Topic Modelling on the customer reviews of Amazon's most successful Digital products incorporating new state of the art NLP model BERT as an embedding layer.
The literature review leads the below-mentioned research questions, which will be addressed in this study:

- What are the latest topics of discussion among the customer reviews on Amazon's Digital products?
- The reviews don't consist of one single topic. How Bayesian Topic modelling techniques work in highly sparsed data since customer's review mostly incoherent?
- How interpretation of the reviews depends highly on the context?
- What are the performance evaluation criteria of the Bayesian methods like LDA?
- What potential embedding techniques can enhance the performance of the topic identification task?
- How incorporation of BERT with a self-attention mechanism along with a traditional Bayesian Technique can enhance the performance?

1.5 Significance of Study

In this digital era, real-world data on the web is highly incoherent, especially when it is a part of human conversation or communication.
In a fixed amount of time or the same paragraph or document or any specific universe, there can be different topics of discussion which are latently connected and hidden. This research can open up a new direction in the field of Topic Identification tasks when the documents are incoherent and abrupt.
This study will consider the approach which resembles the humanized way of understanding the text (using techniques like Self Attention, Masked language modelling) to understand Amazon's Digital product's customers review when it is incomprehensible, unclear, or even confusing. Hence it will be helpful for the E-commerce websites to assess the sentiment of their customers and leverage it to a very engaging and intuitive recommendation system as well.
Apart from the business aspect, the study will also open up a new future scope of implementation of state-of-the-art model to solve to Text mining problem, by using a newly developed Language models.
The latest developments in NLP have given rise to innovative model architecture like GPT-3 along with BERT.
Different pre-trained language models have made machine learning more user friendly. People with very minimum technology background can get their hands-on building ML applications, without training a model from scratch. To solve different problems like making accurate predictions using TL , feature extraction, most of the NLP models are typically trained on a wide range of data, in billions.
GPT-2 and GPT-3: OpenAI's Latest Language Model:
GPT-2 was introduced by OpenAI researchers (Radford et al., 2018). It has been trained on a 1.5Billion parameters Transformer model.

GPT-2 is an auto-regressive model while BERT is not. GPT uses transformer decoder blocks in its architecture. GPT uses masked self-attention i.e. it doesn't allow the tokens to look to its right tokens. Being trained on a very big corpus leads to good performance.

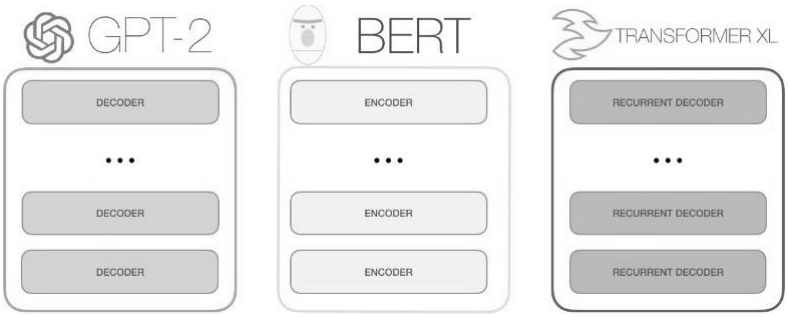


Figure 1.5: Architecture comparison of GPT2, BERT and XLNet(Transformer XL) (Alammar, 2019)

GPT-2 is a sequence-to-sequence model. It can be fine-tuned for the text summarization. Different researches describe the ability GPT-2 to summarize text and was first tested on the Daily Mail dataset and with the CNN.
In this research, the main purpose was to induce summarization behaviour of the TL;DR text. The model was developed to generate around 100 tokens with top-k random sampling with a value of k=2. A low value of k helps in reduced repetitiveness and encourages more abstractive summaries.
In the initial research work, fine-Tuning Language Models from human Preferences has shown the way of fine tuning GPT-2 774M to summarize texts in a more cognitive way. The model was trained and fine-tuned on 60k labels.
Subsequent to the original paper (Liu et al., 2018)proposed a different formation of the transformer block which is only based on "Transformer-Decoder".

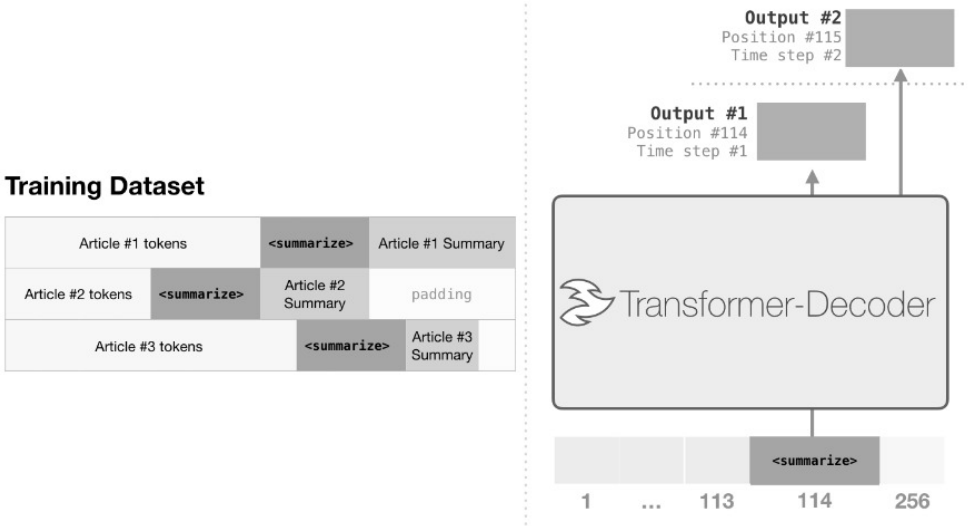


Figure 1.6: GPT-2 (Decoder Transformer)(Alammar, 2019)

Tasks like Machine Translation can be addressed by only a decoder-only transformer:
(Brown et al., 2020)The model architecture of GPT 3 has been kept the same as GPT2. Increasing the number of parameters has achieved the State-of-the-Art results on multiple evaluation benchmarks and metrics.

GPT-3 and BERT both have been relatively new for the industry, but their unique and robust state-of-the-art model performance has made them the winners among other models in the field of NLP. GPT-3 has been trained on 175 billion parameters, which is 470 times bigger in size than BERT-Large.
BERT needs an extensive fine-tuning process where users have to gather data of examples to train the model for specific tasks. Text-in and text-out APIs of GPT-2 allows the users to reprogram and access it as per the instruction.
The user has to train the BERT model on a separate layer on sentence encoding for the tasks like sentiment analysis or question answering.
However, GPT-3 uses a few-shot learning processes on the input token to predict the output result.
GPT-3 works perfectly in most of the NLP tasks by conditioning with a few examples.
Using **Transfer Learning**, pre-trained language models based on Transformer in tasks like Topic modelling or Text summarization can open up a new area of study and improve effective ways of implementing new state of the art models to solve modern business problems across all the domain.

1.6 Scope and Limitations of the Study

The scope of the research is limited to the following factors:

- The research focuses on the development and evaluation of different Deep NLP models incorporating a new state of the art module, BERT as a sentence embedding layer.
- The scope can also include parallelizing the Latent Dirichlet Allocation using all CPU cores to parallelize and speed up the training of the model.

Considering the high computation requirements and the fixed time frame below-mentioned limitations are set on this study:

- The data for the research is directly taken from the Datafinite repository. The collection or extraction of raw data is not under the scope of the study.
- The research does not have much opportunity to compare the final model performance with the models based on word vector, lda2vec.
- BERT training from scratch is out of the scope due to resource constraints. Hence, research will focus on pre-trained BERT by tuning the hyperparameter to get the optimum solution.
- The use of different embedding layers with other versions of BERT, SciBERT, or BART is out of the scope of this study.
- The use of newly developed language model GPT-3 in Topic modelling task is also out of the scope of this study.
- Building Customized reports for different products which could give some more interesting information.

1.7 Structure of the Study

- **Chapter 1** introduces the problem domain of the research. It provides a necessary background on traditional Topic Modelling tasks that is required for an in-depth understanding of the study. This section explains the problem statement and the aims and objectives of the study that will address the problem statement. It also discusses the research questions that the study will try to answer and its desired contribution in the space of using a pre-trained language model in the task of topic modeling that leads to infer the significance of this research work. Finally, the scope and the

limitation of the study will focus on the expected coverage and limitations of implementing these models considering different factors.

- **Chapter 2** will start with the basics of NLP and the history of Topic Modelling, followed by the Distributional Semantic models and their implementation process which includes Word vector and word embedding techniques. The study will also focus on Deep Learning and advanced transformer-based architectures and their usage in various research. This section will also review different traditional Topic Modelling techniques and will end with identifying the gap in existing literature and ideas on improving them few of which are planned to cover in this research.
- **Chapter 3** discussed the proposed methodology in detail. This section will cover data formatting, transformations, pre-processing required for the dataset. It also covers the embeddings and algorithms planned to be used in this study with concluding the evaluation metrics and criteria.
- **Chapter 4** is broadly divided into four sub-sections. The first sub-section discusses the design for data pre-processing and data transformation as per the methodology discussed in chapter 3. This section discusses each step of data processing and data transformation in detail. The second sub-section is the EDA section. It focuses on exploring the underlying contextual hidden layer of the corpus of documents that also include the inclined sentiment of the topic of discussion and its polarity. The third subsection is the model building segment. Here, the research has mentioned the motivation of each approach and detailed implementation of three major models. Finally, in the final section, it talks about the hardware and software resources used for the study.
- **Chapter 5** focuses on the results and the performance metrics generated from all the experiments outlined in chapter 4.This section starts with the performance metric for different HBN models with different inference algorithms to enhance the performance.Then this section discusses the explanations of the results of the Transformer-based models with the use of the Bertopic framework and examines two different pre-trained BERT models.Finally, the results of the model, using a combination of LDA and HBN vectors have been examined and tuned to optimum hyperparameters to get the best results. These explanations are then evaluated based on different qualitative evaluation methods which, help in a comparative analysis of different topic modeling techniques.
- **Chapter 6** concludes this research by discussing how the study meets the aim and objectives set proposed in chapter 1. This section also discusses the research questions and how comprehensively the study could answer them. The contribution of this study in the space of NLP, using advanced language models and a clear traceability of the proposed methods and contribution of this study have been discussed. Finally, this section discusses the limitations of this study and future recommendations from this study that can be used to extend research in the domain of NLP and Transfer learning.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction:

The essence of the Topic modelling is that context of our document is actually hidden, or "latent," which we cannot observe initially. The sole purpose of this task is to identify the latent topics that give the actual context of our document and corpus.
In a huge corpus containing a ton of documents. The underlying latent topics will come out through the process of topic modelling.
A naïve approach is to list the keywords as per their frequency of occurrence, which is termed as TF. In this approach, the actual topic may not be in the top list of keywords, and the topic is hidden which is not feasible to derive from the text of the document (Tokunaga and Iwayama, 1994).
Topic modelling is a computational technique to find the hidden patterns of co-occurrence in the set of documents (Tokunaga and Iwayama, 1994).
TM is one of the most important research area in the field of information retrieval (Onan et al., 2016).

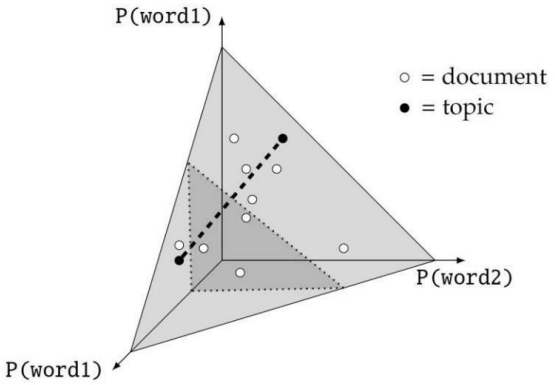


Figure 2.1: Geometric representation of topic modelling (Emmery, 2014)

The above diagram depicts the probability of identifying a certain word. Each document can be represented as a white point. Each word is associated with some weight in the document. Hence topics can also present in the space with black points. Generally using this space, a linear classification problem can be solved and a decision boundary can be found that divides the documents into different topics.
LDA is one of the popular topic modelling techniques.
In this hierarchical Bayesian method probability distribution of words represent topics and a probability distribution over topics represents documents. (Steyvers and Griffiths, 2010).
The main categories in probabilistic topic modelling methods are:
Inter and Intra document correlation, supervised and temporal probabilistic model and, basic traditional methods (Daud et al., 2010).
Most formative works in the topic modelling research area are LSA, PLSA, and LDA(Alghamdi and Alfalqi, 2015).

2.2 History of Topic Modelling

Topic models are also referred to as probabilistic models, where the underlying statistical algorithms discover the latent semantic structures of an extensive document.
In the era of digital transformation, the amount of created data through different means is going beyond our processing capacity.TM helps to get the insights of a large collections of unstructured text bodies. Originally it was developed as a text-mining tool. Topic modelling has so many applications in different fields such as bioinformatics (Blei et al., 2010) and computer vision (Cao and Fei-Fei, 2007).An initial topic model was described by Papadimitriou, Raghavan, Tamaki and Vempala in 1998 (Papadimitriou et al., 2000), PLSA was created by Thomas Hofmann (Hofmann, 1999). Currently LDA is the most common topic model in use. It is a generalization of PLSA.It was first proposed by Andrew Ng , Michael I. Jordan and David Blei in 2002, LDA introduces sparse Dirichlet prior distributions over document-topic and topic-word distributions, encoding the intuition that documents cover a small number of topics and that topics often use a small number of words(Blei et al., 2003) .

2.3 Distributional Semantic Models

The basic idea that we use to quantify the similarity between words is that words that occur in similar contexts are similar to each other.
It is important to represent words in a format that encapsulates its similarity with other words. There are two broad techniques to represent words vectors:

- The term-document occurrence matrix
- The term-term co-occurrence matrix

2.3.1 Occurrence Matrix

The occurrence matrix is also called a term-document matrix since its rows and columns represent terms and documents/occurrence contexts respectively. Term-document matrices (or occurrence context matrices) are commonly used in tasks such as information retrieval.

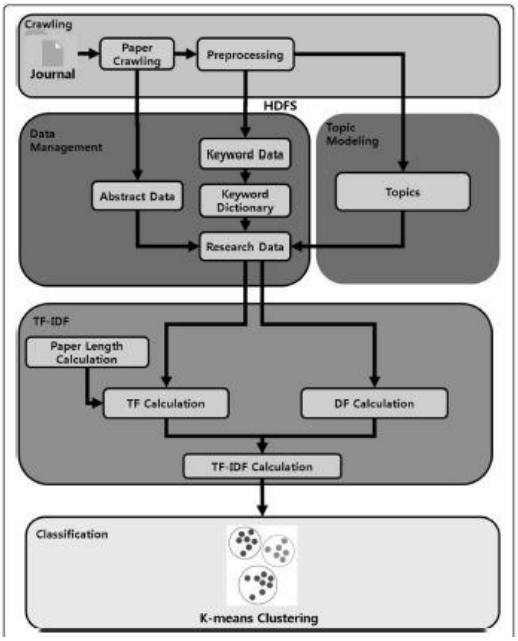


Figure 2.2: Generic TF-IDF based topic modelling system flow diagram(Kim and Gil, 2019)

Two documents having similar words will have similar vectors, where the similarity between vectors can be computed using a standard measure such as the dot product. However, these techniques cannot segment the polysemic words which are having multiple meanings.
Hence in high dimensional space where each document represents one dimension, the resultant vector of the that particular term will be a vector sum of the term's occurrence in the dimensions corresponding to all the documents in which that particular word occurs.
The research (Kim and Gil, 2019) shows that the topic modelling techniques which helps to extracts representative keywords and latent topics from the abstracts of all online and offline published papers, by Latent Dirichlet allocation (LDA) scheme. Then, the K-means clustering algorithm is applied to classify the whole papers into research papers with similar subjects, based on tf-idf value of each paper.

2.3.2 Co-occurrence Matrix

In this matrix column and the rows both represent a word. Thus, the co-occurrence matrix is also sometimes called the term-term matrix.
There are two ways of creating a co-occurrence matrix:

- **Using the occurrence context (e.g. a sentence):**

Each sentence is represented as a context. If two terms occur in the same context, they are said to have occurred in the same occurrence context.

- **Skip-grams (x-skip-n-grams):**

A sliding window will include the (x+n) words. This window will serve as the context now. Terms that co-occur within this context are said to have co-occurred.

2.3.3 Word Vectors

Traditional approaches such as one-hot encoding and BOW models, using dummy variables to represent the presence of a word in an observation, is useful for some ML tasks. However, it fails to capture latent contextual information about the documents.
This kind of encodings technique often provides sufficient baseline models for simple NLP tasks but fails the sophistication for more complex tasks, like machine-translation and speech-recognition etc. It does not capture the syntactic and semantic relationships across collections of words.
On the other hand, word vectors represent words as multidimensional continuous floating-point numbers. In this dimension space, the semantically similar words are mapped to the nearest points in geometric space.
The essence of representing words as vectors is that the mathematical operator can be applied to this. The numbers in the word vector represent the word's distributed weight across dimensions. In a simplified sense, each dimension represents a meaning and the word's numerical weight on that dimension captures the closeness of its association with that meaning. Thus, the semantics of the word are embedded across the dimensions of the vector.
The occurrence and co-occurrence matrices have large dimensions (equal to the size of the vocabulary V). However, working with such huge matrices makes them almost impractical to use.Word embeddings are a compressed, low dimensional version of the mammoth-sized occurrence and co-occurrence matrices.

2.4 Word Embeddings

Neural network techniques are widely applied to obtain high-quality distributed representations of words (i.e., word embeddings) to address text mining, information retrieval, and natural language processing tasks. Most recent efforts have proposed several

efficient methods to learn word embeddings from context such that they can encode both semantic and syntactic relationships between words (Cui et al., 2015).
Recent developments in ML have shown extra-ordinary results in different real-life applications. Due to the extensive usage of ML systems, it has become very important for research scientists to explore the explainable AI by analysing the interpretation of the data by the models.
However, the challenge is the curse of dimensionality of this huge amount of data that requires high processing capable tools to process and analyze.
For a more intuitive exploration process and visualization purpose of high-dimension data, a stand-alone web application has been launched.

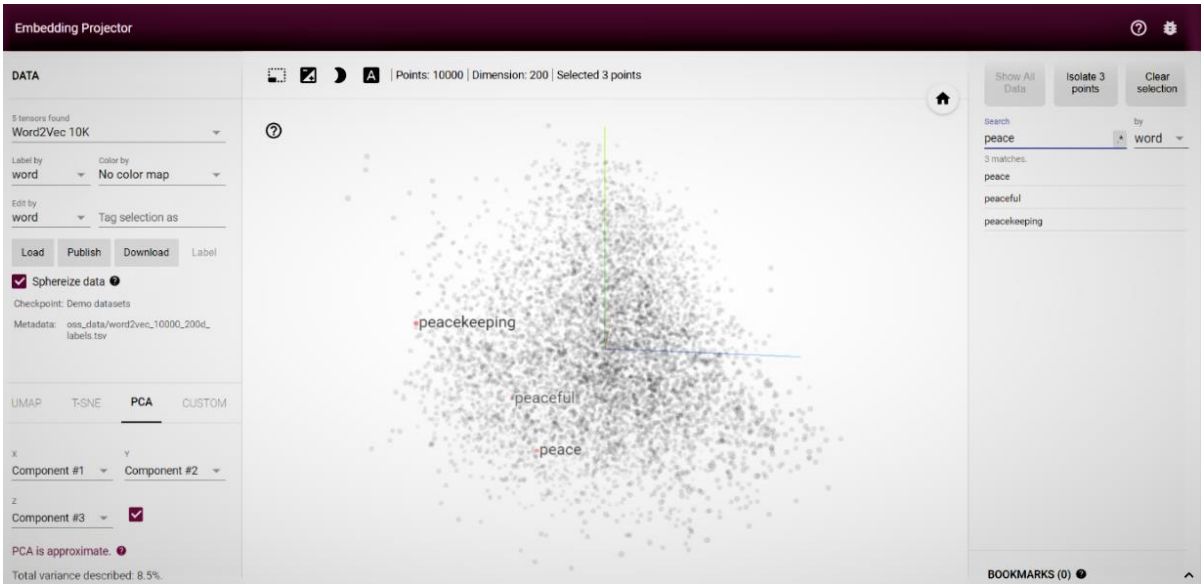


*Figure 2.3:Embedding Projector view of Word "Peace (*http://projector.tensorflow.org/*)(TensorFlow, 2020)*

Above is a figure showing the nearest points to the embedding for the word "Peace". Here a TensorFlow model is being trained using the word2vec.
The Embedding Projector follows three techniques to reduce the dimension of the data to visualize complex data: PCA, t-SNE, and custom linear projections.
PCA explores the internal structure of the embeddings and shows the most instrumental dimension of the dataset.
T-SNE, explores local neighbourhoods and finds clusters preserving the underlying meaning.
Custom linear projections help to discover meaningful directions in data sets - such as the distinction between an official and unofficial tone in any generative language model. (Smilkov and Group, 2016).

2.4.1 Popular Word Embeddings algorithms

In this section we will be discussing about some popular word embedding techniques:

- **Skip-gram Model (Prediction based):**

The Skip-gram model is a prediction-based approach for creating word embeddings.
In the skip-gram approach, the input is the target word and the task of the neural network is to predict the context words (the output) for that target word. The input word is represented in the form of a one-hot-encoded vector. Once trained, the weight matrix between the input layer and the hidden layer gives the words embeddings for any target word (in the vocabulary).

❖ **Latent Semantic Analysis (LSA) (Frequency-based):**

Latent Semantic Analysis (LSA) uses Singular Value Decomposition (SVD) to reduce the dimensionality of the matrix. It is a frequency-based approach. In LSA, it take a noisy higher-dimensional vector of a word and project it onto a lower-dimensional space. The lower-dimensional space is a much richer representation of the semantics of the words. Apart from its many advantages, LSA has some drawbacks as well. One is that the resulting dimensions are not interpretable (the typical disadvantage of any matrix factorization-based technique such as PCA). Also, LSA cannot deal with polysemy issues.

❖ **Word2Vec Model (Prediction based):**

Word2vec, a word embedding technique, has gained significant interest among researchers in natural language processing (NLP) in recent years. The embedding of the word vectors helps to identify a list of words that are used in similar contexts with respect to a given word. There are two aspects of word2vec techniques: CBOW and Skip-Gram. The model takes all the words from the corpus for each sentence and use current words to predict the neighboring word (skip-gram) or use the context to predict the current word (CBOW).

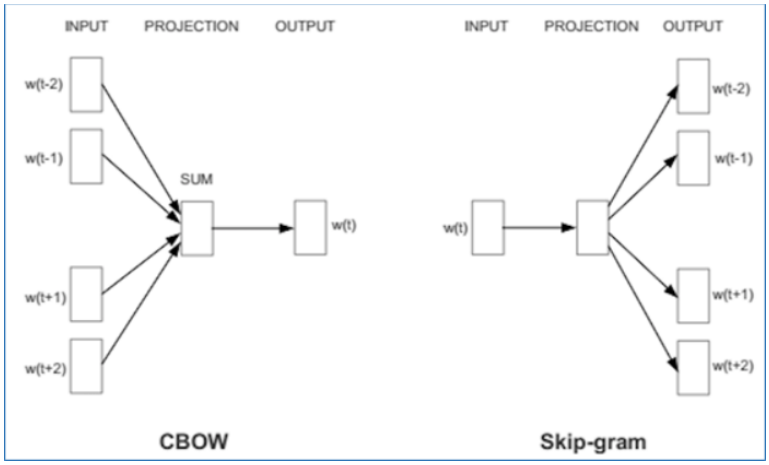


Figure 2.4: Word2Vec Models: CBOW and Skip-Gram (Mao, 2019)

Word2vec is like an autoencoder which trains words against other neighboring words in the input corpus.

2.5 RNN Based Word Embedding algorithm

Deep learning and neural networks are increasingly becoming popular and are being used in several applications. Recurrent Neural Networks (RNN) as proposed in (Rumelhart et al., 2019) are a branch of neural networks that allow previous output to be used as inputs while maintaining hidden states. They can use their memory (internal states) to process variable-length sequence of inputs making them useful for language (text) related tasks where the length of sentences is variable.

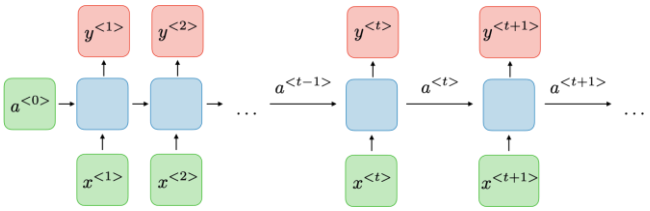


Figure 2.5: Architecture of traditional RNN (Stanford.edu, 2019)

2.5.1 LSTM (Long Short Term Memory)

RNN's might not be always suitable because of exploding or vanishing gradients problem which makes it difficult to learn long term dependencies. To overcome this RNN with gates and backpropagation known as Long Short-Term Memory was proposed by (Hochreiter and Urgen Schmidhuber, 1997). LSTM unit has four main parts cell, input gate, forget gate, and output gate, wherein the cell can remember values over a time interval, and the gates are used to regulate information flow.

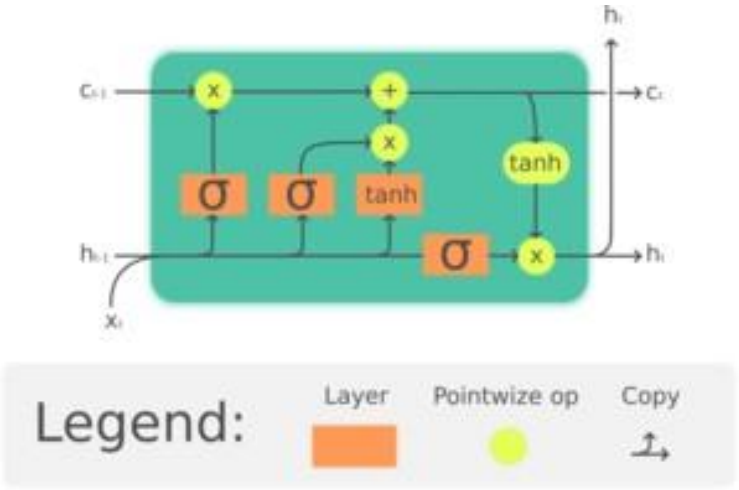


Figure 2.6: Architecture of Long Short Term Memory (Encyclopedia, n.d.)

Gated recurrent units (GRU) proposed by Kyunghyun Cho (Bahdanau et al., 2015)were proposed which was similar to LSTM but had fewer parameters and hence was faster. These were specifically suited for smaller datasets but were not as good as LSTMs. The main trade-off between using LSTM vs GRU is the speed of training and execution.

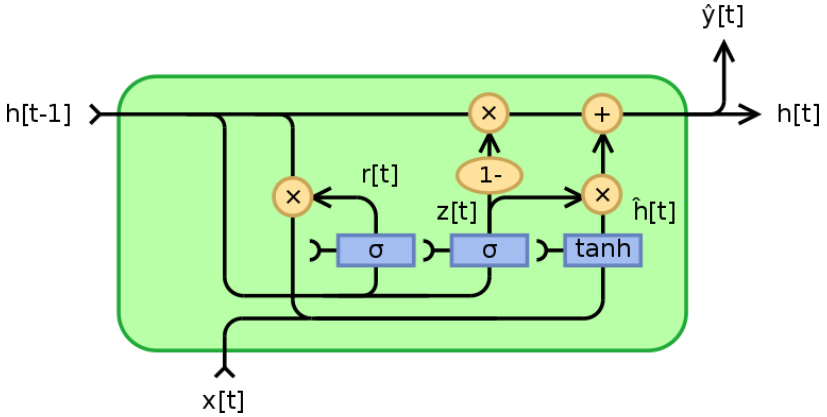


Figure 2.7: Architecture of Gated Recurrent Unit(Wikipedia, n.d.)

Application of LSTM in the context of Topic modelling:
In the below mentioned four models, the input is a BOW representation x(t)of the documents at time point t, which is then converted to a latent document embedding z(t) by either a pre-trained LDA model (as in LSTM+LDA) or an Encoder network (E).

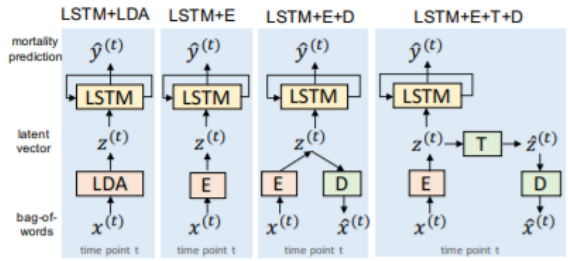


Figure 2.8: Application of LSTM in the context of Topic modelling(Jo et al., 2017)

In the fourth model, before the Decoder (D) converts zˆ(t) to a BOW vector xˆ(t) , an intermediate Transcoder (T) is added which maps the latent vector z(t)to a sparse topic vector zˆ(t)(Jo et al., 2017).

2.5.2 Attention and Mask Transformer based techniques

The latest advancements in neural networks for translation, lead to using attention mechanisms. Attention mechanism (Bahdanau et al., 2015) helps focus the network on some specific parts rather than entire sentences which are important to find relevant information from bigger pieces of text which is a key differentiator as compared to LSTM and other RNN's.

Attention networks are usually of two types: soft and hard attention. Soft attention is a deterministic algorithm that breaks the corpus into different subregions/parts and analyses all of them whereas hard attention is a stochastic sampling model making the accuracy dependent on the sampling.

The fundamental pillars of BERT are Transformers and Self Attention Mechanism.

The architecture of BERT depends on Google Transformers. It is basically a bidirectional neural network model.

BERT is developed by the bidirectional training of the Transformer for language modelling. A novel technique is integrated in BERT named Masked Language Modelling to train the model bi-directionally which was not possible previously.

Bert behaves as a Super Encoder for various sequences to sequence model.

Bert uses the below-mentioned approach for pre-training (Transfer Learning):

- Data: It is trained on BookCorpus(800M words) and English Wikipedia (2500M words). A huge document-level corpus to have a long continuous sequence is needed.
- Basic two Tasks: Predicting a word and Next sentence Prediction
- Time required for pre-training: Around 3 days on 16 TPU.

Google Brain tried to implement two type of BERT:

BERT_base:L=12,H=768,A=12

parameters: 110M

Bert_large: L=24,H=1024,A=16

Parameters=340M

L: Number of Encoder Layers

H: Hidden size ("Embedding Dim")

A: Number of Self-attention head

Transformers: Transformer is developed with an idea called attention mechanisms. However, it has not dispensed the recurrence and convolutions entirely. It shows extremely high performance in different machine translation tasks mainly because of its parallel processing techniques which significantly reduced the training time as well (Vaswani et al., 2017).RNN, LSTM (Hochreiter and Urgen Schmidhuber, 1997) and gated RNN (Burk, 1999) neural networks are an established and one of the most popular state of the art approaches in

sequence to sequence learning and problems like language modelling and machine translation (Burk, 1999).
Attention mechanisms plays a pivotal role in different tasks such as sequence modelling and transduction models. It permits modelling of dependencies regardless of their separation in the input or output sequences (Kim et al., 2017).

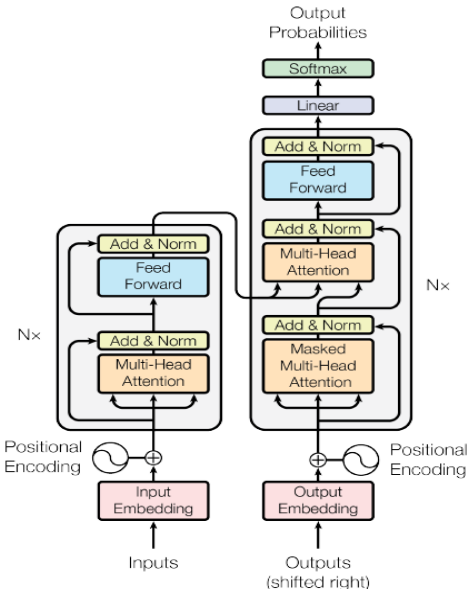


Figure 2.9: The Transformer - model architecture(Vaswani et al., 2017)

Initially, the sequence-to-sequence model was developed using RNN with gated LSTM and it was improved by the attention mechanism. The attention mechanism has added global behaviour to the decoding phase. However, in sequential processing, RNN loses information for long sentence. Hence Google's Transformer only uses the attention mechanism to encode and decode sequence and comfortably gets rid of the RNNS. The input of the encoder is the whole sequence or the sentence that is important to keep the Global processing aspect.
We can see "Attention" as a memory of the network which preserves the hidden states of the model, and the model fetches that information from the memory to get the contextual information.

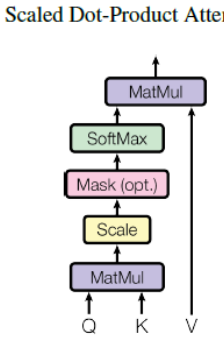


Figure 2.10: Attention Dot-Product (Vaswani et al., 2017)

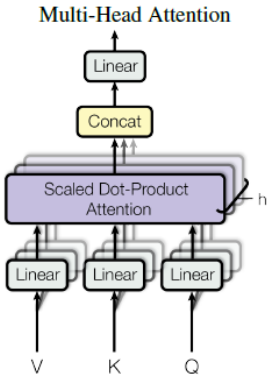


Figure 2.11: Multi-Head Attention (Vaswani et al., 2017)

In three different ways multi-head attention can be used by the Transformer:
1. The queries from the previous decoder layer, and the memory keys and values from the output of the encoder goes into the encoder-decoder attention.
2. All the keys, values and queries from the same place come to the self-attention layer of the encoder.

3.The decoder has a self-attention layer. It allows each position to attend to all end to end positions in the decoder including itself.
The Transformer deliberately avoid using of recurrence and instead rely entirely on an attention mechanism to identify dependencies between input and output. The Transformer permits significant parallelization in the entire process (Vaswani et al., 2017).

2.6 Traditional Topic Modelling techniques

In this section the research will focus on discussing about the traditional HBN based Topic modelling techniques.

2. 6.1 PLSA

PLSA is an advanced development on the top of LSA to overcome few disadvantages (Alghamdi and Alfalqi, 2015).The main objective of PLSA is to identify contexts of used words in the document without referring any dictionary.

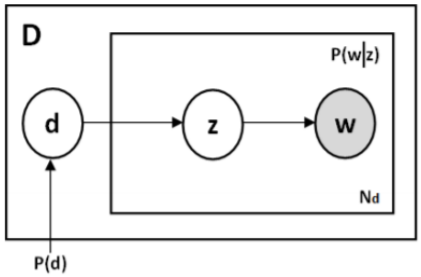


Figure 2.12: Plate Notation of PLSA(Alghamdi and Alfalqi, 2015)

1) Probability of selecting a document di, is P(di)
2) zk, is the latent class. The probability for a given document P(zk|di)
3) wj is the word that will be generated for a given latent class. The probability will P(wj |zk)
PLSA disambiguate the polysemy .It uncovers similar topics .It basically groups words together that share a common context (Hofmann, 2001)

2.6.2 LDA

Latent Dirichlet Allocation is a combination of pLSA and the Bayesian techniques. it uses Dirichlet Prior Process for the distribution of document-topic and word-topic for better generalization.

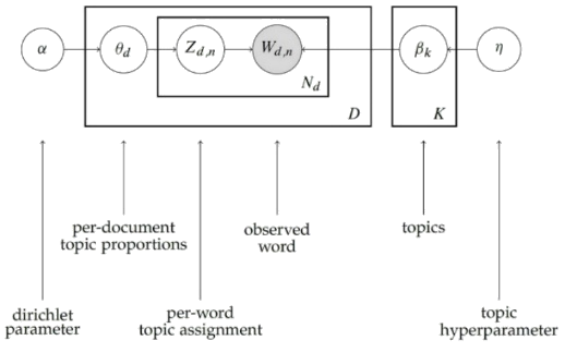


Figure 2.13: Plate Notation of LDA(Blei and Lafferty, 2009)

N = Number of words, M = Number of documents, θ = Topic selecting dice
Z = Selected Topic,W = Selected word by topic's dice
K = Number of topics,
α = K dimensional vector that defines how K topics are distributed across documents
V = Number of vocabulary words
β= V-dimensional vector that defines how V words are associated with topics

LDA hardly considers the sequence of the words in the document. Usually, LDA uses the traditional BOW techniques. When the topic modelling task becomes more complex it is very important to identify the underlying meaning of the text at every level (word, paragraph, document).
Typically, something like word2vec is been used for the vector representation of the words. lda2vec is a combination of word2vec and LDA which has been built using skip-gram model to generate topic, document and word vectors.

2.7 Gap in the existing research

Topic modelling using different hierarchical Bayesian techniques like LDA, LSA, PLSA and techniques like Non-Negative matrix factorization (NMF) of course, a good start. However, it takes a huge effort to tune the hyperparameter to create a meaningful topic.
On the other hand, Pre-trained transformer-based model like BERT, GPT2 or current development, GPT3 have shown amazing results in various NLP tasks over the last few years.
Over the years, researchers have used multiple algorithms to identify the latent topic of a corpus of document. But in most of the real-life problems, like mentioned in this study the documents /text are incoherent, unstructured, and also confusing.
Situation like analysing customer review, the texts are highly contextual and can be on different topics. Hence the algorithms based on one-hot-encoding cannot capture the context and the semantic relationship.
Development of Doc2Vec, where the resulting document and word are jointly embedded in the same space, which allows document embeddings to be represented by nearby word embeddings, leveraged the implementation process of topic modelling.
However, BERT's raw word embeddings capture useful and separable information about a word in terms of other words in BERT's vocabulary.
The information can be fetched from both raw embeddings and their transformed versions after they pass through BERT with a Masked language model (MLM) head. Bert/ELMO (dynamic word embedding) considers the context and for each token.
On Amazon's digital product's customer review, where the data is highly sparse, it is difficult to get useful information from them.
Hence, in this study, a new technique will be used. It will embed the full content of the sentence, which can later be clustered with similar topics by giving adequate weightage to both the word occurrence-based vector and BERT embedded vector to balance the relative importance of information from each source.
This study will open up a new thought process to implement a pre-trained language model to solve traditional NLP oriented business problem.

2.8 Summary

This section started with an introduction to Topic Modelling and its technique-wise development over the years. The section focuses on the Distributional Semantic Models of NLP and various processes to implement it, which includes the different word embedding techniques. Deep Learning models like LSTM has shown impressive performance in many of the NLP tasks. Attention and transformer-based models introduced first in 2017-18 are now the State-of-the-Art models for language modelling. There is also some advanced development like GPT3, which is trained on a very large number of parameters. Then we have discussed the traditional Topic modelling techniques like LDA, PLSA. Finally, the existing literature gaps are identified, which will be addressed in this study.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

In the recent past, the transition from research output to products carried few constraints other than technical advancement. It has initiated a kind of confusion between research and design thinking. However, nowadays the author focuses on the validation of the proposition of engineering and engineering research. The importance of this idea lies in the 'totality of design'. There are numerous of products emerging from such research. The author inspects the direction and re-direction of engineering research emerging from rigorous product design interaction and triggered by this process (Pugh, 1989).
There are different types of research methodologies:
Exploratory: Exploratory research is undertaken when few or no previous studies exist. The aim is to look for patterns, hypotheses, or ideas that can be tested and will form the basis of further research.
Descriptive: Descriptive research can be used to identify and classify the elements or characteristics of the subject.
Analytical: Analytical research often extends the Descriptive approach to suggest or explain why or how something is happening.
Predictive: The aim of Predictive research is to speculate intelligently on future possibilities, based on close analysis of available evidence of cause and effect.
The research approach can in the form of Quantitative or Qualitative, Applied or Basic, Deductive and Inductive. However, many research projects combine different approaches (Neville, 2007).

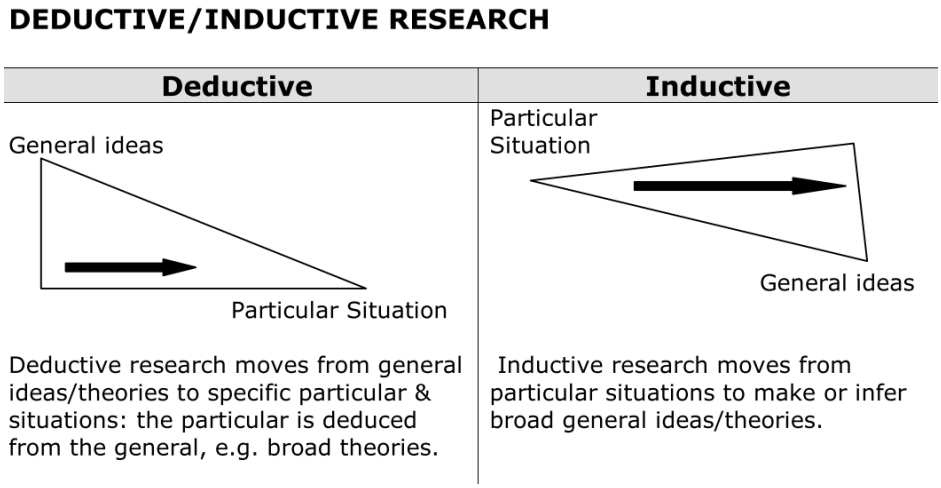


Figure3.1: Deductive and Inductive research(Neville, 2007)

The primary aim of this research work is to improve knowledge in general, keeping a particular applied purpose in mind. Then subsequently focus on the applied research from the start to apply its findings to a particular situation.
This work and the methodologies will depict both the deductive and Inductive nature of the research which bridge the application and the modern state of the art model.
The broad objective of this research work is to do a scientific study to solve a practical problem and unlock the potential of digital content, the amazon customer review, and empower and enhance online engagement between customers and different stakeholders. Hence it can be categorized as Applied Research.

It focuses on the study of AI explainability of contextual Topic Modelling to identify groups that are semantically similar and assign the most appropriate category tags of the reviews incorporating embedding techniques with traditional Topic modelling technique, LDA. Therefore, we can categorize it as Qualitative Research as well.
The entire study is based on the combination of Basic and Qualitative applied research, where the basic research retains a core position within the research mindset (Bentley et al., 2015) and qualitative applied research focuses on solving the real business problem.

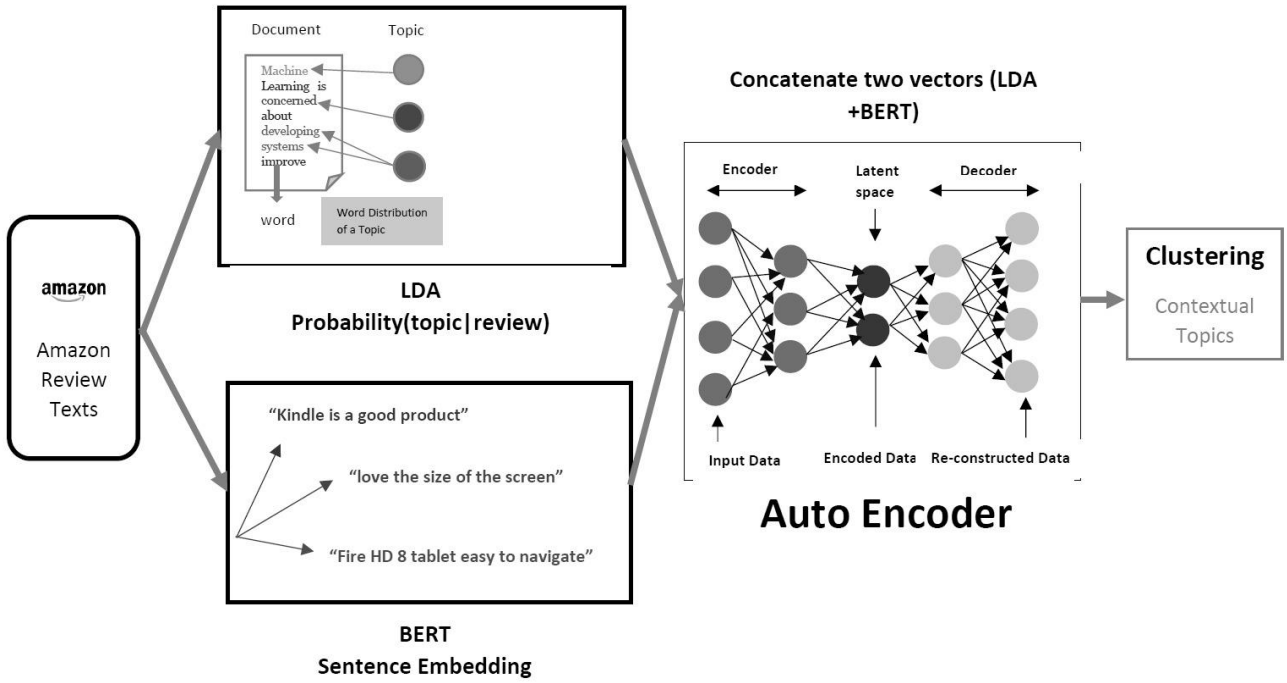


Figure 3.2: Contextual Topic Identification model design

Subsequent subsections will discuss all steps that will be required to address the goals mentioned above. The diagram in Figure 3.2 depicts the planned sequence of activities in the modelling and explanation generation phase.

3.2 Dataset Description

Datafiniti transforms unstructured data from web to usable format and gives instant access to it. Their master repository consists of data from thousands of websites which leveraged a standardized database of different consumer product, business strategy and property information (Datafiniti(Kaggle), n.d.).
The dataset used for this study is a list of more than 30,000 consumer reviews of different Amazon digital products like the Home Theatre, mobile, computer, media player, etc (Datafiniti(Kaggle), n.d.).The study will be utilizing this data of customer reviews of Amazon's electronics product and will discover the contextual topic of their discussion.

Data Dictionary			
#	**Field Name**	**Field Type**	**Field Description**
2.	Name	Text	The product's name.
3.	Brand	Text	The brand of the Product
4.	Category	Text	Category key of products from multiple sources.
6	manufacturer	Text	The manufacturer of this product.
7	reviews.date	Date	Date of the customer reviews

10	Reviews. doRecommend	Boolean	If recommended by the reviewer
12	reviews.Helpful_num	Integer	The number of visitor found this review helpful
13	Review_rating	Float	1 to 5 (5 is the best).
14	Source_URLs	Keyword	URLs where this reviews were seen.
15	Reviews_text	Text	text of the review or comment
16	Reviews_header	Text	Title of the review.
17	reviews_user_City	Text	City of the reviewer's
19	reviews.user_name	Text	Username of reviewer

Table 3.1: Amazon Digital product review dataset(Datafiniti(Kaggle), n.d.)

The review data could be valuable for various reasons like:

- Understand trends: to understand what people are talking about, things they like or things they do not like about.
- Improve your products from users feedbacks.
- To follow up with your user about the product that they don't like and further to understand the problem.
- To decrease return rate, re-stocking fees is one of the big expenses for e-commerce to succeed or even stay alive.

Here in this topic modelling or customer temporal behaviour analysis problem we will be focussing on the review text field and will try to find the latent topic of their discussion.

3.3 Dataset Pre-processing methods

The essence of Text mining lies on extracting the useful information from the textual data. It discovers latent knowledge from raw texts. The common two terms for this process are knowledge Discovery in Textual Databases (KDT) and Text Data Mining (TDM) (Kannan et al., 2015a).

The data considered in this study is basically is the reviews of customers,
which contains missing values, url, stop words, incorrect spellings, different phonetic ambiguity and languages as well. This data needs to be cleaned and normalized properly to proceed further to the transformation and model building phase.

Here in this study we focused on the traditional initial level of data extraction and sentence level normalization. Then we pass it through the language detector test.

Once we get the exact language, we do a spelling check using "Edit Distance" algorithm. Thereafter, there is a thorough word level preprocessing which includes the stop words removal, filtering out the punctuation,NER, correcting incorrect type ,phonetic ambiguity ,stemming and TF/IDF.

3.3.1 Extraction - Sentence level Pre-processing:

This method is used to tokenize the file content into individual sentence and word levels and convert them into lower case with some basic normalization, like handling missing delimiters, repetition of letters, handling informal parenthesis, and phrase repetition.

- Most of the embeddings and classifier algorithms have been made to deal with and analyse clean text, hence, it's of utmost importance to handle these text cleaning related issues to generate a good model.
- Replace commonly used short forms or slangs like won't, with, will not, or ll with will as they are not recognised and processed properly.
- We add space after the punctuations as its correct way of writing to add space usually after almost all the punctuations and help distinguish new sentences and words.

- For numbers two ways will be explored: we can either keep them as it is and make sure that the numbers are written in the usual form, for eg: 2k17 should be replaced with 2017 or we can simply replace the numbers with some character (say #) and check which of the two gives better results.
- If after step 1 we are still left with some extra spaces, then we need to make sure to remove that appropriately
- Missing values in the text can be replaced by *na* or random string

3.3.2 Language Detection

Amazon's Digital product reviews come from different parts of the globe which are basically in different languages. Every language has its own grammatical structure. This study has included three other languages (, 'French','Spanish' and 'Chinese') along with English.

3.3.3 Spell Check – Edit distance

Here we have used the spell corrector using the concept of edit distance. An edit distance is the number of edits that are needed to convert a source string to a target string. There are three edit operations allowed in the Levenshtein edit distance:

1. Insertion of a letter
2. Deletion of a letter
3. Substitution of a letter with another letter

Along with the above three edit operations, we can also do a transpose operation where we can swap two adjacent letters. This operation is allowed in the Damerau-Levenshtein edit distance.

3.3.4 Word Level Pre-processing:

Word pre-processing is the most important part of any text mining process associated with NLP. These various text pre-processing steps are widely used for dimensionality reduction. In the vector space model, each word/term is an axis/dimension. The text/document is represented as a vector in the multi-dimensional space. The number of unique words mean the number of dimensions.

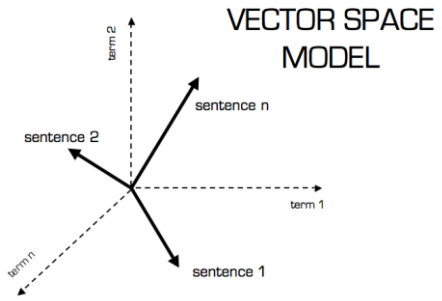


Figure 3.3: Term-sentence distribution in vector space

In this study, we have followed different word pre-processing steps like stop words removal, stemming, Name entity recognition, punctuation filtering, handling error in typing, phonetic hashing, and built the baseline TF-IDF models as well.

Stop words are basically the noise of natural language processing. Stop words are never measured as keywords in text mining applications (Porter, 1980). Hence in the very initial stage it should be removed.

The information extraction method identifies keywords and relationships within the text by pattern matching of predefined sequences in the text. (Gupta and Lehal, 2009).

The purpose of stemming is to identify the root of the words (Neville, 2007).

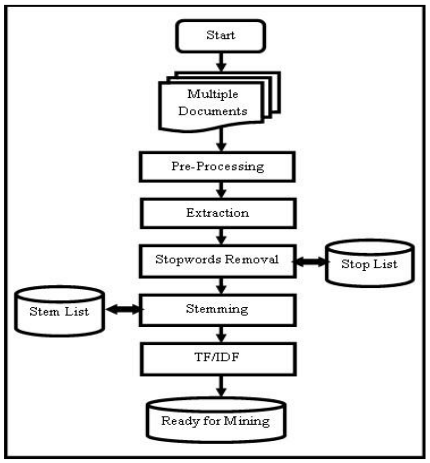


Figure 3.4: Pre-Processing Techniques(Kannan et al., 2015b)

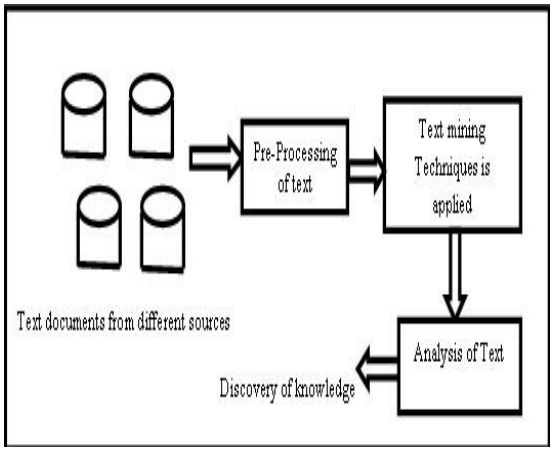


Figure 3.5: Text Mining Process (Kannan et al., 2015b)

There are certain words that have different pronunciations in different languages.
As a result, they end up being spelled differently. Examples of such words include names of people, city names, names of dishes, etc.
Performing stemming or lemmatization to these words will not help us as much because the problem of redundant tokens will still be present.
Hence, we need to reduce all the variations of a particular word to a common word using Soundex algorithm. It will reduce all the words to a four-digit code. All the words that have the same Soundex can then be mapped to a common word.

Algorithm

For a given word, apply the following steps:

1. Retain the first letter of the word as is
2. Replace consonants with digits according to the following:
   - b, f, p, v ⇒ 1
   - c, g, j, k, q, s, x, z⇒ 2
   - d, t ⇒ 3
   - l ⇒ 4
   - m, n ⇒ 5
   - r ⇒ 6
   - h, w, y ⇒ unencoded
3. Remove all the vowels and the unencoded letters. And then merge any consecutive same numbers into a single number
4. Continue till the code has one letter and three numbers. If the length of Code is greater than four then truncate it to four letters. In case the code is shorter than four letters, pad it with zeroes

Figure 3.6: Algorithm for the Soundex

Apart from the above-mentioned methods we did punctuation filtering and typo correction and prepared data for embedding transformation.

3.4 Transformation and Model Building

Once we complete the data pre-processing, we need to represent words as vectors to fed to our ML model as input.

3.4.1 Transformation (Word Embedding):

There are an estimated 13 million words in the English language. However, many of these words are related. Hence, we don't need to separate each vector. We must search for an N-dimensional vector space, which is much less than 13 million words.

That vector space will be sufficient to encode all semantics in our language. We can have a sense of similarity and difference between words using different concepts of vectors and distances between them (Euclidean, Cosine etc.)
Here in this study, we will be using the prediction-based approach with the combination of transformer-based BERT as a sentence embedding layer with an attention mechanism that learns contextual relations between words in a text.

3.4.2 Model Building:

- In this research, the entire model building pipeline has been divided into **four modules:**
- **LDA:** Topic Modelling using three-level **hierarchical Bayesian** model
- **Transformer based BERT:** Sentence embedding layer in the raw text
- **Autoencoder:** Dimension reduction (To surface the concatenated vector to the lower dimension)
- **K means clustering:** Topic clustering and visualization

LDA: Latent Dirichlet Allocation is a probabilistic topic model that considers documents as a bag-of-words. However, lda2vec builds document representations on top of word embeddings.
Bag-of-words: Text documents are represented in traditional NLP as a bag-of-words.
In BOW model each document is represented as a fixed-length vector with length equal to the vocabulary size.
If there is lots of data then only BOW gives good results on topic classification.

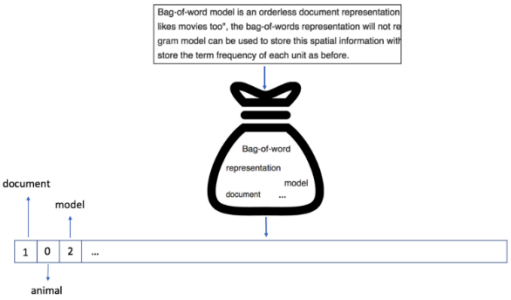


Figure 3.7: idea of BOW models

The LDA training starts with a collection of documents and each of these is represented by a fixed-length vector (BOW). Training an LDA model on N documents with M topics corresponds with finding the document and the latest topic that best explains the context of the text. Here once we pass the raw data through the LDA algorithm, we will get LDA vector.
Transformer based BERT: The LDA can suffer from the same disadvantages as the BOW since it predicts words inside of that document and disregard any structure or how these words interact on a local level. Hence it cannot capture the contextual semantic information properly.
Learning algorithms become more powerful, often at the cost of increased complexity. In response, the demand for algorithms to be transparent is growing. In NLP tasks, attention distributions learned by attention-based deep learning models are used to gain insights in the models' behavior.
In Transformer-like architecture, the distribution is calculated by different attention heads. It improves the transparency in the task of abstractive summarization (Neville, 2007).
The attention models have been inspired by human attention which enables us to identify certain important phrases or areas which are more important in a given corpus or data based on our ingrained learning and memory.

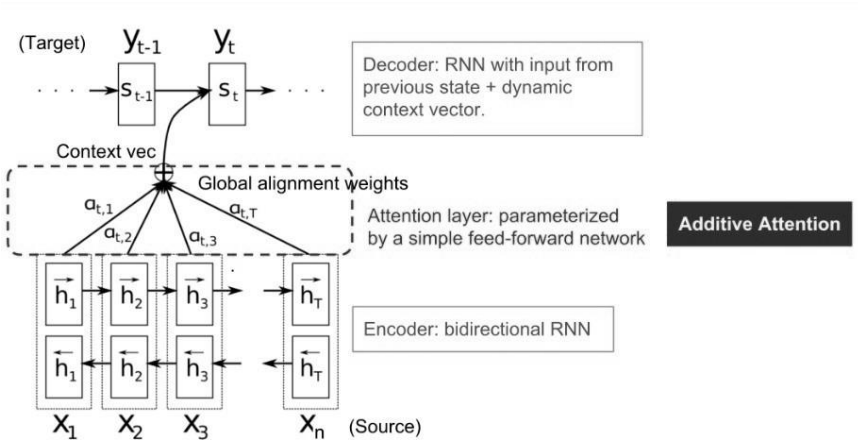


Figure 3.8: The encoder-decoder model with additive attention mechanism(Bahdanau et al., 2015)

BERT perform in the topic modelling task in two steps in its framework. It first trains on unlabelled data for the pre-training phase followed by initializing the model with pre-trained parameters which are then fine-tuned using labelled data from the downstream tasks. Unlike RNNs or LSTMs based sequence modelling the Transformer based models use their attention mechanisms to learn global dependencies between input and output.

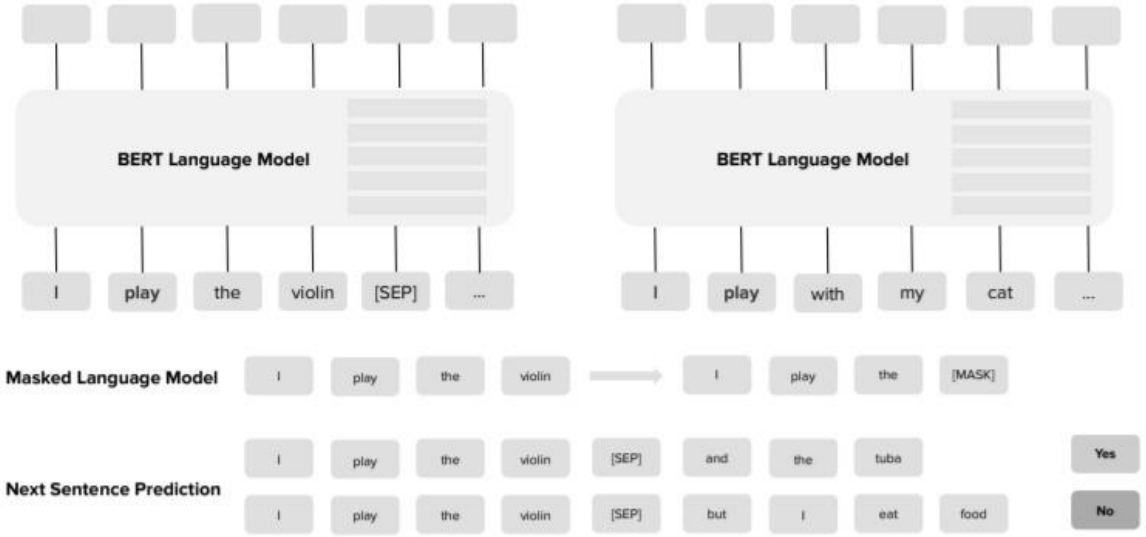


Figure 3.9: A schematic representation of BERT, masked language model and next sentence prediction(Nozza et al., 2020)

Two main components of the BERT pretraining process are the masked language model (MLM) and the next sentence prediction. The latter process is about predicting how likely one sentence is to follow another one in text (Nozza et al., 2020). Once we have the another embedded vector from BERT, we will concatenate this vector with the LDA vector.

Auto Encoder: The Auto Encoder plays a pivotal role in the study.

Auto Encoder is developed on the top of the Neural Network architecture to learn the surfaced lower-dimensional features of the input data.

Bourlard and Kamp (Bourlard and Kamp, 1988) discussed the relationship between auto-association by multi-layer perceptrons and SVD in 1988.

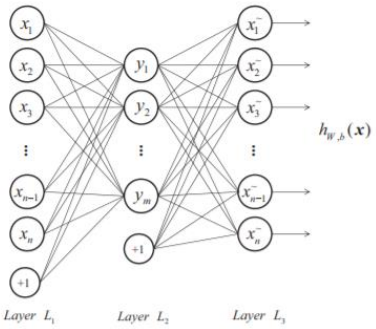


Figure 3.10: The Structure of Auto-Encoder

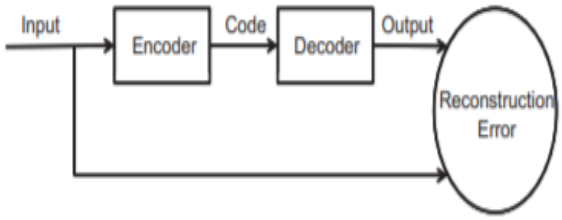


Figure 3.11:The visualization description ofAE (Wang et al., 2015)

Autoencoders consists of 4 parts:
Encoder: The model reduces the input dimension by compressing the input data into an encoded representation.
Bottleneck: It contains the compressed form of data with the lowest possible dimension
Decoder: It takes the compressed data as input and reconstruct it to match the input data as much as it is possible.
Reconstruction Loss: This method evaluates the performance of the decoder.

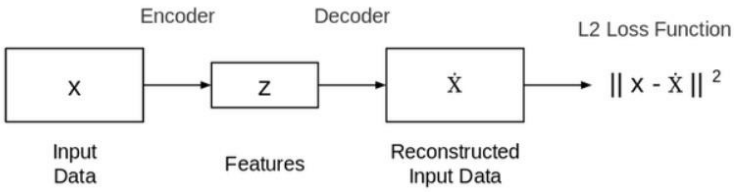


Figure 3.12: Auto Encoder Network (Le and Mikolov, 2015)

This network is trained to reconstruct the original input by using the features.
The L2 loss function indicates the difference between the original input and the decoded output data.
Dimensionality reduction is a dynamic research topic(Wang et al., 2015). It uses the projection method to maps the data from high feature space to low feature space.
Dimensionality reduction methods can be linear and nonlinear(Ghodsi, 2006).
In the topic modelling tasks, the input of the encoder is the whole sequence or the sentence that is important to keep the Global processing aspect.

- An autoencoder will be used to surface the concatenated vector to the lower dimension (Latent Space Representation).
- Once we have the lower dimension representation with more condensed information, clustering techniques will be applied to get the context-based topic.

In the process of dimensionality reduction can discard some dimensions. It will lead to the loss of information. Here the main focus is to keep the important characteristics of the original data as much as possible. (Wang et al., 2015).
Clustering: Document clustering incorporated in topic modelling is highly beneficial. Effective Document clustering can be facilitated by the projection of `documents in the topic space(Xie and Xing, 2013).

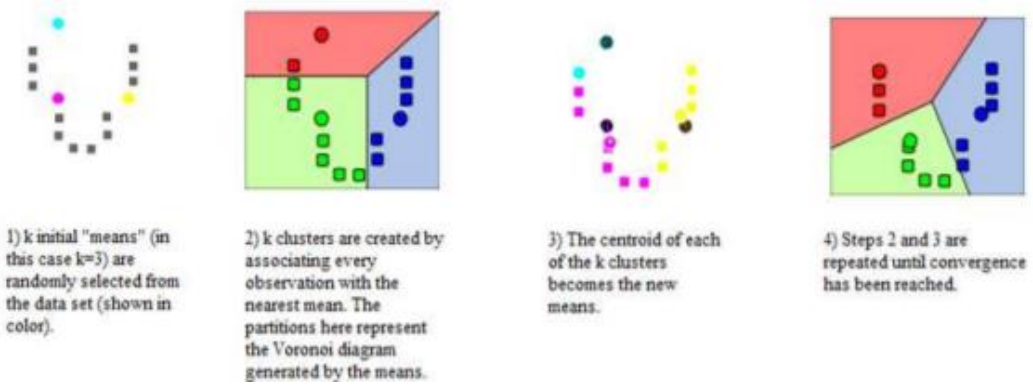


Figure 3.13: Demonstration of k-means clustering (Wikipedia, n.d.)

Implementation of K-means in large data set is very simple(Karandikar and Finin, 2010). K-means is linear in iterations, number of clusters, number of vectors and dimensionality of the space. In terms of efficiency k-means is much better than the hierarchical algorithms (Manning et al., 2009).
Clustering of document enables us to extract local topics specific to each document cluster and global topics shared across clusters.
In this study, we have performed these two tasks simultaneously. First, we have used topic models to project documents into a topic space, then perform clustering algorithms such as K-means in the topic space to obtain clusters.

3.5 Expected result and initial evaluation

The entire research work can be segmented into three major parts.
The first part is the identification of the latent topics of the review which will be done by hierarchical Bayesian Model, LDA. The study will be using the most popular Gensim coherence metrics like c_v and u_mass to evaluate the model.
Coherence Value is based on a sliding-window technique, a one-set segmentation of the words and a validation measure that uses NPMI and the similarity function (Cosine).
The plot of the Coherence score vs the number of topic graphs can give some important insight.
The cv and umass give the coherence score which gives an idea of the interpretability of the topics.
Next, the study will focus on the comparison of the results of the topic modelling using LDA with the different methods, like a combination of TF-IDF, clustering and models like a combination of BERT, LDA and Clustering.
Thereafter we will be using the Silhouette score in our final model to measure the consistency within the cluster. The technique provides a succinct graphical representation of how well each object has been classified.
The study will include visualizing the different clusters depicting the main topics, what most of the customers have concerns about. The resultant performance improvement in the topic identification task by incorporating BERT will be the most probable expected result.
As a part of this research, initial level of implementation has been done on a randomly sampled dataset from the master dataset. For the sample data, below is the clustering result on the vectors from contextual topic embedding, which shows that the clusters are balanced and quite separated.

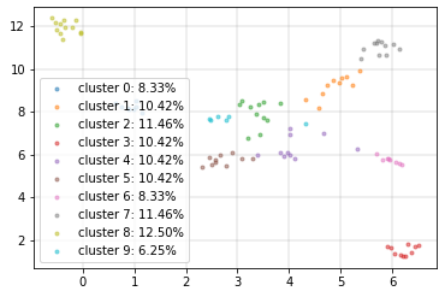


Figure 3.14: Clustering result on vectors from contextual topic embedding (2D UMAP)

3.6 Summary: Implementation Pipeline:

This section summarizes the implementation pipeline and the process flow.
Below are the summarized steps of this research methodology:

- Carefully pre-process the fetched raw data (Customer reviews of Amazon's digital product) before feeding it into the model using different patterns-search filters for normalization.
- Analyse the distribution of the data and improve the data pre-processing to have a more normalized design.
- For identifying the main topics in the customer reviews, we can follow two types of model:
  1. Bayesian models (LDA)
  2. Embed documents into vector to identify their similarities in the vector space by clustering.
- The Amazon digital product review data are highly sparsed and it does not coherently discuss one single topic. Hence it will be difficult for BOW based model like LDA to identify the underlying context-based topic of the reviews.
- We will be applying parallel embedding techniques to embed the full contents which to be clustered with similar topics.
  1. TF-IDF Vector representation of the document.TF-IDF is a BOW based (disregarding grammar and word order). Hence mostly it won't be able to capture the contextual information because of the incoherent and unstructured data.
  2. BERT: Use BERT as a Sentence Embedding layer on the document to capture the contextual information.
- Prepare two vectors:
  1. Probabilistic topic assignment vector by LDA: Use the LDA model to figure out topics by identifying the occurrence of frequent words in the incoherent text.
  2. Sentence embedding vector by BERT: When the used words and the meaning of the sentence in the reviews are incoherent, extra semantic information is required to find the topic of the texts more precisely.
- Combine both the LDA and BERT vector to identify the semantic information and contextual information by tuning relevant hyperparameters to get the optimum information from different sources.
- The resultant vector will be containing highly sparsed information in the high-dimensional space.
- Finally, an autoencoder will reduce the dimensions of the concatenated vector, and the K- means clustering techniques will be applied to visualize the homogeneous topics.

CHAPTER 4: ANALYSIS AND IMPLEMENTATION

4.1 Introduction

This section will discuss the different experiments performed as part of the study. It will cover the detailed steps in data preparation. Thereafter, an extensive exploratory data analysis has been done on the Amazon digital products' review data. Later this section will focus on the different Bayesian Belief Network and BERT language model for topic modelling.
The final segment explains the detailed implementation of each component (LDA, BERT, AutoEncoder, clustering) in the state of the art model pipeline to generate contextual topics from the incoherent text data.

4.2 Data Preparation

The dataset contains 34,660 rows and 21 columns which includes consumer review text of different Amazon digital products along with so many columns having the key and unique identification details of product and customer. But in this study, we only need information such as product name, brand, categories, review text, user recommendation (binary), reviews ratings, and the number of people that found a review helpful. Therefore, most of the redundant columns have been dropped, and reduced the dataset to only six columns.

Data Dictionary			
#	Field Name	Field Type	Field Description
1	product_name	Text	The product's name.
2	brand	Text	The product's brand name.
3	categories	Text	Categories of the Digital products
4	recommendation_indicator	Boolean	If recommended by the reviewer
5	reviews_rating	Integer	Rating of the products by the reviewer
6	reviews_text	Text	text of the review or comment

Table 4.1: Extracted features for the study

Data pre-processing is the most important step in any NLP pipeline. As the dataset is about the review text of different digital products, submitted by humans, they are bound to be highly incoherent and uncleaned.
After cleaning the entire dataset, exploratory data analysis ties all of this together and generate insights, checking assumptions, and revealing underlying hidden patterns in the data.
The main pre-processing steps followed in this study are as below:

1. Language detector: There are many words in English that are biased towards the French. Hence, here a language detector has been used to detect the English language. Here we have used the Spacy-Language detector module
2. Cleaning misspelled words: The review texts are entered manually by the customer. Hence, it's full of misspelled words. Here modules like SymSpell, Verbosity have been used from the package symspellpy.
   Below are the parameters, used in the symspell module:
   pkg_resources = "frequency_dictionary_en_82_765.txt"
   Dictionary edit distance = 3,
   Length of the prefix = 7
   Apart from this, the cleansing is also done by using phonetics algorithm.

We have used the fuzzy module of the python library for implementing common phonetic algorithms quickly. Typically, this is in string similarity exercises.
Below are few examples, we have got in our data set:

Soundex		
Correct Spelling	Phonetically matched spelling	Soundex Code
Amazon	Amajon	A525
Kindle	Kindel	K534
Tab	Tb	T100
resolution	resoluton	R243
clarity	klarity	C463 & K463
pixcel	pxel	P240
memory	memry	M560

Table 4.2: Sample output of the Soundex algorithm

The Soundex algorithm worked perfectly when the initial letter is the same for both the spelling.

1. **Normalize Unicode** – Universal character encoding standard normalization has been performed on the corpus. This is done to make sure the different representations of the same character are normalized and treated as the same. The specific algorithm used is NFKD (Normalization Form Compatibility Decomposition), wherein the characters are decomposed by compatibility.
2. **Expand Contractions:** Contractions are the shortened versions of words like “don’t” for “do not” and “how’ll” for “how will”. These are used to reduce the speaking and writing time of words. We need to expand these contractions for a better analysis of the reviews. Here, we have used a regular expression to find the contraction and a manually created common English contractions to replace it.

Contracted Words	Decontracted words
Won’t	Will not
Can’t	Can not
You’ll	You all
I’m	I am
‘ll	Will

Table 4.3: Sample expanded contractions of words

3. **Basic normalization:**
   Below basic normalizations have been done using Regex :
   normalization 1: Handling missing delimiter
   normalization 2: Convert all words to lower case
   normalization 3: Letter repetition (if more than 2)
   normalization 4: string * as delimiter
   normalization 5: Text in parenthesis, assumed to be less informal
   normalization 11: Cleaning the noisy text
   normalization 12: phrase repetition

4. **Filtering out punctuations and numbers:**
   Spaces were intentionally added before and after the punctuations as they do hold relevance while processing the text. Numbers were either replaced with some special symbol like ‘#’ or removed as they did not affect the overall corpus understanding.

5. **Typographical error correction:** It is done by using the symspell package and the words with no match up have been dropped.

4.3 Exploratory Data Analysis:

In this section, the study will focus on the detailed exploratory data analysis of the review text and the underlying contextual hidden layer that will also include the inclined sentiment of the topic of discussion and its polarity.
Here in the data set, we have total 34660 records and 21 features. The most important features for this research have minimal null values, and for Bayesian Belief Network and BERT model, we need to consider a reasonably large corpus of data. Hence the study will take the records of the entire dataset with selective columns.

Features Name	Count of Null values	Percentage of null values
Product Name	6760	20
brand	0	0
categories	0	0
recommendation_indicator	594	2
reviews_text	1	0
reviews_rating	33	0

Table 4.4: Percentage of null values of the features

Below are the different stages of EDA performed as a part of this study:

4.3.1 New features creation:

After cleaning the data set to analyze the length and word count of a generic review and sentiment polarity, below mentioned instrumental features have been created:
1. Variable to capture the length of the review.
2. Variable to capture the word count of the review.
3. To calculate sentiment polarity that lies in the range of [-1,1] where 1 means positive sentiment and -1 means a negative sentiment using TextBlob.

4.3.2 Sentiment polarity analysis:

The key aspect of sentiment analysis is to analyze a body of text for understanding the opinion expressed by it. Typically, it quantifies this sentiment with a positive or negative value, called polarity.
Below are the five random reviews with the highest positive and lowest negative sentiment polarity:

Reviews with highest Polarity
Bought this tablet for my 4 yo and 9 yo. Tablet was perfect for their use of watching movies and kid games
I purchased for my son, he said it is the best gift he has ever received
This thing is great! It connects to everything! Thank you Best Buy!
Awesome gadget plays music answers questions tells the time.
This product works perfectly for our family's needs.

Table 4.5: Reviews with highest polarity

Reviews with lowest Polarity
i got this tablet for traveling. Just games, reading, stuff like that so I didn't need anything fancy. It works but it lags on games and the battery life is horrible.
We are a Kindle family! Use them all the time! Reading books, checking email, watching Netflix. Would be very sad without it!!!
Works just as well as my Rokus, however the interface is completely awful. Buy the fire for kodi buy a roku for everything else.
One of the worst purchases or investments you could make for technology.
Bought it so my son would stop grabbing for my phone. Does what it needs to but battery life is horrible

Table 4.6: Reviews with lowest polarity

There are so many different amazon's digital products in the dataset and all have different average sentiment polarity. Below are the top five products with the highest polarity:

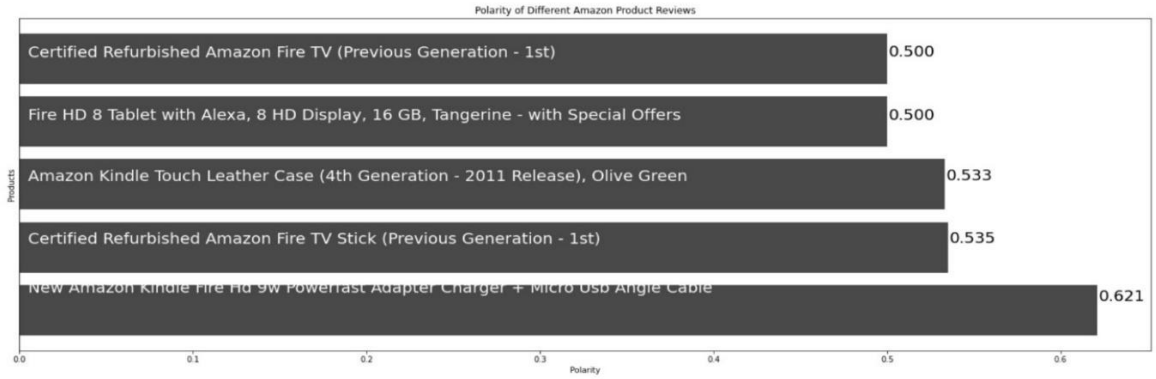


Figure 4.1: Polarity of different Amazon product's reviews

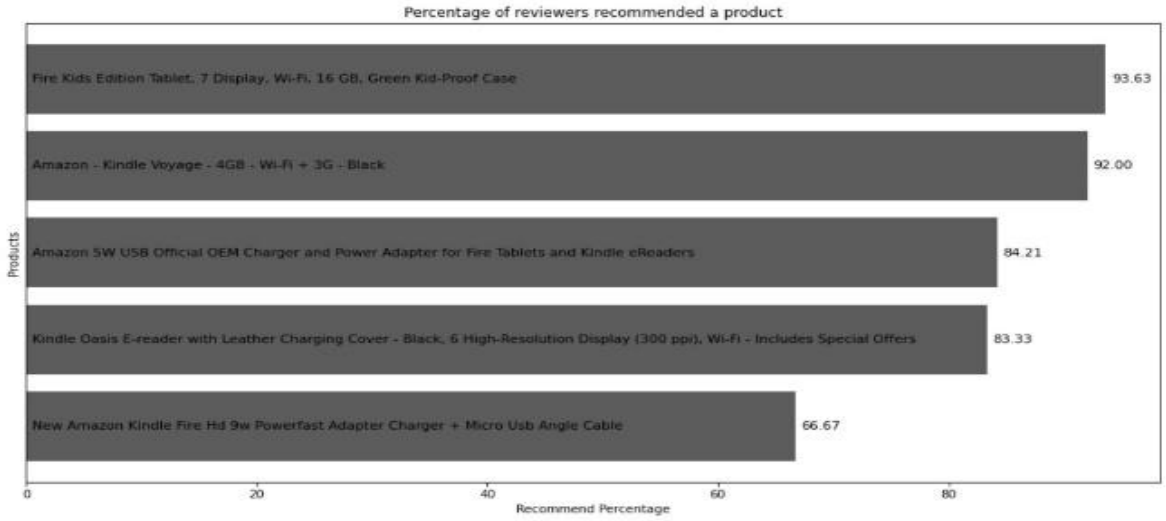


Figure 4.2: Percentage of reviewers recommended a product

Here, we can see that the new amazon HD Kindle has the lowest recommendation percentage. It's reviews also have the very less polarity. So, we can say that the polarity of reviews affects the chances of a product getting recommended.

4.3.3 Univariate and Bivariate Visualization with Plotly:

Single-variable or univariate visualization is the simplest type of visualization which consists of observations on only a single characteristic or attribute.

Here the Univariate and the Bivariate analysis mainly focused on two aspects of analysis that will be used to introspect and validate the algorithmic approach to identify the latent topic of the documents.

- Polarity and Ratings distribution – To understand the overall inclination and temporal behaviour of the reviewers.
- Different text level analysis of review texts

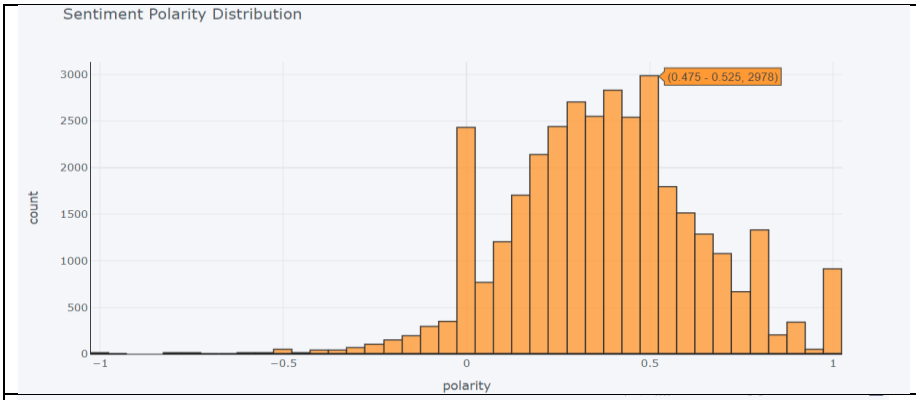
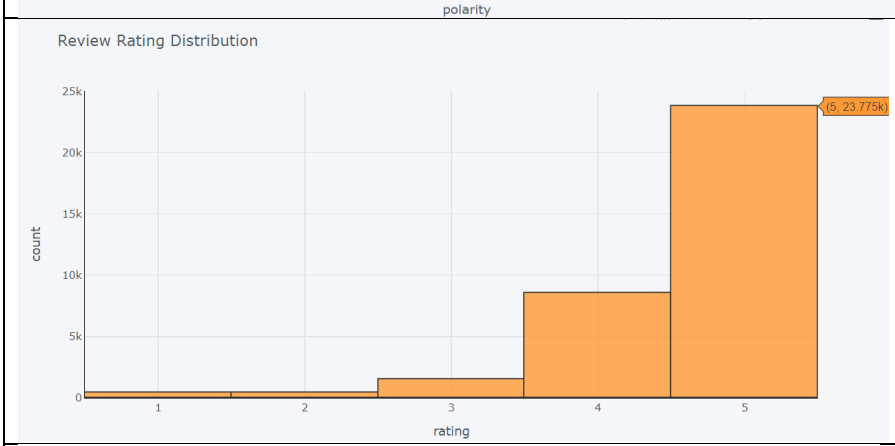
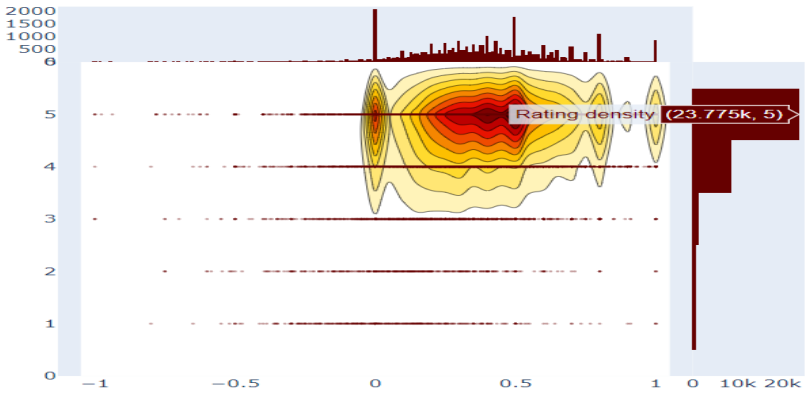
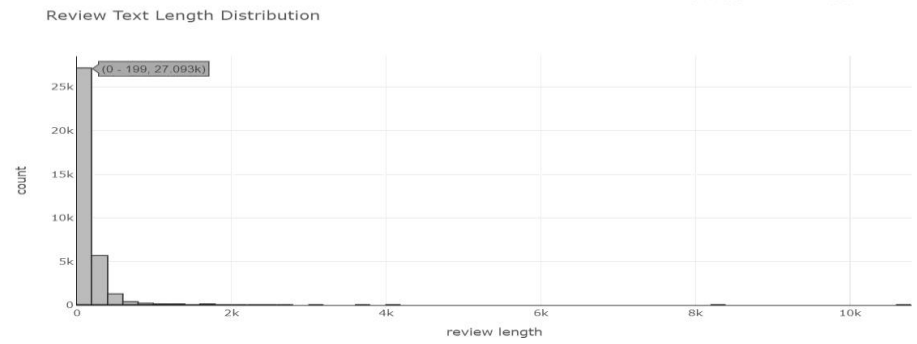
	Majority of the Sentiment-polarity scores are greater than zero, means most of them are pretty positive.
	The ratings are in align with the polarity score, that is, most of the ratings are pretty high at 4 or 5 ranges
	2D Density Jointplot of Sentiment Polarity vs. Rating

Figure 4.3: Polarity and Ratings distribution of the reviews

Different text level analysis of the reviews:

The purpose of this method is to understand the unstructured information and extract the meaningful numeric indices from the text. So that, the information contained in the text be accessible to the various NLP algorithms to derive summaries of the documents and to analyze the clusters of words of the documents.

	Here in this distribution, we can see that most of the reviews (approx. 27 k) text average text length lies between 0 to 200.

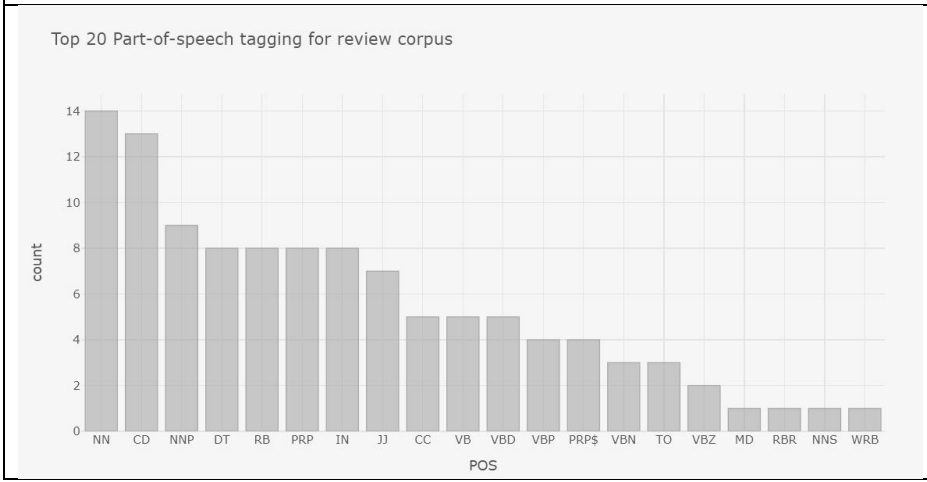
<table>
<tr>
<td> 
</td>
<td>The distribution of top unigrams after removing stop words</td>
</tr>
<tr>
<td> 
</td>
<td>POS tagging gives an overall idea of the distribution of different parts of speech used in the entire corpus of documents
which can leverage to more efficient Topic Modelling “Through a Noun Only Approach”.</td>
</tr>
</table>

Figure 4.4: Text level analysis (EDA)

The distribution of Top trigrams after removing stop words:

A trigram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a $(3-1) = 2$ order Markov model. Trigram models are now widely used in probability, communication theory, computational linguistics -statistical natural language processing).The main benefits of trigram models is, that it gives an idea of underlying closely clustered words (3 words for trigram) of a large amount of document, which can later be leveraged to an another way of validating the Topic modelling system.

Top trigrams of the review text	count
easy set easy use	53
bought tablet year old	38
year old son loves	34
great product great price	30
battery lasts long time	30
year old daughter loves	30
watch movies play games	27
play games watch movies	27
play games read books	26
great tablet great price	26
great product easy use	26
read books play games	26
tablet year old son	24
love tablet easy use	23
works great easy use	23
bought year old loves	22
bought year old son	21
play games watch videos	21
year old grandson loves	19
tablet year old daughter	18

Table 4.7: Top Trigrams of the review text

4.3.4 Word Cloud:

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is.

Also known as tag clouds or text clouds, these are ideal ways to pull out the most pertinent parts of textual data, from blog posts to databases. They can also help business users compare and contrast two different pieces of text to find the wording similarities between the two. Here as a part of EDA word cloud technique has been applied on the raw corpus of documents to analyse, what are the most frequent words are clustered together.

Figure 4.5: Word Cloud of the text of the reviews (EDA)

The above word cloud indicates the most of the reviewers love the amazon products like tablet, Kindle, and Alexa. They have used words like great, good and love most frequently with the product's name mentioned above.
The above-mentioned EDA approach is completely **frequentist**. Hence the performance of the word-clustering is expected to get higher precision in the later explained **Bayesian** and **Transformer** based approaches subsequently.

4.4 Model Building:

This section is broadly classified into three major segments with three different ways of implementation approaches for contextual Topic modelling from a highly incoherent corpus of documents.

1. Hierarchical Bayesian Belief Network (HBN)
2. BERT based language model
3. The combination of HBN and BERT

4.4.1 Hierarchical Bayesian Belief Network:

- **Model motivation**

Bayesian inference is a method by which we can calculate the probability of an event based on some common assumptions and the outcomes of previous related events. It also allows us to use new observations to improve the model, by going through many iterations where a prior probability is updated with observational evidence to produce a new and improved posterior probability. In this way the more iterations we run, the more effective our model becomes.

In topic modelling as it relates to text documents, the goal is to infer the words related to a given topic and the topics being discussed in a given document, based on analysis of a set of documents we've already observed, i.e set of documents a "corpus". We also want our topic models to improve as they continue observing new documents. In LDA it is Bayesian inference which makes these goals possible.

- **Implementation of Latent Dirichlet Allocation:**

Here in this study for a faster implementation of LDA (parallelized for multicore machines), we have used the gensim model. In the data pre-processing steps we have transformed the data to train the LDA model including the corpus and the dictionary. Apart from that, alpha and eta are hyperparameters that affect sparsity of the topics. According to the Gensim docs, both defaults to 1.0/number of topics prior.

Two most important parameter of this model are: Dictionary and the corpus.

Below is the example of snippet created as a part of this study:

DICTIONORY		CORPUS		
Parameter name	Sample	Parameter name	Input: Lemmatized data	Output: corpus as the model parameter
id2word	{0: 'ability', 1: 'child', 2: 'content', 3: 'control', 4: 'disappointed', 5: 'ease', 6: 'far', 7: 'love', 8: 'monitor', 9: 'product'}	corpus (It has been created from the pre-processed lemmatized data)	['product', 'far', 'disappointed', 'child', 'love', 'ability', 'monitor', 'control', 'content', 'see', 'ease']	[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (10, 1)]]

Table 4.8: sample dictionary and the corpus for LDA

Model parameter:

Function	Parameter name	value
Number of documents to be used in each training chunk	chunksize	100
how often the model parameters should be updated	update_every	1
total number of training passes	passes	10
If True, the model also computes a list of topics, sorted in descending order of most likely topics for each word, along with their phi values multiplied by the feature length	per_word_topics	TRUE

Table 4.9: Model parameter for base model (LDA)

Once the baseline coherence score for the default LDA model is achieved, the study performs a series of sensitivity tests to determine the following model hyperparameters:

- Number of Topics (K)
- Dirichlet hyperparameter alpha: Document-Topic Density

- Dirichlet hyperparameter eta: Word-Topic Density

Here the test has been taken place in sequence, one parameter at a time by keeping others constant and run them over the two different validation corpus sets.
Here,c_v is used as the metric for the performance comparison.

Validation_Set	Topics	Alpha	eta	Coherence
100% Corpus	8	0.61	0.01	0.662706731
100% Corpus	6	0.61	0.31	0.658955638
100% Corpus	7	0.61	0.61	0.657480747
100% Corpus	9	0.61	symmetric	0.655653166
100% Corpus	9	0.61	0.01	0.655508018
100% Corpus	8	0.61	symmetric	0.655090744
100% Corpus	6	0.61	0.91	0.6539911
100% Corpus	5	0.91	symmetric	0.650763062
100% Corpus	8	0.61	0.31	0.650440079
100% Corpus	10	0.61	symmetric	0.64997502
100% Corpus	7	0.61	0.31	0.649868391
100% Corpus	7	0.61	0.91	0.648391177
100% Corpus	10	0.61	0.01	0.646971708

Validation_Set	Topics	Alpha	eta	Coherence
75% Corpus	10	0.61	symmetric	0.646809165
75% Corpus	9	0.61	symmetric	0.637289413
75% Corpus	7	0.61	symmetric	0.635720044
75% Corpus	7	0.61	0.01	0.634935214
75% Corpus	10	0.61	0.01	0.634618935
75% Corpus	8	0.61	symmetric	0.630415143
75% Corpus	9	0.61	0.01	0.627436947
75% Corpus	9	0.91	0.31	0.625676486
75% Corpus	8	0.61	0.31	0.624057731
75% Corpus	6	0.91	0.91	0.623728092
75% Corpus	6	0.61	0.61	0.621565759
75% Corpus	10	0.61	0.31	0.621418366
75% Corpus	6	0.91	0.61	0.621099711

Table 4.10: Comparison of coherence score for different alpha and eta (LDA)

From the above result, the values of hyperparameter alpha and eta, that yielded maximum C_v score, have been considered for the final model for the number of topic(K)=8

- **LDA Mallet Model:**

In the previous model we had used Gensim's inbuilt version of the LDA algorithm. Mallet's version, however, often gives a better quality of topics.Both are two completely independent implementations of Latent Dirichlet Allocation. gensim.models.LdaModel is the single-core version of LDA implemented in gensim. There is also parallelized LDA version available in gensim (gensim.models.ldamulticore).
Both Gensim implementations use an online variational Bayes (VB) algorithm for Latent Dirichlet Allocation.
Most of the parameters, e.g., the number of topics, alpha and eta are shared between both algorithms because both implement LDA.

mallet_path : Path to the mallet binary	*/home/usernamemallet-2.0.8/bin/mallet*
corpus	Collection of texts in BoW format
id2word	Mapping between tokens ids and words from corpus, if not specified - will be inferred from *corpus*
num_topics	20

Table 4.11: Mallet LDA parameters

- **Implementation of Latent semantic analysis:**

Using LDA machine would not be able to capture this concept as it cannot understand the context in which the words have been used because mapping words to documents won't really help. This is where Latent Semantic Analysis (LSA) comes into play as it attempts to leverage the context around the words to capture the hidden concepts, also known as topics. LSA is one such technique that can find these hidden topics.
For m number of text documents with n number of total unique terms (words) the number of desired topics, k, has to be specified as input.
It first generates a document-term matrix of shape m x n having TF-IDF scores and then reduce the dimensions of the above-mentioned matrix to k (no. of desired topics) dimensions, using singular-value decomposition (SVD).

SVD decomposes a matrix into three other matrices. It will be decomposed into matrix U, matrix S, and VT (transpose of matrix V). Each row of the matrix Uk (document-term matrix) is the vector representation of the corresponding document. The length of these vectors is k, which is the number of desired topics. Vector representation for the terms in the data can be found in the matrix Vk (term-topic matrix).
So, SVD gives vectors for every document and term in our data. The length of each vector would be k which can be used to find similar words and similar documents using the cosine similarity method.
Document-Term-Matrix (scipy.sparse.csr.csr_matrix) has been created using tfidf_vectorizer with below parameters :

Stop words	English
use_idf	True
smooth_idf	True

Table 4.12: Parameters of DTM

Below is the sample Document term matrix for document number one:

Document : 1
(0, 4030) 0.32724465147251425
(0, 2899) 0.31585734211779815
(0, 2926) 0.2856614831323374
(0, 7918) 0.4427791920701659
(0, 422) 0.3276055449625041
(0, 7215) 0.1921343732725319
(0, 12972)0.15127133867306986
(0, 7392) 0.15903873589694864
(0, 2404) 0.31023704498593746
(0, 3647) 0.34751928708257596
(0, 4648) 0.26252614321717016
(0, 9421) 0.1927049010433224

There after the LSA model with Single value decomposition (SVD) method has been applied with below mentioned parameters:

Desired number of topic (SVD n_components)	8
algorithm	randomized
n_iter(iteration)	100
random_state1	122

Table 4.13 Parameters for LSA model

The components of SVD model are the desired topics,

Topic Number	**Top Three words of each topic**
Topic 1	great tablet use
Topic 2	easy use set
Topic 3	loves bought gift
Topic 4	easy tablet use
Topic 5	great works product
Topic 6	kindle love great
Topic 7	love kids echo
Topic 8	product good recommend

Table 4.14: LSA model output

4.4.2 BERT based language model

- **Model motivation**

The main three reasons why BERT is likely to be playing the most significant role in NLP are:

- ❖ It’s bidirectional
- ❖ It combines Mask Language Model (MLM) and Next Sentence Prediction (NSP).
- ❖ So far, it’s the best method in NLP to understand context-heavy texts

- **Model implementation (RoBERTa and DistilBERT)**

In this segment the research mainly focuses on topic modelling framework BERTopic using BERT as an embedding layer.
BERTopic leverages transformers and c-TF-IDF to create a dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions. It even supports visualizations similar to LDAvis(Maarten Grootendorst, 2020).
For BERTopic, two different types of embedding models have been used.

- Embedding model: " **xlm-r-100langs-bert-base-nli-stsb-mean-tokens"** with the below-mentioned embedded parameters:

Architecture : XLMRoBERTaModel	
attention_probs_dropout_prob	0.1
eos_token_id	2
gradient_checkpointing	false
hidden_act	"gelu"
hidden_dropout_prob	0.1
hidden_size	768
initializer_range	0.02
intermediate_size	3072
layer_norm_eps	1e-05
max_position_embeddings	514
model_type	"xlm-RoBERTa"
num_attention_heads	12
num_hidden_layers	12
output_past	true
vocab_size	250002

Table 4.15: XLMRoBERTa Model parameters

Apart from the above-mentioned model, the study also considered newly introduced DistilBERT. It is smaller, faster, cheaper and lighter compared to most of the recent development. It also gives a nice balance between speed and performance.

Architectures : DistilBERTModel	
Activation : "gelu"	
attention_dropout	0.1
dim	768
dropout	0.1
hidden_dim	3072
initializer_range	0.02
max_position_embeddings	512
model_type	"DistilBERT"
n_heads	12
n_layers	6
pad_token_id	0
qa_dropout	0.1
seq_classif_dropout	0.2

sinusoidal_pos_embds	False
tie_weights_	true
vocab_size	30522

Table 4.16: DistilBERT Model parameters

DistilBERT makes it possible to reduce the size of a generic BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster.
To leverage the inductive biases learned by larger models during pretraining a triple loss combining language modelling, distillation and cosine-distance losses has also been introduced to improve the quality of the product.

4.4.3 The combination of HBN and BERT

Here in this proposed method the entire model building and training have been divided in five major components:
1. Creating the probabilistic topic assignment vector (LDA)
2. Creating the sentence embedding vector (BERT)
3. concatenated above two vectors with the assigned weights for these two vectors (Hyperparameters tuning) and create a vector in the high dimensional space.
4. lower dimensional latent space representation of the concatenated vector using an Autoencoder
5. Implementation of K-means clustering to latent space representations to get the contextual Topics

Figure 3.2 shows the architecture of the contextual Topic Modelling state of the art model (combination of BERT and LDA).
Section mentioned below provides further details of the architecture, parameters and hyperparameters used in different components of the pipeline.

LDA Vector:

Corpus	Stream of document vectors or sparse matrix of shape
id2word	Mapping from word IDs to words. It is used to determine the vocabulary size
num_topics : 10	The number of requested latent topics to be extracted from the training corpus.
Alpha(**Hyperparameter)** : auto	Learns an asymmetric prior from the corpus
Passes : 20	Number of passes through the corpus during training.

Table 4.17: LDA parameters for combination model

BERT Vector:

Here in order to create the sentence embedding vector, the pretrained BERT model "bert-base-nli-max-tokens" has been used with the below mentioned parameters:

Parameter	Values
attention_probs_dropout_prob	0.1
gradient_checkpointing	false
hidden_act	"gelu"
hidden_dropout_prob	0.1
hidden_size	768
initializer_range	0.02
intermediate_size	3072
layer_norm_eps	1.00E-12

max_position_embeddings	512
model_type	"bert"
num_attention_heads	12
num_hidden_layers	12
type_vocab_size	2
vocab_size	30522

Table 4.18: Bert sentence embedder parameter for combination model

Hyper Parameter (Gamma) tuning: In the process pipeline, once the LDA vector and BERT embedded vector are generated, it has to be concatenated with adequate weightage to maximize the coherence score.
Below is the series of test for different values of gamma(The weight associated with the BERT vector) and the achieved coherence of the final model .

gamma	coherence	Silhouette
0.5	0.424	0.075
1	0.437	0.083
5	0.466	0.076
10	0.506	0.101
15	0.55	0.133
20	0.563	0.201
25	0.522	0.201
30	0.517	0.231
35	0.547	0.266

The value of the gamma has been chosen to achieve optimum coherence score with a balanced Silhouette score. Since the concatenated vector is in the higher dimensional space with sparse and corelated information, it is then passed through an auto encoder with below-mentioned parameter to learn a lower dimensional latent space representation of the concatenated vector.

Table 4.19: Coherence score and Silhouette score for different Gamma values

Auto Encoder Model Parameter

latent_dim	32
activation	relu
epochs	200
batch_size	128
shuffle	True
verbose	0

Table 4.20: Auto Encoder Model Parameter

Auto Encoder Compilation:

optimizer	adam
loss	mean_squared_error

Table 4.21: Auto Encoder Compilation

The output of the autoencoder is the concatenated vector in the lower rich dimension. It then passes through the K- Means clustering algorithm. It partitions of objects into k non-empty subsets and identify the cluster centroids (mean point) of the current partition.
Then it assigns each point to a specific cluster and compute the distances from each point and allot points to the cluster where the distance from the centroid is minimum and iteratively again find the centroid of the new cluster formed.

The K-means clustering used in the model implementation pipe line used below mentioned parameters:

Parameter	Values	Description
n_clusters	10	The number of clusters to form as well as the number of centroids to generate.
n_init	10	Number of times the k-means algorithm will be run with different centroid seeds.
max_iter	200	Maximum number of iterations of the k-means algorithm for a single run
tol	0.0005	Relative tolerance with regards to Frobenius norm of the difference in the cluster centers of two consecutive iterations to declare convergence

Table 4.22: K-means clustering model parameter

4.5 Required Resources

4.5.1 Hardware Requirements

For this study, the computation of the experiments will be performed on a Windows system with the below configurations:

- OS Name: Microsoft Windows 10 Home Single Language
- Version:10.0.18362 Build 18362
- Processor: Intel Core i5-8250U CPU
- Installed RAM :8.00 GB
- Total Virtual Memory:15.4 GB

4.5.2 Software Requirements

In this study, the ML and DL algorithm will be implemented in the Python framework.
The data pre-processing, model building, and fetching insights will be done by using several open-source libraries available in the Python framework.

- Programming Language: Python 3.7
  Open source libraries:
  - sentence-transformers: Sentence Embedding with BERT
  - spacy-langdetect - language detection capabilities
  - language-detector- language detection capabilities
  - gensim - Unsupervised topic modelling
  - symspellpy- Python port of SymSpell v6.5, which provides much higher speed and lower memory consumption
  - sklearn -machine learning library
  - NLTK - Natural Language Toolkit
  - Numpy and pandas - Data pre-processing, Mathematical function
  - Matplotlib (Seaborn) - Visualization

4.5.3 Cloud provider for building AI project (GPU/TPU):

- NimbleBox

4.6 Summary

The design and implementation of the study can be divided into three major stages.
In the first stage, the raw review data of amazon product's reviews are pre-processed to make it usable for EDA (Frequentist analysis), Bayesian and transfer learning (Transformer based) model. The data preparation section includes different Text mining and cleaning techniques, like dropping redundant features, detecting languages, handling incorrect spellings, normalization, filtering out punctuations and numbers etc.
In the EDA section, the research focuses on exploring the underlying contextual hidden layer of the corpus of documents that also includes the inclined sentiment of the topic of discussion and its polarity.
Later in the model building segment, the research has mentioned the motivation of each approach and the corresponding implementation techniques.
Here, three fundamental approaches have been shown to identify the latent topics.
In the HBN approach LDA, Mallet LDA with different inference algorithms and LSA models have been developed with essential hyperparameter tuning.
Finally, the transformer-based model implementation shows two ways of implementation, using the BERTopic framework (RoBERTa and DistilBERT) and with the combination of LDA and Bert-embedded vector transformation to find the latent topic.
The implementation of BERTopic using RoBERTa and the combination of HBN and embedded vector on the incoherent data, generated through digital platform are novel to this study and has shown a significant pathway of the future, to implement giant advanced language model on the daily NLP related problem.

CHAPTER 5: RESULTS AND DISCUSSIONS

This section will discuss the results and performance metrics generated from all the experiments outlined in the previous chapter. This section will first discuss the results of the different Hierarchical Bayesian techniques to find the latent topic of the document. Here in this study as a part of the Bayesian experiment, two algorithms have been used, Latent Dirichlet allocation (LDA) and latent semantic analysis (LSA). As the focus of the study is the exploration of the pre-trained language model, BERT to find the latent topic distribution, it will first discuss the interpretation of the BERTopic technique, that leverages transformers and c-TF-IDF with Bert base embedded model. Finally, the study will examine the results come from the state-of-the-art model, proposed in this research, the combination of LDA(HBN) and Bert sentence embedding vector with hyperparameter tuning.

5.1 Latent Dirichlet Allocation (LDA) model performance metrics

Model perplexity and topic coherence provide a convenient measure to judge how good a given topic model is.

Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topics. These measurements help to distinguish between the topics that are semantically interpretable, are artifacts of statistical inference. Here in this study, the c_v measure has been used.It is based on a sliding window, one-set segmentation of the top words, and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity.

On the other hand, in the information theory, perplexity is a measurement of how well a probability distribution or probability model predicts a sample. It is used to compare probability models. A low perplexity indicates the probability distribution is good at predicting the sample.

Considering the basic idea of LDA, the major challenge is finding the latent distribution by getting the normalization factor of the Bayesian Belief Network (Bayesian Inference problem). Below mentioned sections focus on the solutions of the Bayesian Inference problem in two different approaches.

5.1.1 Gensim (Using Variational Bayes sampling method) :

Here initially, the baseline LDA model has been developed with TF-IDF corpus using the Gensim package.

Gensim uses the variational Bayes sampling method to train the model.

Variational Inference methods find the best approximation of a distribution among a parametrized family. Specifically, the idea is to optimize over the parameters to obtain the closest element to the target with respect to a well-defined error measure.

The basic parameters of the model: learns an asymmetric prior from the corpus, alpha = “auto”, with topic size 20

The result found in the study:

Perplexity	Coherence Score
-13.38	0.45583310717988335

Sample distribution of 5 topics:

[(0, '0.257*"need" + 0.220*"make" + 0.068*"memory" + 0.060*"enough" + ' '0.053*"sure" + 0.051*"help" + 0.046*"may" + 0.045*"fact" + 0.042*"cheap" + ' '0.026*"thank"'),	(1, '0.260*"set" + 0.207*"content" + 0.110*"access" + 0.107*"allow" + ' '0.048*"kid" + 0.045*"spend" + 0.035*"instal" + 0.026*"travel" + ' '0.023*"plan" + 0.015*"limit"'),

(2, '0.537*"good" + 0.104*"money" + 0.096*"worth" + 0.089*"internet" + ' '0.089*"nice" + 0.048*"small" + 0.000*"surfing" + 0.000*"cable" + ' '0.000*"box" + 0.000*"stick"')	(3, '0.264*"great" + 0.179*"work" + 0.151*"well" + 0.092*"would" + ' '0.074*"recommend" + 0.045*"price" + 0.034*"look" + 0.033*"little" + ' '0.032*"say" + 0.020*"definitely"')
(4, '0.113*"way" + 0.112*"find" + 0.102*"far" + 0.086*"take" + 0.074*"control" + ' '0.062*"old" + 0.054*"never" + 0.053*"year" + 0.051*"perfect" + ' '0.047*"seem"')	(5, '0.236*"lot" + 0.117*"speed" + 0.093*"sound" + 0.086*"always" + 0.049*"hard" ' '+ 0.043*"speaker" + 0.042*"fine" + 0.042*"full" + 0.040*"reason" + ' '0.035*"simply"')

Table 5.1: Sample distribution of 5 topics using LDA

Visualization of the topic-keywords using pyLDAvis:

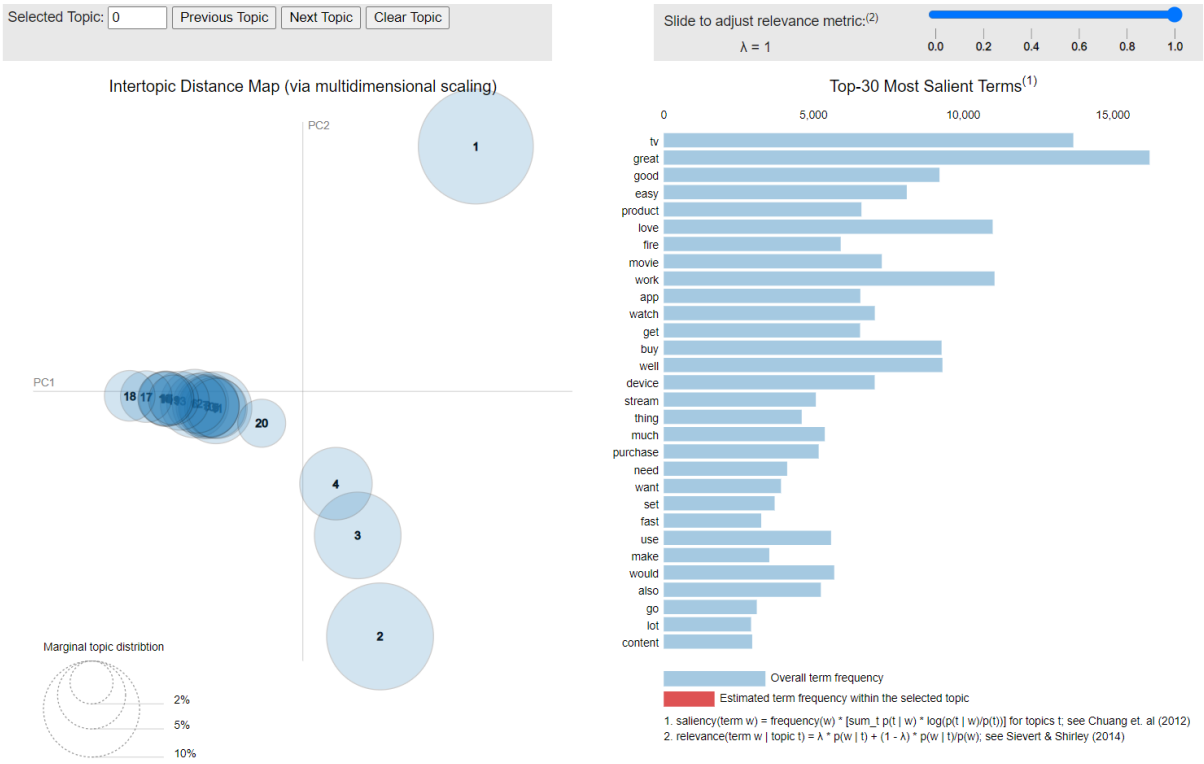


Figure 5.1: Visualization of the topic-keywords

However, to compare different model building techniques using TF-IDF corpus a manual test for the optimum coherence for different alpha and eta values has been performed. Here we have checked the different combination of the hyperparameter (alpha and eta) and fit the models with the below-mentioned parameter:

num_topics=10
random_state=100
chunksize=100
passes=10
alpha=0.01
eta=0.9

```
Topic: 0
Words: 0.059*"echo" + 0.045*"alexa" + 0.036*"music" + 0.030*"love" + 0.029*"great" + 0.028*"cabl" + 0.020*"home" + 0.018*"work"
+ 0.016*"play" + 0.016*"question"
Topic: 1
Words: 0.038*"work" + 0.034*"tablet" + 0.024*"perfect" + 0.020*"purchas" + 0.020*"firetv" + 0.017*"screen" + 0.016*"buy" + 0.01
6*"replac" + 0.015*"kindl" + 0.014*"want"
Topic: 2
Words: 0.071*"product" + 0.063*"great" + 0.044*"recommend" + 0.041*"easi" + 0.039*"purchas" + 0.037*"work" + 0.028*"love" + 0.0
21*"buy" + 0.020*"good" + 0.020*"best"
Topic: 3
Words: 0.045*"love" + 0.034*"book" + 0.031*"buy" + 0.026*"read" + 0.024*"year" + 0.021*"thing" + 0.019*"like" + 0.019*"play" +
0.016*"time" + 0.016*"list"
Topic: 4
Words: 0.080*"great" + 0.044*"tablet" + 0.036*"price" + 0.034*"good" + 0.032*"amazon" + 0.026*"qualiti" + 0.021*"sound" + 0.018
*"product" + 0.015*"speaker" + 0.015*"money"
Topic: 5
Words: 0.035*"batteri" + 0.026*"life" + 0.016*"good" + 0.016*"buy" + 0.016*"need" + 0.014*"long" + 0.014*"alexa" + 0.013*"song"
+ 0.013*"love" + 0.012*"like"
Topic: 6
Words: 0.027*"amazon" + 0.023*"devic" + 0.022*"stick" + 0.018*"like" + 0.016*"great" + 0.016*"work" + 0.016*"stream" + 0.014*"a
pp" + 0.011*"better" + 0.009*"weather"
Topic: 7
Words: 0.068*"read" + 0.062*"kindl" + 0.044*"light" + 0.032*"love" + 0.030*"book" + 0.029*"easi" + 0.022*"great" + 0.017*"scree
n" + 0.015*"reader" + 0.015*"paperwhit"
Topic: 8
Words: 0.052*"amazon" + 0.039*"prime" + 0.025*"easi" + 0.020*"remot" + 0.017*"movi" + 0.017*"need" + 0.016*"good" + 0.015*"grea
t" + 0.014*"stream" + 0.013*"music"
Topic: 9
Words: 0.102*"love" + 0.041*"great" + 0.038*"kid" + 0.031*"buy" + 0.029*"easi" + 0.028*"tablet" + 0.023*"watch" + 0.020*"movi"
+ 0.019*"game" + 0.019*"famili"
```

Figure 5.2: Sample LDA output for optimum alpha and eta values (using TF-IDF)

By doing an exhaustive coherence score maximization test using hyperparameter tuning, the optimum effective coherence score has been achieved = **0.65324**

5.1.2 LDA Mallet Model (Using Markov Chains Monte Carlo - Gibbs Sampling) :

The inference algorithm used in Mallet follows Markov Chains Monte Carlo (Gibbs Sampling). Markov Chain Monte Carlo algorithms are aimed at generating samples from a given probability distribution.

The “Monte Carlo” part of the method’s name is due to the sampling purpose and, the “Markov Chain” part comes from the way we obtain these samples.

Contrarily to VI methods, MCMC approaches assume no model for the studied probability distribution (the posterior in the Bayesian inference case).

As a consequence, these results are most of the time more costly to obtain but also more accurate than the one we can get from VI.

Gibbs sampling is one MCMC technique suitable for the task. The idea of Gibbs sampling is to generate posterior samples by sweeping through each variable (or block of variables) to sample from its conditional distribution with the remaining variables fixed to their current values.

Sample Output: Topic 15 and 11

(15, [('screen', 0.07484636798151163), ('kindle', 0.05998214191921845), ('turn', 0.03513840012605704), ('light', 0.03214454540679657), ('hand', 0.02542150323021167), ('page', 0.024738694259152268), ('model', 0.021534744471873524), ('paperwhite', 0.020799411733809548) ('feel', 0.020379221597772994), ('touch', 0.018173223383581072)])	(11, [('music', 0.1294033753268362), ('echo', 0.07178512003803185), ('alexa', 0.06745899690991206), ('play', 0.050677442357974806), ('question', 0.040836700736867126), ('weather', 0.03332541003090088), ('listen', 0.030758260042785833), ('answer', 0.026764915616829095), ('news', 0.021820774898977893), ('dot', 0.021250297123841216)]),

Table 5.2: Sample output of the Mallet LDA algorithm

`Coherence Score:0.5766526924705814`

Here, we can see that using the mallet class has increased the coherence score by **26.5 %,** since it uses the Gibbs inference algorithm.

It can be inferred:

- MCMC can be used in Bayesian inference to generate, directly from the “not normalized part” of the posterior, samples to work with instead of dealing with intractable computations.

- ❖ Variational Inference (VI) is a method for approximating distributions that uses an optimization process over parameters to find the best approximation among a given distribution.
- ❖ VI optimization process is not sensitive to the multiplicative constant in the target distribution. The method can be used to approximate a posterior only defined up to a normalization factor

Overall, we can infer that the sampling process of MCMC is heavy but has no bias, hence these methods are preferred when accurate results are expected, without considering the time it takes. Although the choice in the family of VI methods can introduce a bias,
it comes with a reasonable optimization process that makes these methods particularly adapted to a large-scale inference problem requiring fast computations.

5.2 Latent Semantic Allocation model performance metrics:

Latent semantic analysis is centred around computing a partial singular value decomposition (SVD) of the document term matrix (DTM). This decomposition reduces the text data into a manageable number of dimensions for analysis. Latent semantic analysis is equivalent to performing principal components analysis (PCA).
Here Top 3 words from the latent 10 topics is shown below:

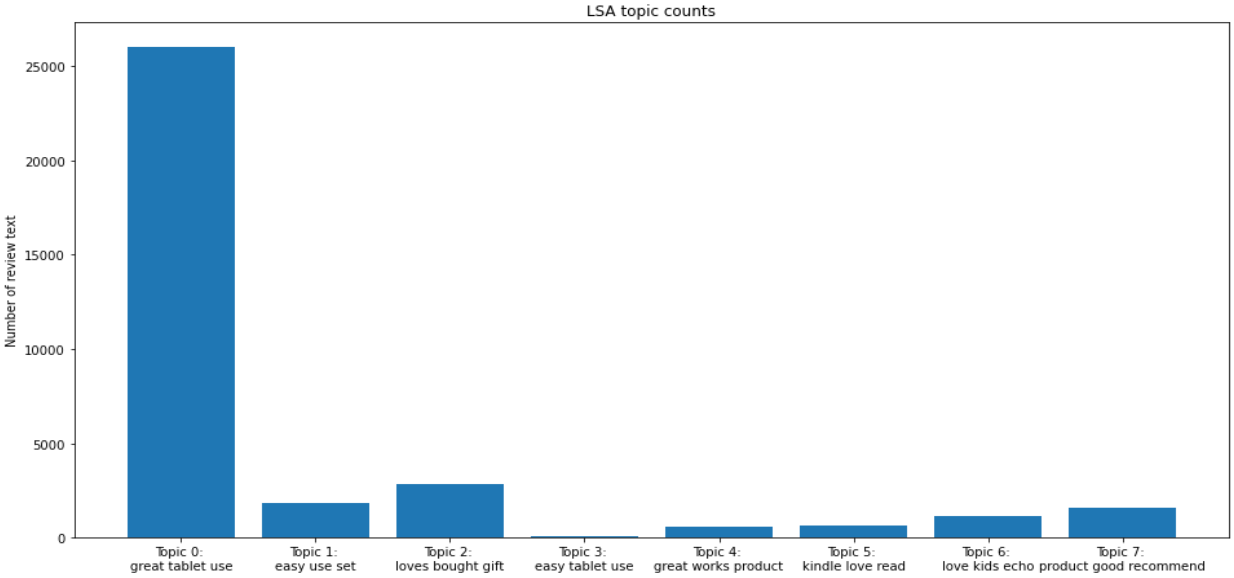


Figure 5.3: Output topic distribution of LSA model

t-Distributed Stochastic Neighbour Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data. In simpler terms, it shows how the data is arranged in a high-dimensional space.
For the TSNE model below mentioned parameters have been used:

Parameter	**Value**	**Description**
n_components	2	Dimension of the embedded space
Perplexity	50	It is related to the number of nearest neighbours that is used in other manifold learning algorithms. Larger datasets usually require a larger perplexity
learning_rate	100	Here keeping the learning rate is too high, the data look like a ‘ball’ with any point approximately equidistant from its nearest neighbours.

verbose	1	Verbosity level
n_iter	2000	Maximum number of iterations for the optimization
angle	0.75	It is used for method='barnes_hut'. This is the trade-off between speed and accuracy for Barnes-Hut T-SNE. 'angle' is the angular size of a distant node as measured from a point.

Table 5.3: T-SNE model parameters

The basic idea behind SNE is to minimize the cost function, that describes the divergence of low-dimensional representation to the actual (high-dimensional) data, using the Gradient Descent. The cost function, considered here, is the Kullback–Leibler divergence (KL Divergence).
The random distribution has been represented by the mathematical function of Shannon Entropy. It encodes information in the form of simple bits. So, if there is some difference in information between our representation and the actual representation, plugging Shannon Entropy into KL Divergence flags it. Minimizing the KL Divergence, therefore, reduces the error of the representation.
The t-SNE- LSA model worked really fast:

```
Indexed 34660 samples in 0.420s
Computed neighbours for 34660 samples in 7.873s
```

We have tracked the **KL divergence with the number of iterations**:

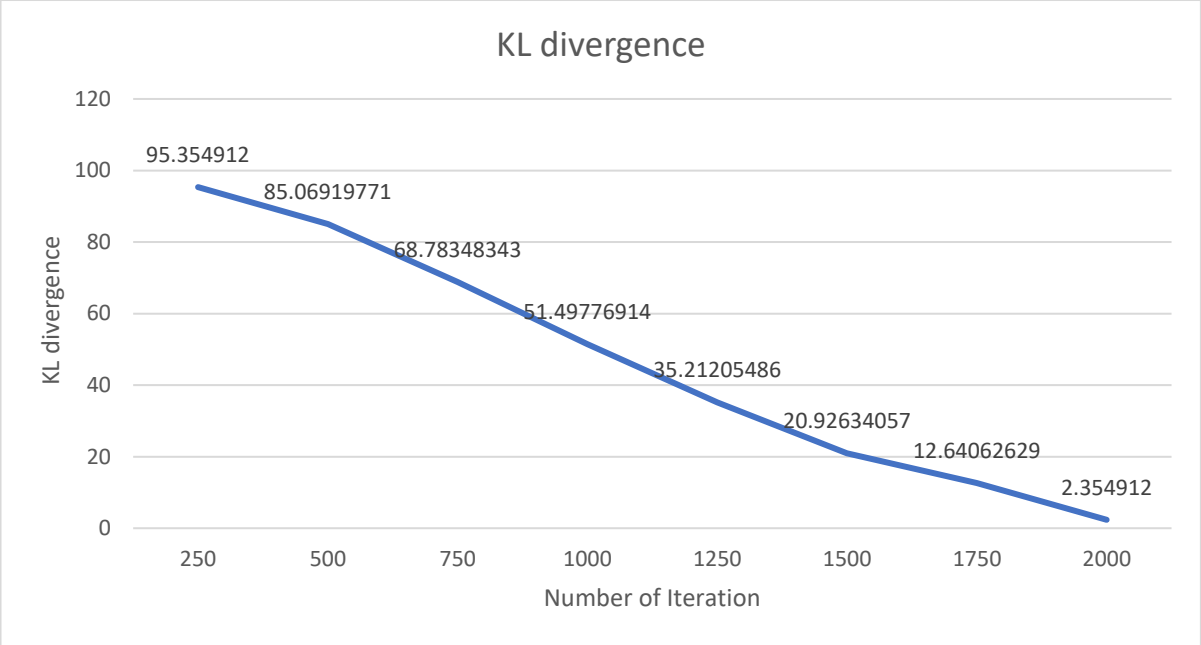


Figure 5.4: KL divergence of LSA model with the number of training iteration

KL divergence after 250 iterations with early exaggeration: 95.354912

KL divergence after 2000 iterations: 2.024379

Final Mean sigma = 0.025102

Below is the plot of lsa_mean_topic_vectors using the instance of
a **GlyphRenderer containing a MultiLine Glyph** that will be rendered as the graph edges:

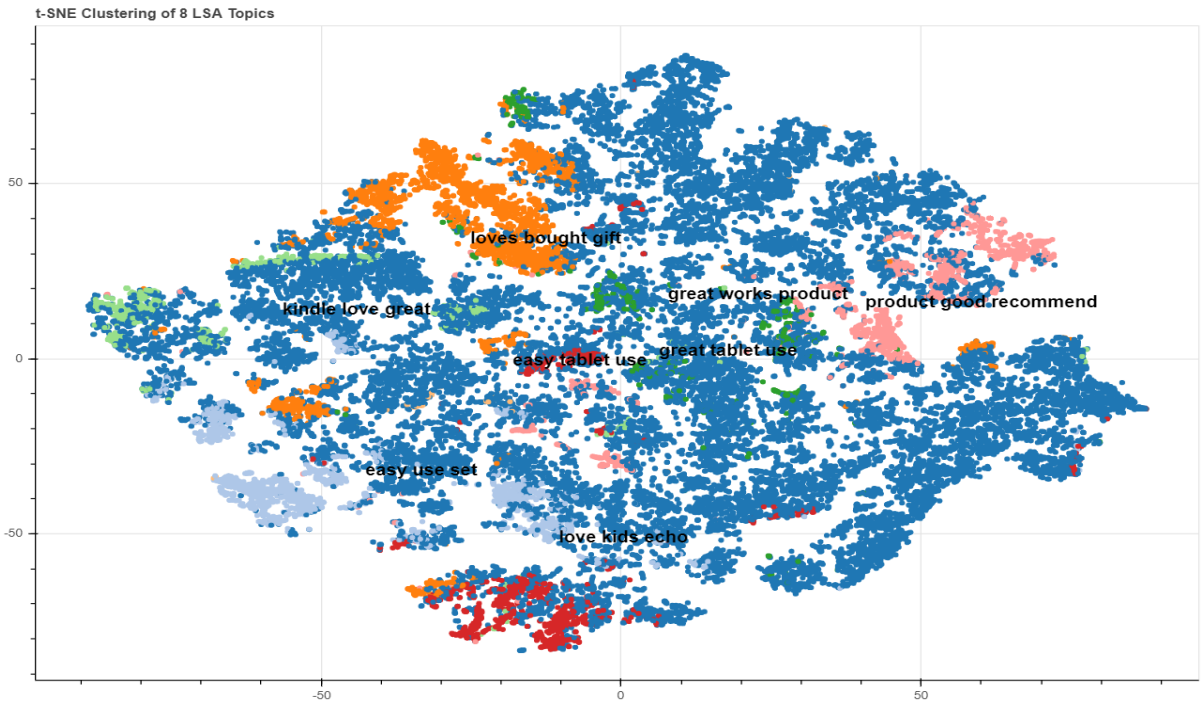


Figure 5.5: LSA topic clustering using GlyphRenderer

The Study has witnessed some of the advantages and disadvantages of using the LSA model:
1. In the model building phase of the study, LSA has taken significantly less run time compared to the other dimension reduction techniques because it only involves decomposing term-document matrix.
2. LSA can be implemented easily with many practical and scalable available tools. If the computational resource is available. The mahout implementation can train on big datasets.
3. LSA assures decent results, much better than plain vector space model even on dataset with diverse topics.
4. It is not sensitive to starting conditions (No weight initialization). Hence consistent.
5. However it has been observed that it is not the best solution to handle non-linear dependencies.
6. The dimension of the latent topics, cannot be chosen to arbitrary numbers. It depends on the rank of the matrix. The number of topics mostly based on heuristics decisions.

5.3 BERTopic model performance metrics:

Although topic models such as LDA and LSA have shown to be a good starting point, these Bayesian belief networks take a lot of effort in hyperparameter tuning to create meaningful and contextual topics.Pre-trained models like BERT helps to get more accurate representations of words and sentences. However, BERT embedding is token-based. It is difficult to get the document and word embedding in the same space. In the same space, the resulting size of the word embeddings will be quite large due to the contextual nature of BERT with degraded quality of sentence or word embedding. This section of the study focuses on a newly developed tool for topic modelling using the pre-trained language model BERT.

The subsequent sections will explain the use of two pre-trained models using in BERTopic framework and will compare the result.

- DistilBERT : DistilBERT-base-nli-mean-tokens
- XLMRoBERTa Model : xlm-r-bert-base-nli-stsb-mean-tokens

5.3.1 DistilBERT :

Transfer learning from large-scale language models has significantly improved upon the state-of-the-art on most of the Natural Language Processing task.
As these models are reaching a larger NLP community, there is a huge challenge to deploy this kind of giant model in the production under low latency constraints. Hence the focus should be on reducing the size of the model using different techniques like quantization (approximating the weights of a network with a smaller precision) and weights pruning (removing some connections in the network).
Distillation is a process that can be used to compress a large model into a smaller one. In the knowledge distillation technique small model is trained to reproduce the behaviour of a larger model. The distilled model, DistilBERT,has about half the total number of parameters of BERT base and retains 95% of BERT's performances on the language understanding benchmark GLUE (Sanh et al., 2019).
Below is the returned most frequent topics and their count using the DistilBERT using the BERTopic framework:

Topic	Count
-1	18011
13	1661
98	485
8	460
62	418

Table 5.4: Frequent topics and their count using the DistilBERT

-1 refers to all outliers and should typically be ignored.
The most four frequent topics with the top 10 words for each with their TF-IDF score have been generated are as follows:

Topic :13 [('alexa', 0.0832439203667664), ('music', 0.025075511691344827), ('echo', 0.020641593315075667), ('home', 0.013990859762077404), ('love', 0.013181085352653242), ('speaker', 0.013118691634023759), ('fun', 0.012850286858424985), ('our', 0.012090212581717441), ('play', 0.01167099095980577), ('all', 0.010578175855521734)]	**Topic: 98** [('son', 0.07870831232554668), ('loves', 0.0748881548121524), ('grandson', 0.06566752426581526), ('nephew', 0.03181200971933449), ('gift', 0.03164463526906614), ('husband', 0.027459794885542718), ('dad', 0.01851409768866387), ('birthday', 0.0182936227297177 54), ('loved', 0.0178299030756207), ('father', 0.014675701795020955)]
Topic:62 [('christmas', 0.15440779545202654), ('gift', 0.055289269639910744), ('bought', 0.03785299789267836), ('loves', 0.02392324631379239), ('loved', 0.01894240779502141), ('gifts', 0.01660183298210019), ('love', 0.01418333344984146), ('wife', 0.013929395155566446), ('family', 0.01088934556014015), ('presents', 0.010542670398870859)]	**Topic:8** [('loves', 0.0902904800495829), ('daughter', 0.07137088977677637), ('wife', 0.04417313375894341), ('gift', 0.04308427525560014), ('granddaughter', 0.034256672312420994), ('mom', 0.03257451597159008), ('mother', 0.02826525508699802), ('niece', 0.02625469791636546), ('loved', 0.01990972491586706), ('sister', 0.011809922500708768)]

Table 5.5: Four most frequent topics with the top 10 words and corresponding TF-IDF scores

BERTopic framework provides a method “find_topics” .
It takes arguments - search_term: the term used for searching for topics and **top_n:** the number of topics to return
It returns - similar_topics: the most similar topics from high to low and **similarity:** the similarity scores from high to low.

```
##model.find_topics("kindle",5)
([176, 139, 174, 165, 164],
 [0.8168287748273131,
  0.7917710579674354,
  0.7680752524552559,
  0.7521524190875335,
  0.7120956277259087])
```

Here it has been observed that the word “kindle” is **highly** associated with the **topic number 176 : The similarity score is = 0.8168287748273131**

```
Topic number : 176
[('kindle', 0.14032855368634517),
 ('love', 0.04208949815441863),
 ('easy', 0.033397214774327985),
 ('great', 0.02909335690650685),
 ('works', 0.027417427111918042),
 ('light', 0.019168915030004677),
 ('best', 0.016207195375987016),
 ('charging', 0.014940260815149119),
 ('perfect', 0.013932945199058445),
 ('kindles', 0.012829015655161735)]
```

Hence above-mentioned latent topic distribution proves that the “Kindle” product is highly appreciated by the reviewers.
However, it has also been observed the video and browser are little slow sometimes in amazon digital products.

##model.find_topics("bad",2)→ ([112, 133], [0.7898764586168462, 0.4899 9990884397326])

Topic :133

```
[('slow', 0.14988785243899994),
 ('little', 0.04805124982440035),
 ('mived', 0.03236328657359008),
 ('slowalso', 0.03236328657359008),
 ('sometimes', 0.029146367632999656),
 ('browsers', 0.028962010138085102),
 ('unreliable', 0.028071355356702696),
 ('slowed', 0.027380506658933428),
 ('flixster', 0.026816043634803537),
 ('videos', 0.025827088646276107)]
```

LDAvis is a web-based interactive visualisation of topics estimated using LDA. It provides a global view of the topics (differences from each other), while at the same time allowing for a deep inspection of the terms most highly associated with each individual topic. It shows topics, their sizes, and the corresponding words. We can see that **topic 13** has the largest size in the corpus.

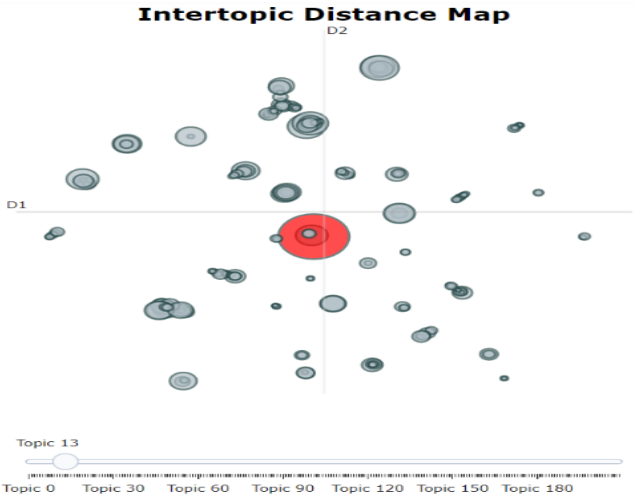


Figure 5.6: Intertopic distance map (using DistilBERT)

The variable probabilities that are returned from transform() or fit_transform() methods, can be used to understand how confident BERTopic is, that certain topics can be found in a document. In the below visualization it can be observed that **for document number 60,topic number 201 can be the best possible choice.**

```
##model.visualize_distribution(probabilities[60])
```

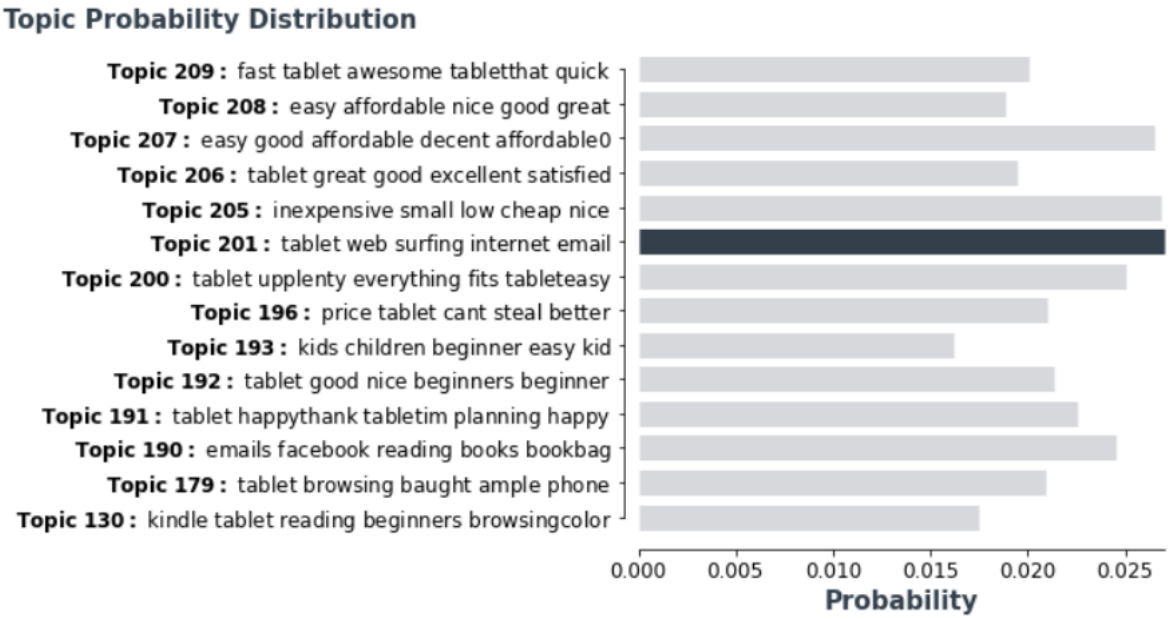


Figure 5.7: Topic Probability distribution (Using DistilBERT)

5.3.2 XLMRoBERTaModel

XLM-RoBERTa is based on Facebook's RoBERTa model released in 2019. It is a large multi-lingual language model, trained on 2.5TB of filtered Common Crawl data.
This pretraining multilingual language models at scale leads to significant performance gains for a wide range of cross-lingual transfer tasks.
The model, dubbed XLM-R, significantly outperforms multilingual BERT (mBERT)on a variety of cross-lingual benchmarks, including +13.8% average accuracy on XNLI,+12.3% average F1 score on MLQA, and +2.1% average F1 score on NER.
Here in this section, the study focuses on figuring out the latent topic using the pre-trained model: "xlm-r-bert-base-nli-stsb-mean-tokens".
Below is the returned most frequent topics and their probability using the RoBERTa using the BERTopic framework :

Total initially number of topics created is 219

Topic	Count
-1	15918
3	4920
157	1114
203	557
40	444

Table 5.6: Frequent topics and their count using the RoBERTa (without the reduction of the topic)

Here we can see that most of the reviews are mapped to the topic number :3

```
Topic 3
[('echo', 0.03687196358786089),
 ('alexa', 0.03238867998637626),
 ('speaker', 0.015381841239869985),
 ('love', 0.012992341978599704),
 ('amazon', 0.011967300788317698),
 ('use', 0.011718386528947617),
 ('fun', 0.011695486038399719),
 ('play', 0.010703743722549643),
 ('all', 0.010600162908186385),
 ('lights', 0.010411746421436802)]
```

The word **"Kindle"** is having below mentioned top 5 most similar topics with the corresponding similarity scores.

```
model.find topics("kindle",5)
```

([195, 191, 84, 82, 114], [0.9056526279433785, 0.8567802950595041, 0.8464722197449064, 0.8195332645966598, 0.8174438273924751])	Topic : 195 [('kindle', 0.13268312806568128), ('love', 0.041508411799301655), ('easy', 0.025240272489372634), ('charging', 0.020566202778353707), ('use', 0.020014809259419633), ('light', 0.017591473063993876), ('super', 0.01572599224255231), ('best', 0.01487347825650475), ('charger', 0.013102136849417792), ('product', 0.012685771755058067)]

Table 5.7: Top five similar topics associated with the word "Kindle"

It can be clearly observed that, for the same topic search("Kindle") RoBERTa returns better similarity score than the DistilBERT.

Reduction of number of topics:

Finally, we can also reduce the number of topics after having trained a BERTopic model.The final number of topics can be decided after knowing how many are actually created in the base model. It is difficult to predict before training the model how many topics that are in your documents and how many can be extracted.

Here in our study initially the number of topics created was 219. On the top of the base model the number of topics have been reduced to 15 and the method generates new set of topics and their associated probabilities.

Topic	Count
-1	18431
3	5141
157	1539
209	1033
203	1031
40	908
191	879
38	855

Topic	Count
178	744
80	725
100	721
128	591
42	583
105	536
121	504
145	439

Table 5.8: Frequent topics and their count using the RoBERTa (Reduced number of topics)

Here we can see for the topic number 3 the clustered words are really **related** and **contextually connected** to each other.

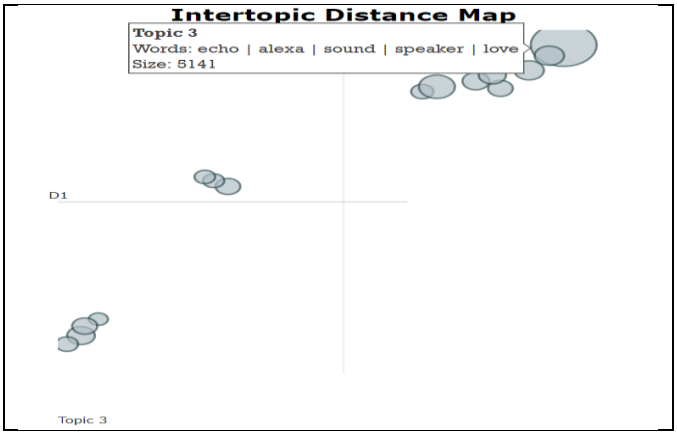


Figure 5.8: Intertopic distance map for reduced number of topics(using RoBERTa)

We can see that Distilled versions of pre-trained Transformers often come pretty close to the performance of the original models with quicker inference. Depending upon the hyperparameter it sometimes surpasses the original models as well.
It is also observed that RoBERTa with the reduced topic number gives more contextually related words .However the new word distribution has altered the similarity score for the same word, example The most similar topic associated with the term "KIndle" has achieved less similarity score compared to the baseline RoBERTa model.

```
##model.find_topics("kindle",5)
([191, -1, 3, 145, 121],
 [0.7670670868378444,
  0.7394531463151468,
  0.6792300248409371,
  0.6436546431271519,
  0.631368983651063])
```

5.4 Combination of BERT and LDA :

The final model, the research has been focused on is the combination of LDA and BERT vector with proper hyperparameter tuning. Here in this section in the state of the art model an iterative test has been done with different value of the hyperparameter, gamma(weightage of the BERT embedded vector) and has examined the coherence and Silhouette score.
The research has found at gamma value=20 the model gives the best coherence score.

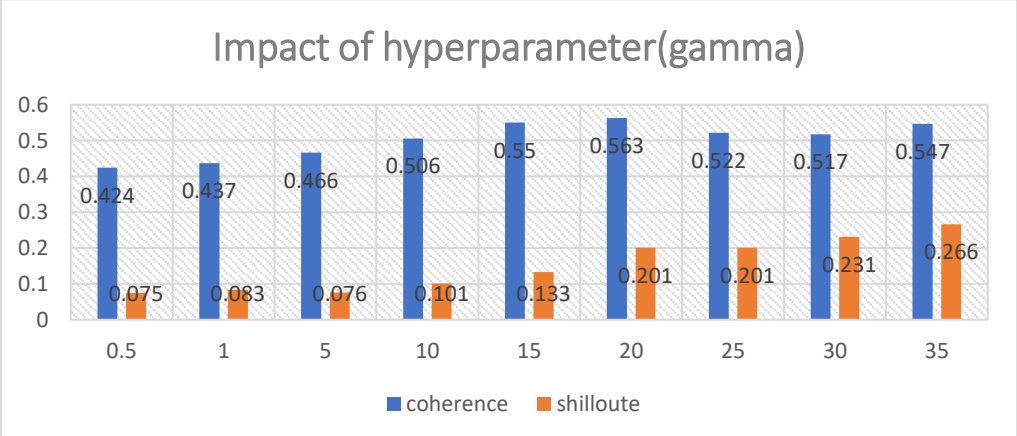


Figure 5.9: Impact of hyperparameter gamma in coherence and Silhouette score

The concatenated vector presents in the high dimension space. Hence it passes through an autoencoder to get the bottleneck layer of vector representation in the lower dimension space, with the help of the validating component decoder (minimizing the loss).
Sample of concatenated vector in the lower dimension:

```
array([[0.79890937, 1.1763643 , 2.604254  , 0.7206831 , 0.
,
2.0258577 , 2.0263774 , 1.2521442 , 1.5883904 , 2.5221255 ,
0.        , 0.65937734, 0.        , 0.5991399 , 0.        ,
0.5736817 , 1.9525373 , 0.9691872 , 0.8326144 , 1.7708691 ,
2.0201058 , 0.19014683, 1.2993094 , 1.2632446 , 0.43972808,
1.4793357 , 0.25728795, 0.62760544, 1.3991053 , 2.2291074 ,
0.        , 1.1836087 ],
[1.3131022 , 1.9998624 , 1.0745927 , 1.7032753 , 0.        ,
1.7238976 , 1.4294902 , 1.3298627 , 1.9024223 , 2.7282615 ,
0.        , 1.4629661 , 0.        , 1.2921325 , 0.        ,
1.0336426 , 0.8561941 , 0.9157458 , 0.8644724 , 1.7075573 ,
0.5368787 , 0.66143817, 1.8525901 , 0.49463996, 3.1818063 ,
1.7228783 , 1.7243465 , 2.0797246 , 1.9717574 , 1.9647243 ,
0.        , 2.5908735 ],
[0.8245282 , 0.65194345, 1.1003748 , 2.5253303 , 0.        ,
1.5046338 , 1.3541301 , 1.3545483 , 0.9203954 , 1.9174873 ,
0.        , 0.45859468, 0.        , 1.9758368 , 0.        ,
1.2402754 , 1.1543295 , 1.7337443 , 0.992851  , 0.99614376,
0.        , 0.9306882 , 1.8602655 , 0.9828926 , 2.2802725 ,
2.2648747 , 1.0512795 , 1.3900199 , 0.8663801 , 0.48777187,
0.        , 2.1417358 ]], dtype=float32)
```

Figure 5.10: Sample of concatenated vector in the lower dimension

In the final segment of the implementation pipeline has used the customized utility class with the required visualization methods to derive following evaluation metrics:

Functions	Description
get_topic_words	get top words within each topic from clustering results
get_coherence	Get model coherence from gensim.models.coherencemodel :param model: Topic_Model object :param token_lists: token lists of docs :param topics: topics as top words :param measure: coherence metrics :return: coherence score
get_silhouette	Get silhouette score from model :param model: Topic_Model object :return: silhouette score
plot_proj	Plot UMAP embeddings :param embedding: UMAP (or other) embeddings :param lbs: labels
visualize	Visualize the result for the topic model by 2D embedding (UMAP) :param model: Topic_Model object
get_wordcloud	Get word cloud of each topic from fitted model :param model: Topic_Model object :param sentences: preprocessed sentences from docs

Table 5.9: Utility class to evaluate the combination model performance

The below mentioned Silhoutte score and coherence score have been achieved.

Silhouette Score: 0.201
Coherence Score: 0.563

Word cloud output analysis: In the EDA section we had examined the word clouds generated from the cleaned raw text. It shows the words frequently clustered together. However, it did not show us the words which are contextually related. Here in this section, the output vector we have got using BERT embedding methods show much better coherence score with a lucid representation of the contextual word cloud, which realistically represent the original reviews of the customer.

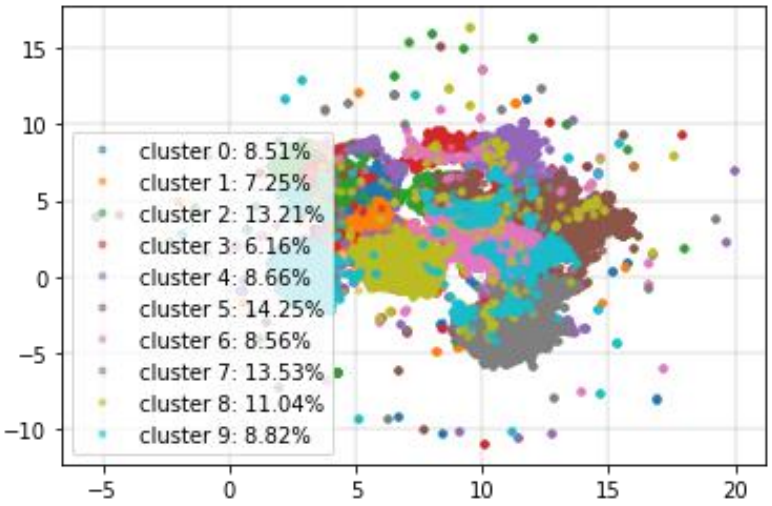


Figure 5.11: Clustering result on vectors from contextual topic embedding (2D UMAP)

Cluster number : 0

Figure 5.12: Word cloud for cluster number zero

In cluster 0, word cloud points out the most effective related words are app,Amazon,tablet,movie ,fire ,store etc .
Here people mostly talk about the amazon app store or operating system, tablet, and it is been observed that the reviews are highly contextual with mix emotions.
Examples:

“great device for the price. good looking screen, quick and efficient. amazon app store isn't that bad either”
“This is a superb tablet for the price. It doesn't come with the Android operating system. It has Amazon's operating system. The battery last a good long time and apps run very fast. I would recommend this to anyone”
“This tablet runs smooth with all amazon apps, the internet runs at a steady frame rate and it had a decent sized screen.”
"Biting as a gift, my only concern was the size and the Amazon interface. It was easy enough to download the Google play store, so it was familiar to the user. Size was an initial concern, but it was just right to watch movies without being so cumbersome."
"I love this tablet. The battery life last a long time. I use my Amazon Kindle Fire predominately for watching movies on Amazon Prime,Netflix's, and checking my emails. I also carry it to my appointments and meetings. It is a compact tablet, and not big and bulky. The playback of the movies are great, and the movies are HD. Great little tablet if you are looking for something to watch movies and checking emails. I would definitely recommend this tablet to friends and family members."

Table 5.110: Examples of reviews corresponding to the cluster number 0

5.5 Summary

This section has discussed the results and performance metrics generated from all the experiments outlined in chapter 5.
Firstly, it shows the performance of LDA topic modelling techniques on the incoherent data and explains how the variation of LDA(Mallet) with different inference algorithm(Gibbs Sampling) helps to enhance the performance.
As a part of HBN implementation, the research also applies LSA model to compare the results and efficiency in terms of processing time and clustering contextual words of different topics.
Later in the Transformer based model implementation, the research uses a newly developed framework for topic modeling and examines two different pre-trained BERT model(RoBERTa and Distil-compressed BERT). It founds Distil BERT gives much faster processing time. However, RoBERTa provides more accurate metrics.
Finally, the model with the combination of LDA and BERT vector with the essential weightage also shows at a certain weight to the BERT vector the model gives the highest coherence score and an optimum silhouette score.

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

This section will discuss how this study performed to meet the objectives and goals proposed in the research proposal.
The segment will likewise summarise the appropriate responses assembled as a feature of this investigation as a part of answering the research questions mentioned in section 1.4.
Finally, it sums up the commitment of this study in the field of AI(Deep NLP), with the explanation of the contributions to the knowledge and the impediments of this investigation that can be stretched out in future examinations.

6.2 Answering Research Questions

Initially, as a part of EDA, the study focused on the basic NLP techniques to find the latent temporal behaviours of the customers' review.
Then gradually, the research work drives towards different traditional Topic modelling, hierarchical Bayesian techniques and finally ends up with the implementation of the latest language model, BERT, in different ways to enhance the performance of topic modelling task.
Below are the precise answers to the research questions, that the study had proposed to examine:

- What are the latest topics of discussion among the customer reviews on Amazon's Digital products? →As a part of EDA, basic NLP text mining techniques have been implemented to understand the temporal behaviour of the reviewers. It shows that majority of the sentiment polarity scores are greater than zero, and the ratings are aligned with the polarity score, that is, quite high at 4 or 5 ranges.
  To get the essence of the underlying context, the trigram model has been built. Product-wise Kindle and amazon's tablets are loved by the customers.
- The reviews don't consist of one single topic. How Bayesian Topic modelling techniques work in highly sparsed data and what are the performance evaluation criteria of these methods? → As a part of model implementation, to find the latent topic, the study uses HBN, like LDA and LSA. The variation of LDA, mallet, shows a significant rise in the coherence score. Hence, concluded as an inferential algorithm Gibbs sample works better than the Variational inference.
- Use of BERT as an embedding layer → It has improved the coherence score. Apart from time constraints, RoBERTa works better than DistilBERT.
- How incorporation of BERT embedded vector with a self-attention mechanism in a traditional Bayesian technique can enhance the model performance? → It mitigates the exhaustive hyperparameter tuning steps in the hierarchical Bayesian networks (LDA and LSA). Integration of LDA and BERT embedded vector gives considerably good coherence score, and the word cloud in the lower dimension shows the contextually related words clustered together.

6.3 Discussion and Conclusion

The objective of this research work was to examine the capabilities of different Bayesian belief networks to identify the underlying topics of highly sparsed and incoherent corpus of document and compare it with the potential of pre-trained NLP model BERT to get a more accurate context-based topic.

Here in this study, as a part of EDA, the generic NLP text mining technique shows that the overall reviews are inclined towards the positive trend with a good recommendation and product rating.
Initially, the research focuses on the LDA and LSA models and has witnessed that LSA works much faster than LDA. However, Mallet LDA shows the best performance. The baseline Mallet model with Markov Chain Monte Carlo inference algorithm, to sample the parameters, surpass the performance of generic LDA and LSA.
However, language model like BERT has shown considerably huge progress in the performance metrics with a very less effort.
Framework BERTopic has been used in this study and as an argument two different BERT models have been used, RoBERTa and DistilBERT.
DistilBERT gives a faster inference speed with a bit compromised prediction metrics.
The result shows that on average RoBERTa has increased the similarity score of a particular topic by around 11 %. DistilBERT, is a starting reasonable choice. However, for the best prediction metrics, Facebook's RoBERTa should be the ideal choice.
BERTopic framework is designed in such a way, that any pre-trained model can be used for the topic modelling task.
Finally, the research shows the result comes from the combination of BERT and LDA vector, which gives a significant high Coherence Score: 0.563 and a good Silhouette Score: 0.201.
After visual inspection of the explanations and analysis of evaluation generated from different models outlined in sections 4 and 5, the research has concluded that if the raw text is highly incoherent, using the BERT language model with a self-attention mechanism is the optimum choice. BERT can be used as a stand-alone solution or, it can also be combined with LDA or LSA vectors. The result of the model also shows that words clustered together are highly contextual and related.
Starting from the literature review and the proposal, the research focuses on the use of language model in modern day NLP tasks.

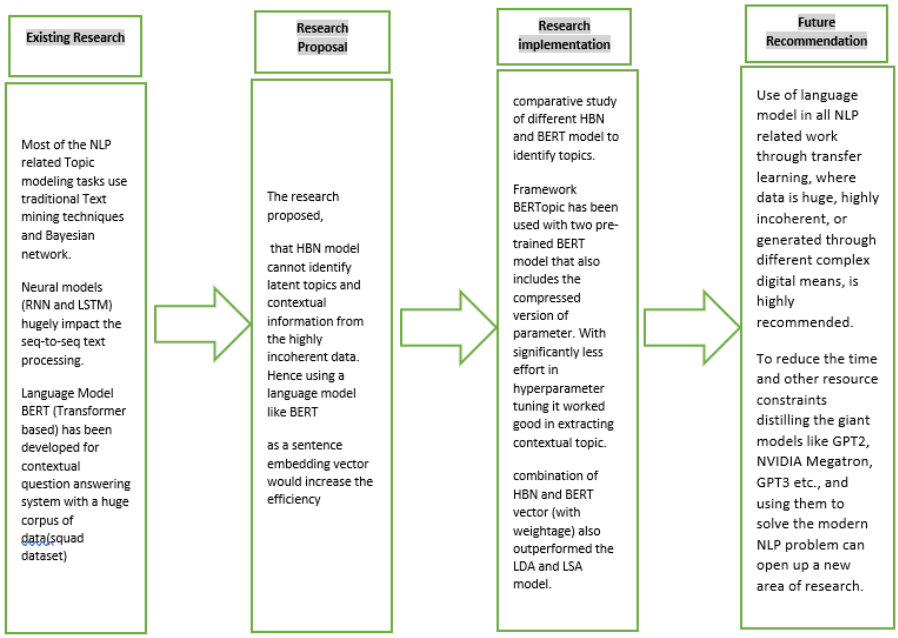


Figure 6.1: Progress and Impact of the research work

In the following sub sections the detailed contribution and the future recommendation of this research will be explained.

6.4 Contribution to knowledge

The major breakthrough of neural NLP was the introduction of variation of RNNs such as Bidirectional-RNNs which process text in both left to right and right to left and character-level RNNs for enhancing underrepresented or out of vocabulary word embeddings. It opens up a new approach to process the sequential data.
While standard RNN architectures have led to incredible breakthroughs in NLP they suffer from a variety of challenges.
One cause for sub-optimal performance standard RNN encoder-decoder models for sequence-to-sequence tasks such as NER or translation is that they weigh the impact of each input vector evenly on each output vector when in reality specific words in the input sequence may carry more importance at different time steps.
The attention mechanism provides a means of weighting the contextual impact of each input vector on each output prediction of the RNN.
BERT is a state-of-the-art language model, primarily built with the intention to develop a contextual question answering system (SQuAD v1.1, SQuAD v2.0 T).
However, when a very promising development of neural NLP, ELMo has spawned the technique BERT, with attention transformers instead of bi-directional RNNs to encode context, the state-of-the-art language model started impacting so many real-life business problems.
Nowadays topic modelling is a very important business problem. In this digital and competitive era the success of any real business lies on the success of understanding the temporal behaviour of its customer.
In the last two decades, most of the business problems used to be solved using Bayesian techniques. It takes a lot of effort to build and also gives compromised predictions.
However, the ready to use language model with a very minimum tuning can give us a much more desirable prediction.
This research tried to show that, even for generic NLP related problems pre-trained language models work better, though they were initially built as a state of the art model for a different specific task.

6.5 Future Recommendations

The essence of this research work lies in the idea of efficiently using advanced language models in the common NLP related business problems.
The objective can be achieved in two steps:
1. Using language model through transfer learning with hyperparameter tuning
2. Distilling the giant model (compression)
This research solely focused on the implementation of the BERT language model to solve the Topic modelling task on the realistically generated incoherent data through modern digital platforms. We have found that BERT works considerably well in this kind of data set with a very minimum effort. For each task, the pre-trained model needs to be fine-tuned to customize it to the data at hand. Fine-tuning involves gradient updates for most of the pre-trained neural model and, the updated weights are then stored for making predictions on respective NLP tasks.
Here in this research, a few hyperparameter tuning was required in BERT compared to the naive LDA or LSA kind of models. However, in the era of recent developments in the NLP space, it is very evident that GPT 3 is the most promising language model for any NLP related tasks.It's a giant language model developed by the OpenAI researchers.

The largest GPT-3 model has a 175 billion parameter. This is 470 times bigger than the largest BERT model (375 million parameters). But the major challenge is to put it in production.

In this study it has been observed that the DistilBERT has improved the processing time, using Knowledge distillation (**teacher-student learning**).

This is a compression technique, in which a small model is trained to reproduce the behaviour of a larger model. In many cases, a good performance model predicts an output distribution with the correct class having a high probability, leaving other classes with probabilities near zero.

But, some of these "almost-zero" probabilities are larger than the others, and this reflects, in part, the generalization capabilities of the model.This uncertainty is referred to as the **"dark knowledge".** In the teacher-student training of DistilBERT, the student network mimics the full output distribution of the teacher network (its knowledge) and the optimization of Kullback-Leibler loss ensure the training.

Usually, based on the Transformer architecture of these pre-trained language models keep getting larger and larger and being trained on bigger datasets.

The latest model from Nvidia has 8.3 billion parameters: 24 times larger than BERT-large, 5 times larger than GPT-2, while RoBERTa, the latest work from Facebook AI was trained on 160GB of text.The most exciting future scope of this research work in this space would be the utilization and compression of an already developed advanced language model in solving modern-day NLP problems, like topic identification.

REFERENCE

Alammar, J., (2019) *The Illustrated GPT-2 (Visualizing Transformer Language Models)*. [online] Available at: http://jalammar.github.io/illustrated-gpt2/.

Alghamdi, R. and Alfalqi, K., (2015) A Survey of Topic Modeling in Text Mining. *nternational Journal of Advanced Computer Science and Applications*, [online] 61, p.7. Available at: http://thesai.org/Downloads/Volume6No1/Paper_21-A_Survey_of_Topic_Modeling_in_Text_Mining.pdf.

Bahdanau, D., Cho, K.H. and Bengio, Y., (2015) Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, [online] pp.1–15. Available at: https://arxiv.org/pdf/1409.0473.pdf.

Bentley, P.J., Gulbrandsen, M. and Kyvik, S., (2015) The relationship between basic and applied research in universities. *Higher Education*, [online] 704, pp.689–709. Available at: http://dx.doi.org/10.1007/s10734-015-9861-2.

Bianchi, F., Terragni, S. and Hovy, D., (2020) Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. [online] Available at: http://arxiv.org/abs/2004.03974.

Blei, D., Carin, L. and Dunson, D., (2010) Probabilistic topic models. *IEEE Signal Processing Magazine*, [online] 276, pp.55–65. Available at: https://www.semanticscholar.org/paper/Probabilistic-topic-models-Blei/7314be5cd836c8f06bd1ecab565b00b65259eac6.

Blei, D.M. and Lafferty, J.D., (2009) Topic models. [online] Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.463.1205&rep=rep1&type=pdf#page=96.

Blei, D.M., Ng, A.Y., Jordan, M.I. and Lafferty, J., (2003) Latent Dirichlet allocation. *Journal of Machine Learning Research. 3: 993–1022. doi:10.1162/jmlr.2003.3.4-5.993*. [online] Available at: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Bourlard, H. and Kamp, Y., (1988) Auto-association by multilayer perceptrons and singular value decomposition. [online] Available at: https://link.springer.com/article/10.1007/BF00332918.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., (2020) Language Models are Few-Shot Learners. [online] Available at: http://arxiv.org/abs/2005.14165.

Burk, H., (1999) Das Hunderttage-Stadion : Entstehungsgeschichte des Bad Nauheimer Kunsteisstadions unter Colonel Paul R. Knight. [online] pp.1715–1725. Available at: http://www.aclweb.org/anthology/P16-1162.

Cao, L. and Fei-Fei, L., (2007) Spatially coherent latent topic model for concurrent

segmentation and classification of objects and scenes. *2007 IEEE 11th International Conference on Computer Vision*. [online] Available at: http://www.ifp.illinois.edu/~cao4/papers/CaoFei-Fei_ICCV2007_final.pdf.

Clark, C., Lee, K., Zettlemoyer, L., Peters, M.E., Neumann, M., Iyyer, M. and Gardner, M., (2018) Deep contextualized word representations. [online] Available at: https://arxiv.org/pdf/1802.05365.pdf.

Cui, Q., Gao, B., Bian, J., Qiu, S., Dai, H. and Liu, T.Y., (2015) KNET: A general framework for learning word embedding using morphological knowledge. *ACM Transactions on Information Systems*, [online] 341. Available at: https://dl.acm.org/doi/10.1145/2797137.

Datafiniti(Kaggle), (n.d.) *Datafiniti- Amazon Digital Product*. [online] Available at: https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products?select=1429_1.csv.

Daud, A., Li, J., Zhou, L. and Muhammad, F., (2010) Knowledge discovery through directed probabilistic topic models: A survey. *Frontiers of Computer Science in China*, [online] 42, pp.280–301. Available at: https://link.springer.com/article/10.1007/s11704-009-0062-y.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, [online] 1Mlm, pp.4171–4186. Available at: https://www.arxiv-vanity.com/papers/1706.03762/.

Emmery, C., (2014) Topic Modelling in Online Discussions: Analysis of the Developments within the Dutch Privacy Debate on News Websites. [online] Master of, p.102. Available at: https://www.clips.uantwerpen.be/clips.bak/sites/default/files/thesisfinal_p.pdf.

Encyclopedia, (n.d.) Long short-term memory. [online] Available at: http://sciencewise.info/resource/Long_short_term_memory/Long_short_term_memory_by_ Wikipedia.

Ghodsi, A., (2006) Dimensionality Reduction A Short Tutorial. [online] Available at: https://www.math.uwaterloo.ca/~aghodsib/courses/f06stat890/readings/tutorial_stat890.pdf.

Gupta, V. and Lehal, G.S., (2009) A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, [online] 11, pp.60–76. Available at: http://www.jetwi.us/uploadfile/2014/1230/20141230112729939.pdf.

Hochreiter, S. and Urgen Schmidhuber, J., (1997) LONG SHORT-TERM MEMORY. *Neural Computation*, [online] 98, p.17351780. Available at: http://www7.informatik.tu-muenchen.de/~hochreit%0Ahttp://www.idsia.ch/~juergen.

Hofmann, T., (1999) Probabilistic Latent Semantic Analysis. *Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval*. [online] Available at: https://arxiv.org/ftp/arxiv/papers/1301/1301.6705.pdf.

Hofmann, T., (2001) Unsupervised learning by probabilistic Latent Semantic Analysis. *Machine Learning*, [online] 421–2, pp.177–196. Available at: https://link.springer.com/article/10.1023/A:1007617005950.

Hugging Face AI, (2019) PyTorch Pretrained BERT: The Big & Extending Repository of pretrained Transformers. [online] Available at: https://github.com/huggingface/pytorch-pretrained-BERT%0D.

Jo, Y., Lee, L. and Palaskar, S., (2017) Combining LSTM and Latent Topic Modeling for Mortality Prediction. [online] Available at: https://arxiv.org/pdf/1709.02842.pdf.

Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S. and Gurusamy, V., (2015a) Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, [online] 51, pp.7–16. Available at: https://www.researchgate.net/publication/273127322_Preprocessing_Techniques_for_Text_Mining.

Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S. and Gurusamy, V., (2015b) Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, [online] 51, pp.7–16. Available at: https://www.researchgate.net/profile/Vijayarani_Mohan/publication/339529230_Preprocessing_Techniques_for_Text_Mining_-_An_Overview/links/5e57a59f299bf1bdb83e7972/Preprocessing-Techniques-for-Text-Mining-An-Overview.pdf.

Karandikar, A. and Finin, T., (2010) Clustering short status messages : A topic model based approach. [online] Available at: https://ebiquity.umbc.edu/_file_directory_/papers/518.pdf.

Kim, S.W. and Gil, J.M., (2019) Research paper classification systems based on TF-IDF and LDA schemes. *Human-centric Computing and Information Sciences*, [online] 91. Available at: https://doi.org/10.1186/s13673-019-0192-7.

Kim, Y., Denton, C., Hoang, L. and Rush, A.M., (2017) Structured attention networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, [online] pp.1–21. Available at: https://arxiv.org/pdf/1702.00887.pdf.

Le, Q. and Mikolov, T., (2015) Distributed Representations of Sentences and Documents. [online] 32, pp.29–30. Available at: https://cs.stanford.edu/~quocle/paragraph_vector.pdf%0Ahttp://dl.acm.org/citation.cfm?doid=2740908.2742760.

Liu, P.J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, Ł. and Shazeer, N., (2018) Generating wikipedia by summarizing long sequences. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, [online] pp.1–18. Available at: https://arxiv.org/pdf/1801.10198.pdf.

Maarten Grootendorst, (2020) *BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics*. Available at: https://doi.org/10.5281/zenodo.4430182.

Manning, C.D., Raghavan, P. and Schütze, H., (2009) An Introduction to Information Retrieval. *Cambridge University Press Cambridge, England*, [online] c. Available at: https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf.

Mao, L., (2019) *Word2Vec Models Revisited*. Available at: https://leimao.github.io/article/Word2Vec-Classic/.

Neville, C., (2007) Effective Learning Service: Introduction to Research and Research Methods. *Bradford University School of Management*, [online] pp.1–44. Available at: https://www.unrwa.org/sites/default/files/introduction-to-research-and-research-methods.pdf.

Nozza, D., Bianchi, F. and Hovy, D., (2020) What the [MASK]? Making Sense of Language-Specific BERT Models. [online] Available at: https://arxiv.org/pdf/2003.02912.pdf.

Onan, A., Korukoğlu, S. and Bulut, H., (2016) LDA-based Topic Modelling in Text Sentiment Classification: An Empirical Analysis. *International Journal of Computational Linguistics and Applications*, [online] 71, pp.101–119. Available at: https://www.ijcla.org/2016-1/IJCLA-2016-1-pp-101-119-preprint.pdf.

Papadimitriou, C.H., Raghavan, P., Tamaki, H. and Vempala, S., (2000) Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, [online] 612, pp.217–235. Available at: https://www.sciencedirect.com/science/article/pii/S0022000000917112.

Porter, M.F., (1980) An algorithm for suffix stripping. *Program: electronic library and information systems*. [online] Available at: https://www.emerald.com/insight/content/doi/10.1108/eb046814/full/html.

Pugh, S., (1989) Research in engineering, research in design research in engineering design. They're not one and the same thing. *IEE Colloquium on Research in Engineering Design*. [online] Available at: https://ieeexplore.ieee.org/document/197941.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2018) Language Models are Unsupervised Multitask Learners. [online] Available at: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Rogers, A., Kovaleva, O. and Rumshisky, A., (2020) A Primer in BERTology: What we know about how BERT works. [online] Available at: http://arxiv.org/abs/2002.12327.

Rumelhart, D.E., Hinton, G.. and Williams, R.J., (2019) Parallel Distributed Processing. *Parallel Distributed Processing*. [online] Available at: https://web.stanford.edu/class/psych209a/ReadingsByDate/02_06/PDPVolIChapter8.pdf.

Sanh, V., Debut, L., Chaumond, J. and Wolf, T., (2019) DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv*, [online] pp.2–6. Available at: https://arxiv.org/pdf/1910.01108v4.pdf.

Shao, S., (n.d.) Contextual Topic Identification. [online] Available at: https://blog.insightdatascience.com/contextual-topic-identification-4291d256a032.

Smilkov, D. and Group, B.P., (2016) *Open sourcing the Embedding Projector: a tool for visualizing high dimensional data*. [online] Google AI Blog. Available at: https://ai.googleblog.com/2016/12/open-sourcing-embedding-projector-tool.html.

Stanford.edu, (2019) *Recurrent Neural Networks*. Available at: https://stanford.edu/~shervine/teaching/cs-230/.

Steyvers, M. and Griffiths, T., (2010) Probalistic Topic Models. *Latent Semantic Analysis: A*

Road To Meaning, [online] 33, pp.993–1022. Available at: http://www.sciencedirect.com/science/article/pii/S0140366413001047%5Cnhttp://ceas.cc/2004/167.pdf%5Cnhttp://doi.acm.org/10.1145/1806338.1806450%5Cnhttp://eprints.soton.ac.uk/272254/%5Cnhttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7033160%25.

TensorFlow, (2020) *Embedding Projector TensorFlow*. [online] Available at: https://towardsdatascience.com/visualizing-bias-in-data-using-embedding-projector-649bc65e7487%0Ahttp://projector.tensorflow.org/.

Tokunaga, T. and Iwayama, M., (1994) Text categorization based on weighted inverse document frequency. [online] 1994, pp.5–31. Available at: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=E984BFA36AA311456E6F2B4D07539F93?doi=10.1.1.49.7015&rep=rep1&type=pdf.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017) Attention is all you need. *Advances in Neural Information Processing Systems*, [online] 2017-DecemNips, pp.5999–6009. Available at: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, A. and Ranganathan, V., (2019) Is this question sincere ? Identifying insincere questions on Quora using BERT and variations. *web.stanford.edu*. [online] Available at: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15763730.pdf.

Wang, Y., Yao, H. and Zhao, S., (2015) Auto-Encoder Based Dimensionality Reduction Neurocomputing Auto-encoder based dimensionality reduction. *Neurocomputing*, [online] 184November, pp.232–242. Available at: http://dx.doi.org/10.1016/j.neucom.2015.08.104.

Wikipedia, (2020a) *Amazon.com Wikipedia*. [online] Available at: https://en.wikipedia.org/wiki/Amazon_(company).

Wikipedia, (2020b) *Natural Language Processing*. [online] Available at: https://en.wikipedia.org/wiki/Natural_language_processing.

Wikipedia, (n.d.) Gated recurrent unit. [online] Available at: https://en.wikipedia.org/wiki/Gated_recurrent_unit.

Wikipedia, (n.d.) k-means clustering. [online] Available at: https://en.wikipedia.org/wiki/K-means_clustering.

Xie, P. and Xing, E.P., (2013) Integrating document clustering and topic modeling. In: *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*. [online] pp.694–703. Available at: https://arxiv.org/ftp/arxiv/papers/1309/1309.6874.pdf.

APPENDIX A: RESEARCH PLAN

The research plan considering the optimum situation (Gantt chart):

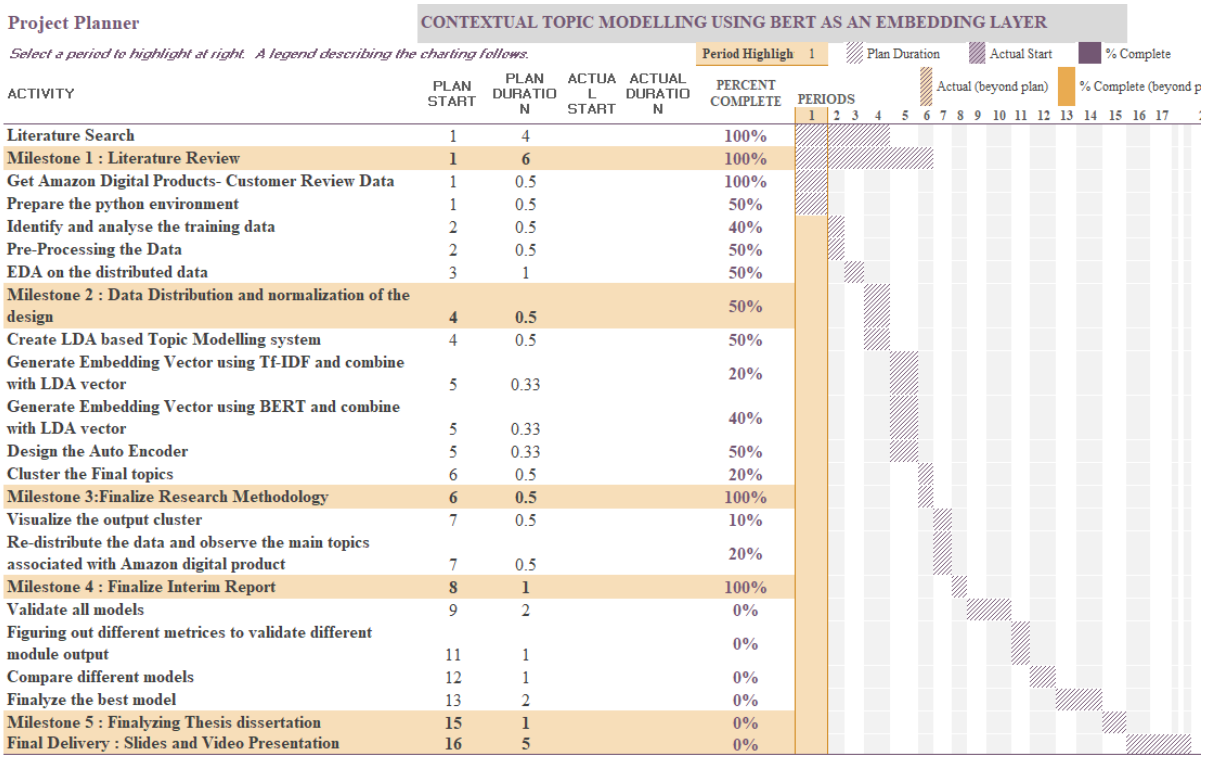


Figure A.1: Research timeline and milestones

A-1 Risk in the Study: This study requires a huge corpus of data pre-processing and the entire process depends on a broad spectrum of compute-intensive and mission-critical applications which completely depends only on GPU-computing unit.
Hence GPU down-time and limitation of allocated GPU cloud space are major risk factors associated with the study.

A-2 Mitigation plan:

- Focus on core utilization of sentence embedding techniques and Auto Encoder Part. Once few objectives are met then deep dive into furthermore as an iterative process.
- Prepare different Data Pre-processing module and make it a one-time processing.

A-3 Contingencies:

- If Google BERT instigate huge implementation complexity, consider different configuration class (like, huggingface) to store the configuration of a BertMode and instantiate the BERT model as per the required arguments.
- Para rally focus should be on the word embedding techniques along with Document embedding.

As per the above-mentioned plan and considering the risk and contingency plan research has been completed in the due time.

APPENDIX B: RESEARCH PROPOSAL

1. Introduction

Amazon is known for its disruption of well-established industries through technological innovation and mass scale (Wikipedia, 2020a).
Product reviews are a very essential tool for Amazon's customers to decide whether to buy the product.
Natural language processing (NLP) is a subfield of explainable AI associated with linguistics and computer science. It focuses on the interactions between computers and human language. Typically it is a research area to study how computers learn to process and analyse large amounts of natural language (Wikipedia, 2020b)
Topic Modelling is one such NLP task which is largely used in real-life business problem like sentiment analysis of customers, analysing the trending topics among them about the newly launched product, as a solution of explainable AI.
In this research, we will be focusing on a new state of the art for topic modelling incorporating BERT as a sentence embedding layer.
Unlike recent language representation models (Clark et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabelled text which is being trained with information from both the left and the right side of a token's with the idea of Masked Language Model(MLM) and multilayer Self-Attention mechanism. It is modelled in such a generic way so that it can be used in different NLP related tasks (Transfer Learning).
We will focus on combining the BERT sentence embedded vector with traditional Bayesian technique LDA to understand the context of the customer reviews more precisely.

2. Background and Related Research

2.1 Introduction:

The essence of the Topic modelling is that context of our document is actually hidden, or "latent," which we cannot observe initially. The sole purpose of this task is to identify the latent topics that give the actual context of our document and corpus.
In a huge corpus containing a ton of document. The underlying latent topics will come out through the process of topic modelling.
A naïve approach is to list the keywords as per their frequency of occurrence, which is termed as TF. In this approach, the actual topic may not be in the top list of keywords, and the topic is hidden which is not feasible to derive from the text of the document (Tokunaga and Iwayama, 1994).
Topic modelling is a computational technique to find the hidden patterns of co-occurrence in the set of documents (Tokunaga and Iwayama, 1994).
TM is one of the most important research area in the field of information retrieval (Onan et al., 2016).

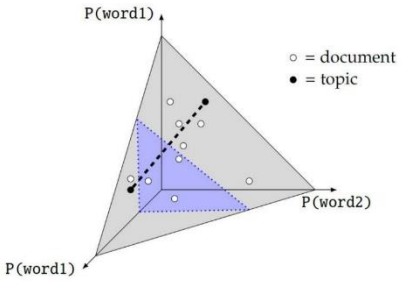


Figure :1 Geometric representation of topic modelling (Emmery, 2014)

The above diagram depicts the probability of identifying a certain word. Each document can be represented as a white point. Each word is associated with some weight in the document. Hence topics can also present in the space with black points. Generally using this space, a linear classification problem can be solved and a decision boundary can be found that divides the documents into different topics.
LDA is one of the popular topic modelling techniques.
In this hierarchical Bayesian method probability distribution of words represent topics and a probability distribution over topics represents documents. (Steyvers and Griffiths, 2010).
The main categories in probabilistic topic modelling methods are:
Inter and Intra document correlation, supervised and temporal probabilistic model and, basic traditional methods (Daud et al., 2010).
Most formative works in the topic modelling research area are LSA, PLSA, and LDA(Alghamdi and Alfalqi, 2015).

2.2 LSA: Here the document-term matrix will be decomposed into two matrices: document-topic and topic-term .LSA is based on SVD. It reduces the dimensionality of the matrix.
It is a frequency-based model. In LSA, we take a noisy higher dimensional vector of a word and project it onto a lower dimensional space. The lower dimensional space is a much richer representation of the semantics of the word.
However, SA has some drawbacks as well. One is that the resulting dimensions are not interpretable (the typical disadvantage of any matrix factorisation-based technique such as PCA). Also, LSA cannot deal with issues such as polysemy. For e.g. A single word can have many senses, and the representation of the term in the lower dimensional space will represent some sort of an 'average meaning' of the term rather than three different meanings.

2.3 PLSA: PLSA is an advanced development on the top of LSA to overcome few disadvantages (Alghamdi and Alfalqi, 2015).The main objective of PLSA is to identify contexts of used words in the document without referring any dictionary.

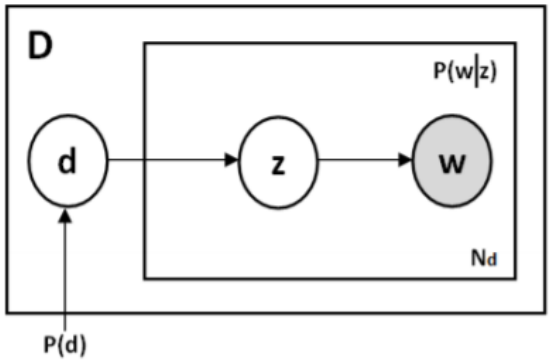


Figure 2: Plate Notation of PLSA (Alghamdi and Alfalqi, 2015)

1) Probability of selecting a document di, is P(di)
2) zk, is the latent class. The probability for a given document P(zk|di)
3) wj is the word that will be generated for a given latent class. The probability will P(wj |zk)

PLSA disambiguate the polysemy .It uncovers similar topics .It basically groups words together that share a common context (Hofmann, 2001)

2.4 LDA: Latent Dirichlet Allocation is a combination of **pLSA** and the **Bayesian techniques.** it uses Dirichlet Prior Process for the distribution of document-topic and word-topic for better generalization.

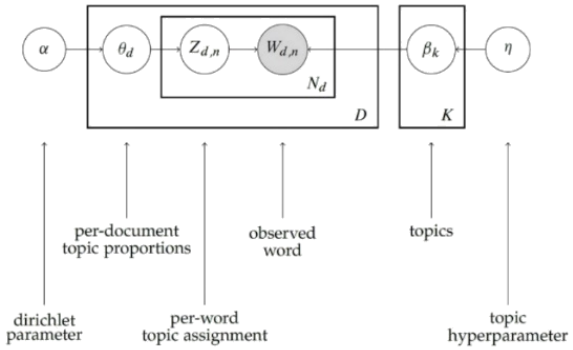


Figure 3: Plate Notation of LDA (Blei and Lafferty, 2009)

N = Number of words
M = Number of documents
θ = Topic selecting dice
Z = Selected Topic
W = Selected word by topic's dice
K = Number of topics
α = K dimensional vector that defines how K topics are distributed across documents
V = Number of vocabulary words
β= V-dimensional vector that defines how V words are associated with topics
LDA hardly considers the sequence of the words in the document. Usually, LDA uses the traditional BOW techniques. When the topic modelling task becomes more complex it is very important to identify the underlying meaning of the text at every level (word, paragraph, document).
Typically, something like word2vec is been used for the vector representation of the words. lda2vec is a combination of word2vec and LDA which has been built using skip-gram model to generate topic, document and word vectors.

3. Problem Statement

This section will start with an overview of related works in the field of Deep NLP and finally will explain the problem statement that the study will focus on.
LDA of TF-IDF is sufficient for identifying topics in the coherent texts when they are able to find the most frequent words.
However, when the choice of the words and the meaning of the sentences are incoherent, extra contextual information is required to represent the idea of the texts(Shao, n.d.).
Adding more contextual knowledge to the model improves the coherence. The recent development of topic models based on neural networks are gaining huge attention, while BERT-based models are pushing the general neural models as state of the art(Bianchi et al., 2020).

3.1 Word Embedding: In word embedding, we would generate an embedding for each word in the set. The simplest method would be just like one-hot encoding for the sequence of words. Word embedding not only converts the word but also identifies the semantics and syntaxes of the word to build a vector representation of the information. However, we miss the entire context of the sentence and the document.

3.2 Sentence Embedding:
The new state of the art for understanding sequential text is Sentence embedding.
Sentence embedding techniques represent entire sentences and their semantic information as vectors. This helps the machine in understanding the context, intention, and other nuances in the entire text.
BERT representations have been largely used by the research community in a diverse set of NLP applications (Rogers et al., 2020).
Currently, BERT is the most effective solution for sentence embedding.

3.3 Doc2Vec Embedding:
Doc2Vec embedding is an extension of the Word2Vec model incorporating 'paragraph vector' at the document level. It is an unsupervised model.
There are two main ways of applying this technique:

PV-DM (Distributed Memory version of Paragraph Vector): Here, the model predicts the next word given a set of words.
Here the word vectors are being shared among all the sentences and a paragraph vector to the sentences. Then the paragraph and word vectors are combined to get the final sentence representation.

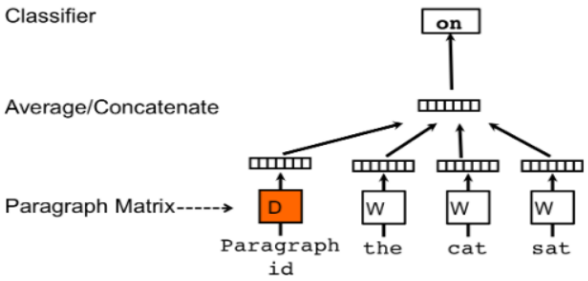


Figure 4: PV-DM framework (Le and Mikolov, 2015)

PV-DBOW (Distributed Bag of Words version of Paragraph Vector): This is based on the skip-gram model. Here the model predicts the source(sentence) of the word by sampling random word across the documents

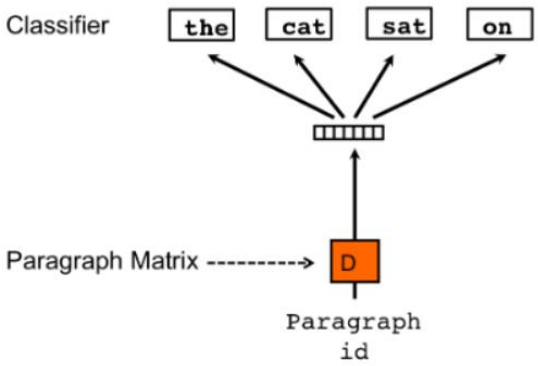


Figure 5: PV-DBOW version of paragraph vectors(Le and Mikolov, 2015)

3.4 BERT (Bidirectional Encoder Representations from Transformers.):

BERT is basically a pre-trained deep bidirectional representation from unlabelled text by training from both directions of the text. BERT can be tuned incorporating different output layer and leverage to a new state of the art models for various NLP related use cases. There is no need to build any specific model from scratch to achieve any particular task (Devlin et al., 2019).

The fundamental pillars of BERT are Transformers and Self Attention Mechanism.

The architecture of BERT depends on Google Transformers. It is basically a bidirectional neural network model.

BERT is developed by the bidirectional training of the Transformer for language modelling. A novel technique is integrated in BERT named Masked Language Modelling to train the model bi-directionally which was not possible previously

Bert behaves as a Super Encoder for various sequences to sequence model.

Bert uses the below-mentioned approach for pre-training (Transfer Learning):

- Data: It is trained on BookCorpus(800M words) and English Wikipedia (2500M words). A huge document-level corpus to have a long continuous sequence is needed.
- Basic two Tasks: Predicting a word and Next sentence Prediction
- Time required for pre-training: Around 3 days on 16 TPU.

Google Brain tried to implement two type of BERT:

BERT_base:L=12,H=768,A=12

parameters: 110M

Bert_large: L=24,H=1024,A=16

Parameters=340M

L: Number of Encoder Layers

H: Hidden size ("Embedding Dim")

A: Number of Self-attention head

3.4.1 Transformers: Transformer is developed with an idea called attention mechanisms. However, it has not dispensed the recurrence and convolutions entirely. It shows extremely high performance in different machine translation tasks mainly because of its parallel processing techniques which significantly reduced the training time as well (Vaswani et al., 2017).

RNN, LSTM (Hochreiter and Urgen Schmidhuber, 1997) and gated RNN (Burk, 1999) neural networks are an established and one of the most popular state of the art approaches in sequence to sequence learning and problems like language modelling and machine translation (Burk, 1999).

Attention mechanisms plays a pivotal role in different tasks such as sequence modelling and transduction models. It permits modelling of dependencies regardless of their separation in the input or output sequences (Kim et al., 2017).

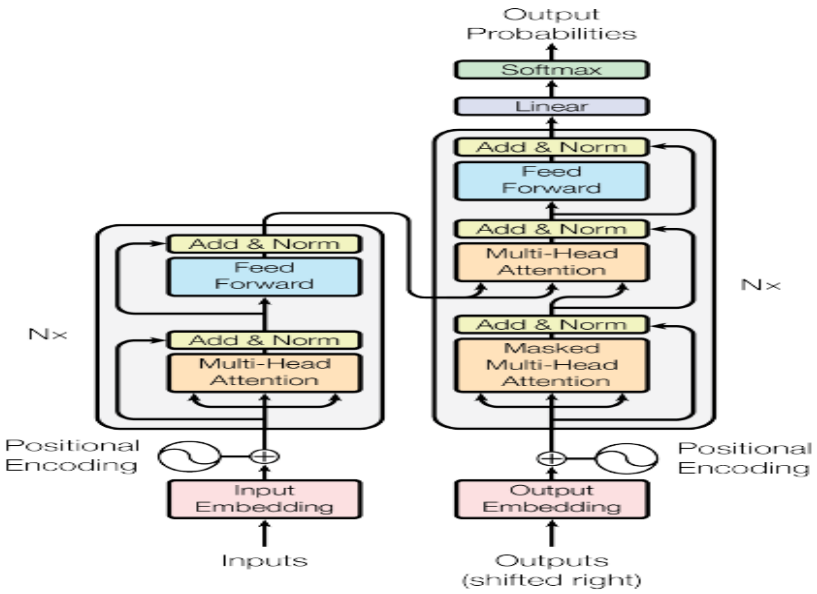


Figure 6: The Transformer - model architecture (Vaswani et al., 2017)

Initially, the sequence to sequence model was developed using RNN with gated LSTM and it was improved by the attention mechanism. The attention mechanism has added global behaviour to the decoding phase.

However, in sequential processing, RNN loses information for long sentence.

Hence Google's Transformer only uses the attention mechanism to encode and decode sequence and comfortably gets rid of the RNNS.

The input of the encoder is the whole sequence or the sentence that is important to keep the Global processing aspect.

We can see "Attention" as a memory of the network which preserves the hidden states of the model, and the model fetches that information from the memory to get the contextual information

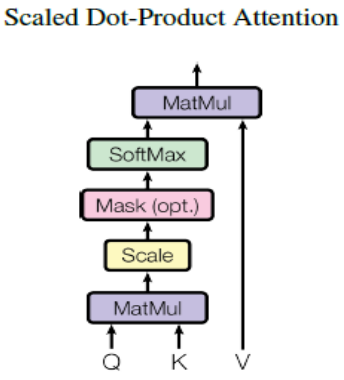


Figure 7:Attention Dot-Product (Vaswani et al., 2017)

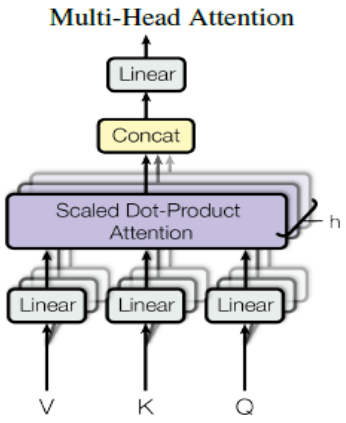


Figure 8: Multi-Head Attention (Vaswani et al., 2017)

In three different ways multi-head attention can be used by the Transformer:
1. The queries from the previous decoder layer, and the memory keys and values from the output of the encoder goes into the encoder-decoder attention
2. All the keys, values and queries from the same place come to the self-attention layer of the encoder.
3.The decoder has a self-attention layer. It allows each position to attend to all end to end positions in the decoder including itself.

The Transformer, deliberately avoid using of recurrence and instead rely entirely on an attention mechanism to identify dependencies between input and output. The Transformer permits significant parallelization in the entire process (Vaswani et al., 2017).

Amazon receives thousands of orders daily, and in-turns receives thousands of customer reviews every day.
Considering their digital product, customers generally talk about so many different topics in the same review.
Customers can have different concerns like server issues, hacker issues, processing speed, product aesthetics, budget, display and screen size, customized App performance in those products, issues related to integrated hardware like Bluetooth, memory drive, etc.
In this research work to understand the underlying context-based concerns of customers or their appreciation about the product, document level embedment using a Pre-trained model along with classical TM technique will be experimented to enhance the latent topic identification task.

4. Research Questions

A comprehensive literature review indicates there have been multiple research works done in the field of Text mining, specifically in Topic Modelling. However, most of the Topic modelling techniques are developed using Probabilistic Latent Semantic Analysis like PLSA or its Bayesian version like LDA.
There is no research work found on Topic Modelling on the customer reviews of Amazon's most successful Digital products incorporating new state of the art NLP model BERT as an embedding layer.
The literature review leads the below-mentioned research questions, which will be addressed in this study:

- What are the latest topics of discussion among the customer reviews on Amazon's Digital products?
- The reviews don’t consist of one single topic. How Bayesian Topic modelling techniques work in highly sparsed data since customer’s review mostly incoherent?
- How interpretation of the reviews depends highly on the context?
- What are the performance evaluation criteria of the Bayesian methods like LDA?
- What potential embedding techniques can enhance the performance of the topic identification task?
- How incorporation of BERT with self-attention mechanism along with a traditional Bayesian Technique can enhance the performance?

5. Aim and Objectives

The aim of this study is to propose an approach to enhance the contextual topic identification capabilities of Bayesian method-based models on a large incoherent corpus of documents. The goal of this study is to identify the hidden topic of customer reviews of Amazon's digital products.

Considering the aim of the study the research objectives are articulated as follows:

- To analyse the topic identification capability of traditional Bayesian methods when customer review data is hugely sparsed.
- To investigate the potential of document embedding techniques and pre-trained NLP model BERT to understand the underlying idea of the review.
- To develop an approach which combines the sentence embedding and Bayesian techniques to get more accurate context-based topic discussed among the customers.

6. Scope and Limitations of the Study

The scope of the research is limited to the following factors:

- The research focuses on the development and evaluation of different Deep NLP models incorporating a new state of the art module, BERT as a sentence embedding layer.
- The scope can also include parallelizing the Latent Dirichlet Allocation using all CPU cores to parallelize and speed up training of the model.

Considering the high computation requirements and the fixed time frame below-mentioned limitations are set on this study:

- The data for the research is directly taken from the Datafinite repository.The collection or extraction of raw data is not under the scope of the study.
- The research does not have much opportunity to compare the final model performance with the models based on word vector, lda2vec.
- BERT training from scratch is out of the scope due to the resource constraints. Hence, research will focus on pre-trained BERT by tuning the hyperparameter to get the optimum solution.
- Use of different embedding layers with other versions of BERT, SciBERT, or BART is out of the scope of this study.
- Building Customized reports for different products which could give some more interesting information.

7. Significance of Study

In this digital era, real-world data on the web is highly incoherent, especially when it is a part of human conversation or communication.

In a fixed amount of time or the same paragraph or document or any specific universe, there can be different topics of discussion which are latently connected and hidden. The research can open up a new direction in the field of Topic Identification tasks when the documents are incoherent and abrupt.

This study will consider the approach which resembles the humanized way of understanding the text(using techniques like Self Attention, Masked language modeling) to understand the Amazon's Digital product's customers review when it is incomprehensible, unclear, or even confusing. Hence it will be helpful for the E-commerce websites to assess the sentiment of their customers and leverage it to a very engaging and intuitive recommendation system as well.

8. Research Methodology

8.1 Introduction

The broad objective of this research work is to do a scientific study to solve a practical problem and unlock the potential of digital content, the amazon customer review, and empower and enhance online engagement between customers and different stakeholders. Hence it can be categorized as Applied Research.
It focuses on the study of AI explainability of contextual Topic Modelling to identify groups that are semantically similar and assign the most appropriate category tags of the reviews incorporating embedding techniques with traditional Topic modelling technique, LDA. Therefore, we can categorize it as Qualitative Research as well.
The entire study based on the combination of Basic and qualitative applied research, where the basic research retains a core position within the research mind set (Bentley et al., 2015) and qualitative applied research focus on the solving the real business problem .

8.2 Dataset Description

Datafiniti transforms unstructured data from web to usable format and gives instant access to it. Their master repository consists of data from thousands of websites which leveraged a standardized database of different consumer product, business strategy and property information (Datafiniti(Kaggle), n.d.).
The dataset used for this study is a list of more than 30,000 consumer reviews of different Amazon digital products like the Home Theatre, mobile, computer, media player, etc (Datafiniti(Kaggle), n.d.).
The study will be utilizing this data of customer reviews of Amazon's electronics product and will discover the contextual topic of their discussion.

Data Dictionary			
#	**Field Name**	**Field Type**	**Field Description**
2.	Name	Text	The product's name.
3.	Brand	Text	The brand of the Product
4.	Category	Text	Category key of products from multiple sources.
6	manufacturer	Text	The manufacturer of this product.
7	reviews.date	Date	Date of the customer reviews
10	Reviews. doRecommend	Boolean	If recommended by the reviewer
12	reviews.Helpful_num	Integer	The number of visitor found this review helpful
13	Review_rating	Float	1 to 5 (5 is the best).
14	Source_URLs	Keyword	URLs where this reviews were seen.
15	Reviews_text	Text	text of the review or comment
16	Reviews_header	Text	Title of the review.
17	reviews_user_City	Text	City of the reviewer's
19	reviews.user_name	Text	Username of reviewer

Table 1: Amazon Digital product review dataset(Datafiniti(Kaggle), n.d.)

8.3 Process Flow

- Fetch Customer reviews of Amazon's most successful consumer electronics products (Data Source: Datafiniti)
- Carefully pre-process the raw data before feeding it into the model using different patterns-search filters for normalization.

- Analyse the distribution of the data and improve the data pre-processing to have a more normalized design.
- For identifying the main topics in the customer reviews, we can follow two types of model:
    1. Bayesian models (LDA)
    2. Embed documents into vector to identify their similarities in the vector space by clustering.
- The Amazon digital product review data are highly sparsed and it does not coherently discuss one single topic. Hence it will be difficult for BOW based model like LDA to identify the underlying context-based topic of the reviews.
- We will be applying parallel embedding techniques to embed the full contents which to be clustered with similar topics.
    1. TF-IDF Vector representation of the document.TF-IDF is a BOW based (disregarding grammar and word order). Hence mostly it won't be able to capture the contextual information because of the incoherent and unstructured data.
    2. BERT: Use BERT as a Sentence Embedding layer on the document to capture the contextual information.

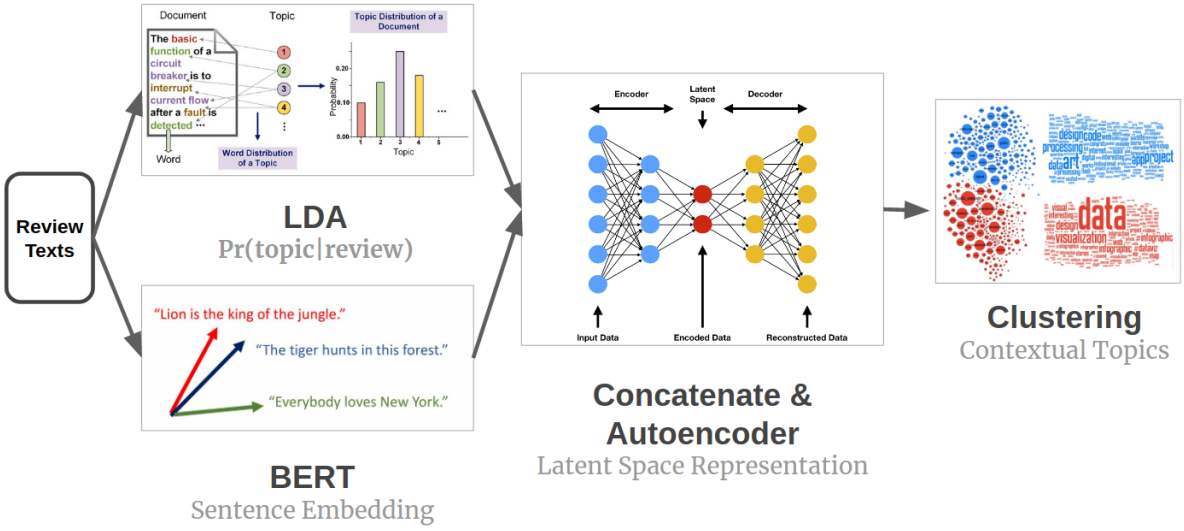


Figure 9: Contextual Topic Identification model design

- Prepare two vectors:
  1. Probabilistic topic assignment vector by LDA: Use the LDA model to figure out topics by identifying the occurrence of frequent words in the incoherent text.
  2. Sentence embedding vector by BERT: When the used words and the meaning of the sentence in the reviews are incoherent, extra semantic information is required to find the topic of the texts more precisely.
- Combine both the LDA and BERT vector to identify the semantic information and contextual information by tuning relevant hyperparameters to get the optimum information from different sources.
- The resultant vector will be containing highly sparsed information in the high-dimensional space.

The Auto Encoder plays a pivotal role in the study
Auto Encoder is developed on the top of the Neural Network architecture to learn the surfaced lower-dimensional features of the input data.
Autoencoders consists of 4 parts:
Encoder: The model reduces the input dimension by compressing the input data into an encoded representation.
Bottleneck: It contains the compressed form of data with the lowest possible dimension
Decoder: It takes the compressed data as input and reconstruct it to match the input data as much as it is possible.
Reconstruction Loss: This method evaluates the performance of the decoder.

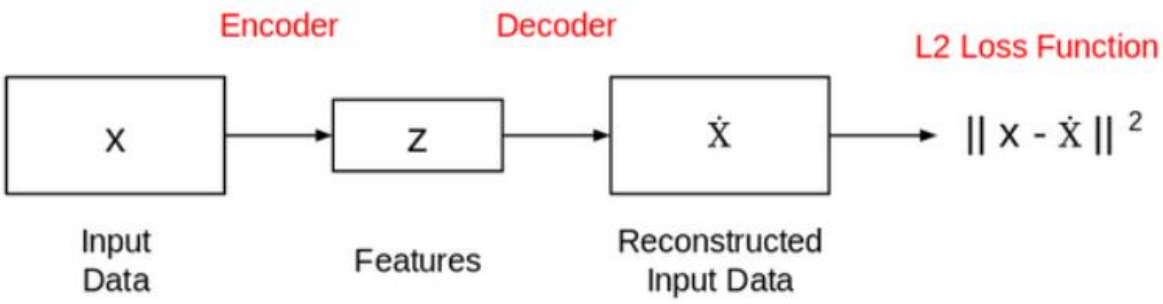


Figure 10: Auto Encoder Network (Le and Mikolov, 2015)

This network is trained to reconstruct the original input by using the features,z.
The L2 loss function indicates the difference between the original input and the decoded output data.
In the topic modelling tasks, the input of the encoder is the whole sequence or the sentence that is important to keep the Global processing aspect.

- An autoencoder will be used to surface the concatenated vector to the lower dimension (Latent Space Representation).
- Once we have the lower dimension representation with more condensed information, clustering techniques will be applied to get the context-based topic.

9. Expected Results

The entire research work can be segmented into three major parts.
The first part is the identification of the latent topics of the review which will be done by hierarchical Bayesian Model, LDA. The study will be using the most popular Gensim coherence metrics like c_v and u_mass to evaluate the model.

Coherence Value is based on a sliding-window technique, a one-set segmentation of the words and a validation measure that uses NPMI and the similarity function (Cosine).
The plot of the Coherence score vs the number of topic graphs can give some important insight.

The cv and umass give the coherence score which gives an idea of the interpretability of the topics.
Next, the study will focus on the comparison of the results of the topic modelling using LDA with the different methods, like a combination of TF-IDF, clustering and models like a combination of BERT, LDA and Clustering.
Thereafter we will be using the Silhouette score in our final model to measure the consistency within the cluster. The technique provides a succinct graphical representation of how well each object has been classified.
The study will include visualizing the different clusters depicting the main topics, what most of the customers have concerns about. The resultant performance improvement in the topic identification task by incorporating BERT will be the most probable expected result.

10. Required Resources

10.1 Hardware Requirements

For this study, the computation of the experiments will be performed on a Windows system with the below configurations:

- OS Name: Microsoft Windows 10 Home Single Language
- Version:10.0.18362 Build 18362
- Processor: Intel Core i5-8250U CPU
- Installed RAM :8.00 GB
- Total Virtual Memory:15.4 GB

10.2 Software Requirements

In this study, the ML and DL algorithm will be implemented in the Python framework.
The data pre-processing, model building, and fetching insights will be done by using several open-source libraries available in the Python framework.

- Programming Language: Python 3.7
  Open source libraries:
  - sentence-transformers: Sentence Embedding with BERT
  - spacy-langdetect - language detection capabilities
  - language-detector- language detection capabilities
  - gensim - Unsupervised topic modelling
  - symspellpy- Python port of SymSpell v6.5, which provides much higher speed and lower memory consumption
  - sklearn -machine learning library
  - NLTK - Natural Language Toolkit
  - Numpy - Data pre-processing, Mathematical function
  - Pandas - Data pre-processing and manipulation
  - Matplotlib (Seaborn) - Visualization

10.3 Cloud provider for building AI project (GPU/TPU):

- NimbleBox