Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Datateknik
2021 | LIU-IDA/STAT-A--21/003--SE

Semantic Topic Modeling and Trend Analysis

Jasleen Kaur Mann

Supervisor : Johan Alenlöv
Examiner : Jose M. Peña

External supervisor : Anders Arpteg and Mattias Lindahl

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Abstract

This thesis focuses on finding an end-to-end unsupervised solution to solve a two-step problem of extracting semantically meaningful topics and trend analysis of these topics from a large temporal text corpus. To achieve this, the focus is on using the latest developments in Natural Language Processing (NLP) related to pre-trained language models like Google's Bidirectional Encoder Representations for Transformers (BERT) and other BERT based models. These transformer-based pre-trained language models provide word and sentence embeddings based on the context of the words. The results are then compared with traditional machine learning techniques for topic modeling. This is done to evaluate if the quality of topic models has improved and how dependent the techniques are on manually defined model hyperparameters and data preprocessing. These topic models provide a good mechanism for summarizing and organizing a large text corpus and give an overview of how the topics evolve with time. In the context of research publications or scientific journals, such analysis of the corpus can give an overview of research/scientific interest areas and how these interests have evolved over the years.

The dataset used for this thesis is research articles and papers from a journal, namely 'Journal of Cleaner Productions'. This journal has more than 24000 research articles at the time of working on this project. We started with implementing Latent Dirichlet Allocation (LDA) topic modeling. In the next step, we implemented LDA along with document clustering to get topics within these clusters. This gave us an idea of the dataset and also gave us a benchmark. After having some base results, we explored transformer-based contextual word and sentence embeddings to evaluate if this leads to more meaningful, contextual, and semantic topics. For document clustering, we have used K-means clustering. In this thesis, we also discuss methods to optimally visualize the topics and the trend changes of these topics over the years.

Finally, we conclude with a method for leveraging contextual embeddings using BERT and Sentence-BERT to solve this problem and achieve semantically meaningful topics. We also discuss the results from traditional machine learning techniques and their limitations.

Acknowledgments

I am incredibly grateful to Anders Arpteg at Peltarion, and Mattias Lindahl at Linköping University for granting me this opportunity to work on this thesis and having faith in me. Thank you for your guidance, suggestions, advice, and support throughout.

I would like to sincerely thank my supervisor Johan Alenlöv at Linköping University, for the constant guidance and valuable feedback throughout the thesis.

I would also like to sincerely thank my examiner Jose M. Peña and my opponent Saman Zahid for sharing valuable, constructive feedback during the revision meeting.

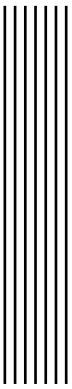
I am also very grateful to my friends at Linköping University Maria, Aashana, Sridhar, Mathew, Brian, and Naveen for being my support system and for making my master's programme memorable. Special thanks to Mathew for sharing this thesis opportunity with me.

Above all, I would like to say a heartfelt thank you to my parents and siblings. To my best friend, Smriti, for always believing in me. To my dear husband, Amit, for being my constant source of encouragement and motivation. I would not have been able to reach here without your love and support!

Contents

List of Figures

List of Tables



Glossary

APIs Application Programming Interface.

BERT Bidirectional Encoder Representations for Transformers.

GPU Graphics Processing Units.

JOCP Journal Of Cleaner Productions.

LDA Latent Dirichlet Allocation.

LSTM Long Short-Term Memory.

MLM Masked Language Modeling.

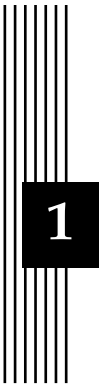
NLP Natural Language Processing.

NSP Next Sentence Prediction.

pLSA Probabilistic Latent Semantic Analysis.

Sentence-BERT Sentence Embeddings using Siamese BERT-Networks.

TF-IDF Term Frequency-Inverse Document Frequency.



1 Introduction

1.1 Motivation

As digital data is increasing and becoming an asset in terms of the information it carries, so is the temporal textual data. In addition to new temporal textual data, the historical data is also being converted to digital format (as in the case with the dataset used in this thesis [1]) and made available at an increasing rate with advancements in storage capabilities and cloud technologies. When we think about information retrieval from textual data, we often think about sentiment analysis from social media comments or product reviews or Twitter comments or figure out what is trending from news posts and comments? Another often ignored fast increasing textual data available in digital format is research papers, journals, scientific articles, e.g., the journal that is a dataset in this project [1]. This journal was getting about 1200 publications in an issue per year until 2015 and had more than 5000 publications for the year 2020 by the month of September. As the temporal textual data in digital format is increasing at a fast rate, there is an increased need for an end-to-end solution that can provide an overview of the data in terms of topics and give a trend analysis of these topics over the years.

This project's primary motivation is to develop a statistical and machine learning model that can find meaningful topics given a large text corpus in an unsupervised manner and identify the trend of these topics over a period of time. The topics need to be meaningful and human-understandable in nature and not mere keywords. The traditional topic modeling techniques like LDA required many hyperparameter tuning and need to know how many topics exist in the corpus apriori. Also, the topics are not semantically meaningful. So, another main motivation of the project is to explore the latest developments in pre-trained word embeddings and leverage them to find semantically meaningful topics. Once the topics are extracted, the next step is to identify a suitable approach to visualize the trend of topics over time.

Such a solution can help summarize and organize a large set of textual documents and data. It will also make searching or finding clusters of similar documents in the corpus a lot easier and more efficient. Since the data carries temporal information, it can be leveraged to identify emerging topics and track the evolution of topics over time. Since the information is available for historical documents as well, such a solution can also provide a way to analyze historical trends in publications. It can offer many other domain-specific insights as well.

For achieving this, the focus is on using the latest developments in the field of NLP and leverage the pre-trained contextual knowledge of transformer-based language models like Google's BERT [2] for achieving the desired goal.

1.2 Aim

The aims of this thesis project can be divided into the following two main groups:

- Unsupervised semantic topic modeling:
  - Identify semantically meaningful topics from the research papers.
  - Understand the topics contextually, i.e., human-understandable and interpretable topics.
  - Identify an approach to visualize topics.
- Topic trend analysis:
  - Identify and visualize changes in topics over the years.
  - Identify emerging trends in the topics.
  - Understand the relation and similarity between evolving topics.

By meaningful topics, we mean topics that make sense and are human-understandable as a collection of words. This project's main aim is to find such topics in an unsupervised way in a large text corpus and then find a way to visualize the trends and changes in these topics over time that were found in the first step and analyze the results.

Another objective that is of importance in this project is to find a way to evaluate the results from traditional statistical and machine learning topic modeling techniques and methods based on transformer-based pre-trained language models to extract topics. And compare the results from these approaches and understand the differences in algorithms, which leads to different results.

1.3 Research Questions

This project focuses on answering the following research questions:

1. Is it possible to find human interpretable topics from a large text corpus using traditional statistical and machine learning approaches and get meaningful trend analysis of these topics over time?
2. Can the latest pre-trained language models (e.g., Google's BERT and it's variants) be leveraged to find semantic topics? What kind of impact does having/including the word/sentence contextual embeddings from the transformer-based large pre-trained language models have on the quality of topic models? Does leveraging the knowledge of these models improve the quality of extracted topics or not?
3. Can the results from these methodologies be compared with each other to decide an optimal approach? Can some conclusion be reached on comparative analysis of results from these two approaches?
4. What could be the solution for visualizing topics and visualizing the trend changes in these topics? How to analyze results given the large temporal textual data?

1.4 Related Work

To get an overview of large textual data, the most common technique used in machine learning is topic modeling. Blei et al. [3] discuss probabilistic topic models, mainly LDA and the use of such models for summarizing and understanding the ever-growing digitized archive of information. Vayansky et al. [4] further discussed in detail various probabilistic topic modeling techniques and also the limitations of the most popular topic modeling technique i.e. LDA.

For handling large text corpora effectively, Venkatesaramani et al. [5] suggest clustering the documents based on their similarity and then attempting to find topics for each cluster.

Regarding data preprocessing, Schofield et al. [6] observe that for LDA, except extremely frequent stopwords, removal of stopwords has little impact on the quality of topic models and stemming can have an adverse impact as topic model inference often places words sharing morphological roots in the same topics.

Guo et al. [7] discuss that the data-driven approach of LDA has some limitations. For infrequent word, LDA cannot learn its correct semantics from the observed distribution and will assign it the dominant document topic.

There is a lot of research going on regarding using the latest advancements in the field of NLP, which is presently mostly being used for either classification or question-answering tasks, to leverage it for summarization and topic modeling tasks. Deng et al. [8] have proposed and developed an innovative semi-supervised learning approach by utilizing deep learning and topic modeling to have a better understanding of the customer's voice from textual reviews. This approach combines a BERT based multi-classification algorithm with a novel Probabilistic and Semantic Hybrid Topic Inference (PSHTI) Model. It aims at automating the process of identifying main topics and sub-topics from the textual feedback and support data. Dieng et al. [9] also discuss an approach where they use traditional machine learning approach for topic modeling with word embeddings and this method outperforms traditional machine learning approach based on LDA. Newman et al. [10] provide an analysis of the ways in which topics can be flawed and, more importantly, an automated evaluation metric for identifying such topics that do not rely on human annotators or reference collections outside the training data. Sahrawat et al. [11] use BERT embeddings for keyphrase extraction from scholarly articles.

For trend analysis of temporal textual data, there are few research papers that discuss the approach to find trends and visualize them. Hall et al. [12] discussed the trend visualization for LDA topic models.

In case of topic models, evaluating the topic quality is a challenging task. Especially finding a statistical methodology to achieve this. Chang et al. [13] discuss the challenges of evaluating the human interpretability of topic models and propose a quantitative method for measuring semantic meaning in inferred topics. Lee et al. [14] also tries to better understand how non-expert users understand, assess, and refine topics.

2 Data

2.1 Journal Of Cleaner Productions

This project's data comes from JOCP [1], an international, transdisciplinary journal focusing on cleaner production, environmental, and sustainability research and practice. The journal was started in 1993 and had 24000+ papers at the time of working on this project. Data can be accessed and downloaded using APIs provided by Science Direct. For this project, Python script was written to extract the title, abstract, published date, and full-text of the research papers.

For the purpose of this project, the abstract text of research papers is being used for topic modeling and the time information like the published date is being used for the trend analysis. As the abstract captures a paper's summary, it can be a fair representation of the topics included in the paper. The keywords information from the journal has intentionally not been used as it is manually added and might not capture all the topics being discussed in the paper. The title of the paper is also not used as it will only reiterate the topics already included in the abstract. Figure [2.2] shares an example of one of the entry in the dataset of this project. These details are of a published paper, which includes the ID, published date, the text of the abstract and keywords by the author of the paper. In the dataset, a unique ID associated with every article is available along with the published year, text of the abstract of the paper and manually mentioned keywords for the paper by the author of the paper.

The number of publications in the journal has been increasing at a very high rate in the past five years and this project makes even more sense with this increase as it is almost impossible to summarize or find a trend manually through this corpus. To get an idea of the size and type of the data set, Figure [2.1] shows the number of publications per year's Issue. Figure [2.3] is the distribution of the number of words per abstract of the document.

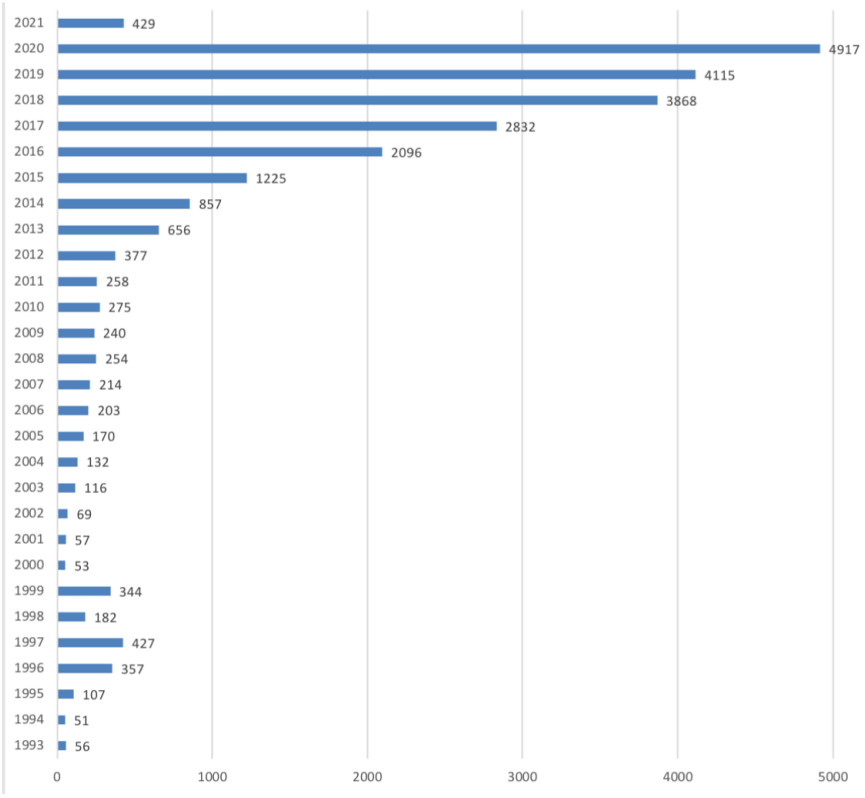


Figure 2.1: Number of publications in an Issue of the Journal/Year

ID	Year	Abstract	Keywords
S0959652619320372	2019	In-situ gasification chemical looping combustion (iG-CLC), which has been tested on pilot plant level, is regarded as an advanced carbon capture and storage (CCS) technology for reducing CO2 emissions. A life cycle global warming impact (GWI) analysis is performed to consider the lifetime emissions of the low-carbon iG-CLC technology. Herein, the capacity is considered to be 610 MWe using natural ilmenite as oxygen carrier and steam as gasification agent. At the condition of operational pressure of 15 and air reactor temperature of 1050 °C, the net power efficiency of 37.7% for achieving 93.5% inherent CO2 capture is obtained in simulations with thermodynamically optimum condition. The life cycle GWI is calculated equal to be 160.3 kg CO2-equivalent/MW h. The effects of several essential parameters, including steam to carbon ratio (S/C), oxygen carrier to fuel ratio (ϕ), different oxygen carriers and lifetime of oxygen carriers, on the lifecycle GWI have been analyzed and discussed to meet the potential possibility for further reducing greenhouse gas (GHG) emissions. To obtain sufficient carbon capture efficiency, the S/C ratio and ϕ are suggested to be 1.3 and 1.2 in this study, respectively. The life cycle GWI is heavily dependent on lifetime of ferrum (Fe) when it is less than 2000 h, beyond that range, the GWI is decreasing, but very slowly.	Global warming impact; CO2 capture; Chemical looping combustion; CCS;

Figure 2.2: Data from one of the published papers in JOCP available in the dataset. [1]

[1] https://www.sciencedirect.com/science/article/abs/pii/S0959652619320372

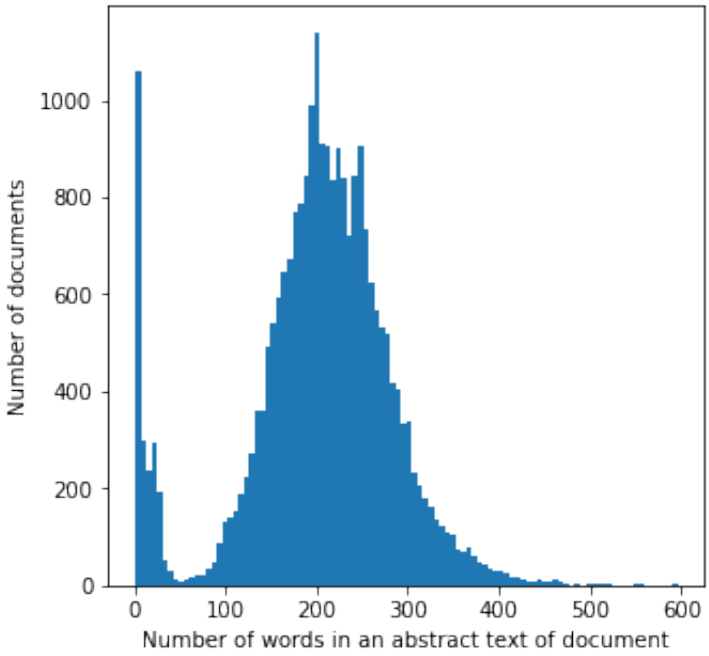


Figure 2.3: Number of words per document abstract. Average length of abstract text in this dataset is around 202 words.

3 Theory

This chapter, contains the theoretical background for the concepts and methods used in the thesis. As quite a few different concepts are explained in this chapter, the variable names and other notations are consolidated in the Table [3.1]. This table is referenced throughout this chapter to avoid confusion caused by variable names, labels and other notations.

M	set of documents
m	a document in set of documents M
K	set of topics
k	a topic in set of topics K
N_m	total number of words in document m
α	parameter of the Dirichlet prior over topic distribution of a document
β	parameter of the Dirichlet prior over word distribution of a topic
θ_m	topic distribution for document m
ϕ_k	word distribution for topic k
z_{mn}	topic for the n^{th} word in document m
x_{mn}	specific word
T	transformer encoder layers
A, B	vectors
u, v	sentence embeddings

Table 3.1: Notations used in the theory chapter of the thesis

3.1 Topic Modeling

Topic modeling techniques refer to the algorithms which discovers the underlying hidden latent semantic representations in a given collection of documents [3], [15]. Topic models analyze text documents to discover underlying themes (referred to as topics) it contains, and how those themes (topics) are connected to each other [16]. In topic models, a topic is a collection of the most probable words in the cluster. These techniques were initially developed as text-mining tools but have also been used to discover structure in genetic data and images and also have applications in the field of bioinformatics [17] and computer vision [18].

Topic modeling approaches are a form of unsupervised machine learning techniques since the topics, and mixture parameters are not known and are inferred solely from the data. In other words, it is not trained on already tagged or labeled data.

The most commonly used probabilistic topic modeling technique is LDA [19], and another very foundational topic modeling technique is pLSA [20]. Both these models are extensively used for topic modeling and have been modified and extended for many new models. Both LDA and pLSA believe that each document comprises of a mixture of topics, and each topic comprises of a collection of words.

For an example of the topic modeling concept, refer to Figure [3.1] that displays the texts from two different documents from JOCP. These are abstracts of the documents. Table [3.2] demonstrates the topics which were extracted using topic modeling technique of LDA (explained further in this chapter) on these two texts. As it can be seen, these topics provide a very reasonable high level summary of the texts. From topics it can be rightly inferred that texts are talking about sustainable practices for tourism and an environmental tool. The words constituting a topic are in order of the highest probabilities.

Topic 1	assessment technology orware management tool environmental
Topic 2	responsible management neglected people nature article
Topic 3	office management mass significant migration sustainable
Topic 4	modelling life consequences mass matter ota
Topic 5	tourism responsible significant sustainable tour negative

Table 3.2: Topics from topic modeling technique used on the abstract texts in Figure [3.1]

Tourism is currently responsible for the largest, annual human migration in history. This great movement of people has significant positive and negative consequences on nature, societies, cultures and economies. Desired worldwide for its economic benefits, tourism is anticipated to double during the next 20 years, and the multiple consequences of such rapid growth, call for a preventative approach at all strategic and professional levels, in order to avoid negative impacts. Considering mass tourism as a reality of our contemporary life that cannot be neglected by current efforts to endorse sustainable tourism, this paper draws attention to one of its key players—the tour operators—advancing the proposition that they play significant roles in affecting changes in behaviors and attitudes towards more responsible forms of tourism. Aiming to facilitate a constructive debate on the matter, the article presents a few of the most important arguments that underscore the potential that tour operators' have in promoting sustainable tourism.

(a)

This article discusses the ORWARE tool, a model originally developed for environmental systems analysis of waste management systems, and shows its prospect as a tool for environmental technology chain assessment. Different concepts of technology assessment are presented to put ORWARE into context in the discussion that has been going for more than two decades since the establishment of the US Congressional Office of Technology Assessment (OTA). An even-handed assessment is important in different ways such as reproducibility, reliability, credibility, etc. Conventional technology assessment (TA) relied on the judgements and intuition of the assessors. A computer-based tool such as ORWARE provides a basis for transparency and a structured management of input and output data that cover ecological and economic parameters. This permits consistent and coherent technology assessments. Using quantitative analysis as in ORWARE makes comparison and addition of values across chain of technologies easier. We illustrate the application of the model in environmental technology chain assessment through a study of alternative technical systems linking waste management to vehicle fuel production and use. The principles of material and substance flow modelling, life cycle perspective, and graphical modelling featured in ORWARE offer a generic structure for environmentally focused TA of chains and networks of technical processes.

(b)

Figure 3.1: (a) and (b) [1]are abstracts from two different documents in JOCP. Table [3.2] shows the results (topics) using one of the topic modeling techniques (LDA) on these two abstract texts.

[1]https://www.sciencedirect.com/science/article/abs/pii/S0959652604000149
[1]https://www.sciencedirect.com/science/article/abs/pii/S0959652604000940

3.1.1 Latent Dirichlet Allocation (LDA)

Blei et al. [19] introduced LDA in their paper. They described LDA as a generative probabilistic model for a text corpus. It is a three-level hierarchical Bayesian model where each document m, in a given collection of documents M is modeled as a finite mixture over a set of topics K in the collection. And each topic k (consisting of a set of words) is modeled as an infinite mixture over a set of topic probabilities. A topic in LDA is its high probability words and a label (aptly representing the words in the topic) is used to identify the topic. e.g., the topic "tennis players football game match" can be represented by the label "sports" or "game". These words are the highest probability words contributing to the topic and hence, each word will have a probability associated with it. So in our example, this topic can look like "tennis (30%) players (20%) football (20%) game (15%) match (15%)". Figure 3.2 showing the plate diagram of LDA model, gives a visual explanation of LDA's process. In the plate diagram, nodes represents the random variables, lines represents probabilistic dependency, rectangles represents repetitions. The grayed-out variable is an observed variable, and the others are latent variables. Table 3.1 shows the notations used in LDA model and the plate diagram of LDA model.

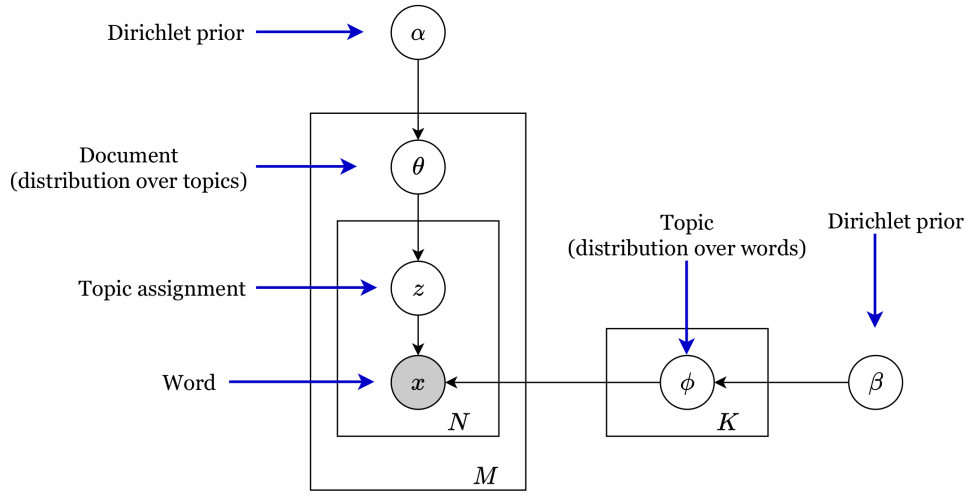


Figure 3.2: Plate diagram - Latent Dirichlet Allocation. Adapted from [21].

For the text corpus with M documents, the step-by-step generative process of LDA to infer K topics is as below [21]:

1. For each topic $k \in \{1, ..., K\}$

   $\phi_k \sim \text{Dirichlet}(\beta)$ [draw distribution over words]

2. For each document $m \in \{1, ..., M\}$

   $\theta_m \sim \text{Dirichlet}(\alpha)$ [draw distribution over topics]

   For each word $n \in \{1, ..., N_m\}$

   $z_{mn} \sim \text{Multinomial}(1, \theta_m)$ [draw topic assignment]

   $x_{mn} \sim \phi_{z_{mn}}$ [draw word]

The generative process of LDA starts with a parameter of the Dirichlet prior, β, on the word distribution of topic k in set of topics K. Each topic k is defined as a Multinomial distribution ϕ_k over the words. α is a parameter of the Dirichlet prior over the topic distribution of document m in the set of documents M. Each document m, is defined as a Multinomial distribution θ_m over the topics. For each word n in the document m with maximum words

N_m, z_{mn} is the topic assignment for the word n.So in the generative process, the model tries to understand the process of creating the documents from the words (vocabulary). This understanding of documents is then used to infer the topics by reverse engineering. In LDA, words (x_{mn}) are the only observed variables. All other variables, topic assignment (z_{mn}), distribution of words defining these topics (ϕ_k), distribution of topics defining the documents (θ_m) are all latent/unobserved variables and are inferred from the words (x_{mn}). For inferring these hidden latent variables given a document, many approximate inference techniques are used in LDA.

3.2 Word Embeddings

Word embeddings are representations of words in vector form (real-valued vectors) such that these representations capture either the syntactic or semantic meaning of the word or the context of the word or combination of these parameters. In word embeddings, words are represented as fixed-length real-valued vectors. These word associations are learned using many different techniques like neural networks, bi-directional LSTM, auto-encoders, etc. Once learned, these pre-trained representations can also be reused for a different data set. Hence it can be inferred that pre-trained word embeddings can be used as a form of transfer learning [22].

Word embeddings vectors can further be used along with mathematical functions like Cosine similarity or Euclidean distance to find semantically similar words.

Cosine similarity measures the similarity between given two non-zero vectors of an inner product space. Given two vectors, **A** and **B**, the cosine similarity, cos(θ), can be derived from the Euclidean dot product formula as below [23] :

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

$$\text{Cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

where A_i and B_i are components of **A** and **B** respectively [23].

Euclidean distance between two points in Euclidean space is the length of a line segment between these two points. For points given by Cartesian coordinates in n-dimensional Euclidean space, the distance is

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$$

Many different models exist to construct different kinds of word embeddings [22], e.g., Word2vec by Mikolov et al. [24], GloVe by Pennington et al. [25], ELMo by Peters et al. [26], BERT by Devlin et al. [2], etc.

3.2.1 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency is a statistical measure of the importance and relevance of a word to a document in a collection of documents. In TF-IDF calculations, a word's importance is proportional to the number of times that word appears in the document but then it is offset by the frequency of that word in the entire corpus. According to the TF-IDF score, words that are common in every document, e.g., word "the" have lower score even though they appear many times in a document, whereas the word "carbon" appears many times in a document, but does not appear in other documents, so it will be considered relevant. TF-IDF score for the word x in document m from the document set M is calculated as follows [27]

$$\text{TF} - \text{IDF}(x, m, M) = \text{TF}(x, m) \cdot \text{IDF}(x, M)$$

where, term frequency is

$$\text{TF}(x, m) = \log(1 + \text{freq}(x, m))$$

and, inverse document frequency is

$$\text{IDF}(x, M) = \log(D/\text{count}(m \in M : x \in m))$$

TF-IDF score representation of words can be a very useful technique to represent documents as it can be a better replacement of word count based statistics and can also be fed into other machine learning algorithms for document clustering and classification. It can also be used to find the most similar documents, as documents with similar important words will have similar vector representation.

3.3 Transformer-based Language Models

This section of the chapter includes theoretical explanation of concepts required to explain transformer-based language models like BERT [2] and Sentence-BERT [28] which are used in this thesis. To explain these language models, this section contains a brief introduction on language models and transformer architecture.

3.3.1 Language Models

Probabilistic language model is a probability distribution over sequences of words in a language. Given a sequence of words, of length j, it assigns a probability $P(X) = P(x_1, ..., x_j)$ to the whole sequence. This can be further decomposed using chain rule [29],

$$P(X) = P(x_1)P(x_2|x_1)P(x_3|x_{1:2})..., P(x_j|x_{1:j-1})$$

$$= \prod_{i=1}^{j} P(x_i|x_{1:i-1})$$

Language model is often approximated by n-gram models. A uni-gram model assumes each word in the sequence is independent,

$$p(x_1, x_2, ..., x_j) = p(x_1)p(x_2)...p(x_j)$$

And a tri-gram model assumes the probability of the current word only depends on the previous two words,

$$p(x_1, ..., x_j) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)...p(x_j|x_{j-2}, x_{j-1})$$

Given a document m, the probability of generating a word x is estimated using maximum likelihood estimation (MLE),

$$p(x|m) = \frac{tf(x, m)}{count(tokens \in m)}$$

where, $tf(x, m)$ is the term frequency of word x in document m and $count(tokens \in m)$ refers to the total number of tokens in document m.

And the probability of generating a given query q is,

$$p(q|m) = \prod_{x \in q} p(x|m) = \prod_{x \in q} \frac{tf(x, m)}{count(tokens \in m)}$$

Documents are further ranked based on this probability $p(x|m)$ and higher probability implies the document is more relevant to the given query [30].

3.3.2 Transformers

Transformer models are based on the concept of attention which increases the speed of training the model as it uses parallelization. The Transformer architecture was presented in the paper *Attention is all you need* by Vaswani et al. [31].

Transformer uses the basic encoder-decoder design of traditional neural machine translation systems [32]. Transformer also uses a multi-head attention block. Attention mechanism was proposed as an efficient way for handling large sentences in sequence-to-sequence neural machine translation models. This was done by selectively focusing on parts of the input sentence during translation [33]. In attention mechanism, only a part of input sentence is paid attention. This relieves the encoder from encoding all information in the input sentence into a fixed length vector [34]. Figure 3.3 demonstrates the transformer encoder architecture and Figure 3.4 demonstrates the transformer decoder architecture details. These images of the architecture are adapted from the original paper [31]. The encoder is a stack of T identical layers. Decoder also consists of the same number of T identical layers. Each encoder layer consists of two sub-layers, a multi-head self-attention layer and a fully connected feed-forward neural network. The multi-head self-attention layer enables the encoder to consider other words in the input sentence as it encodes a word from the input. The output of multi-head self-attention layer then goes to feed-forward neural network. Output of each encoder sub-layer is followed by layer-normalization step [32]. So if input is vector y then output of each encoder sub-layer is $LayerNormalization(y + Sublayer(y))$. Also, each sub-layer has residual connection around it which is also followed by layer-normalization step.

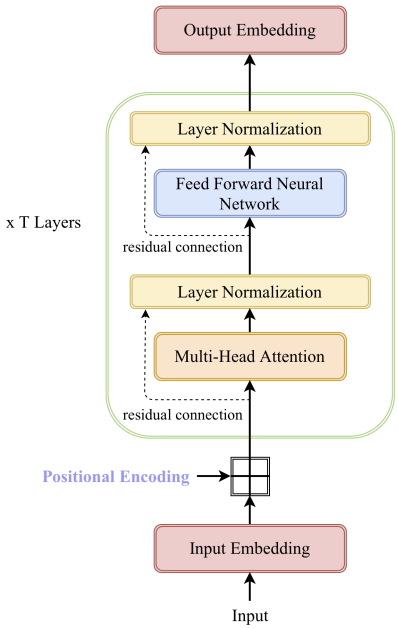


Figure 3.3: Transformer Encoder Architecture. Adapted from "Figure 1: The Transformer - model architecture." in [31].

Transformer decoder also consists of the same number of identical layers T as encoder. Each decoder layer also has both the sub-layers which are there in encoder but in addition to those two sub-layers the decoder consists of an additional third sub-layer for multi-head attention. This third sub-layer helps the decoder to focus on relevant parts of the input sentence [32]. For decoder sub-layer also the layer normalization step is followed the way it is done in encoder. The decoder output is then passed through linear transformation and softmax function to get a probability distribution and this converts the decoder output to predicted probabilities.

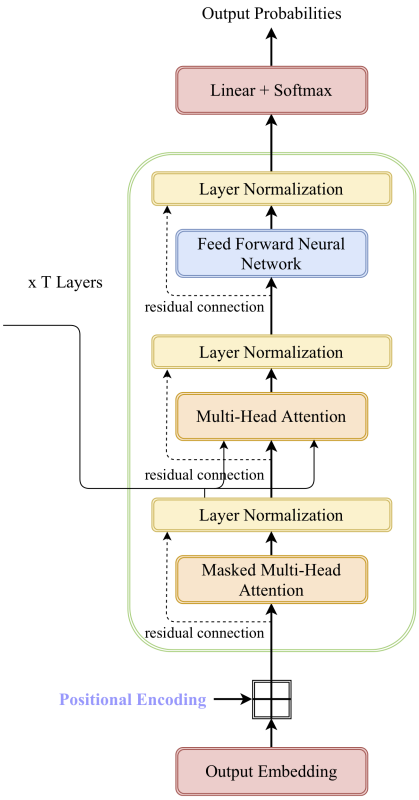


Figure 3.4: Transformer Decoder Architecture. Adapted from "Figure 1: The Transformer - model architecture." in [31].

Another important part of the transformer model is the positional encoding. After the input is converted into input embeddings, positional encoding is added to this embedding to capture the information regarding the relative or absolute position of each word token in the sequence of a sentence. This addition of positional information is done for the input of both encoder and decoder stacks.

Attention mechanism can be described as a mapping function that maps a query and set of key-value pairs to the output. The output here is the weighted sum of values. These weights are computed by a function of query and the corresponding key. There are different attention functions that can be used. The transformer model uses a multi-head attention mechanism that is based on scaled dot-product attention. Scaled dot-product attention computes the dot products of the query with all the keys, scales it, and applies the softmax function to get the weights for the values [31]. These attention weights give an idea of how much attention or focus should be paid to the different words in the input sentence while working on one word of the input sentence. This technique enables better Multi-head attention is refined from scaled dot-product attention and is faster and performs better. Multi-head attention can be thought of as many parallel self attention heads and output from these heads is further concatenated to produce a single output. Because of this, multi-head attention has the ability to focus on different positions and handles multiple mappings of query and key-value pairs to the output in parallel.

These different mechanisms used in transformer model enables it to process long sequences of text faster by leveraging parallel computing along with attention mechanism and at the same time capture the context of the words much more efficiently by implementing positional encoding along with the self-attention.

3.3.3 Bidirectional Encoder Representations for Transformers (BERT)

BERT is a language model [2] based on transformer model architecture. Traditional language models are unidirectional whereas BERT is bi-directional. Since BERT architecture consists of bi-directional transformer encoder layers, that means the transformer encoder reads the entire sequence of the words (from both directions) at once and hence provides word embeddings based on the full context of the word. Transformer encoders also uses attention, that means it can handle long text sentences effectively.

BERT is pre-trained on large data set of English Wikipedia (2.5B words) and BookCorpus (dataset consisting of 11,038 unpublished books from 16 different genres). The same pre-trained model can be fine-tuned and used for more tasks. BERT is pre-trained bidirectionally using unlabeled data and hence, can be fine-tuned for wide range of tasks by adding one additional output layer [2].

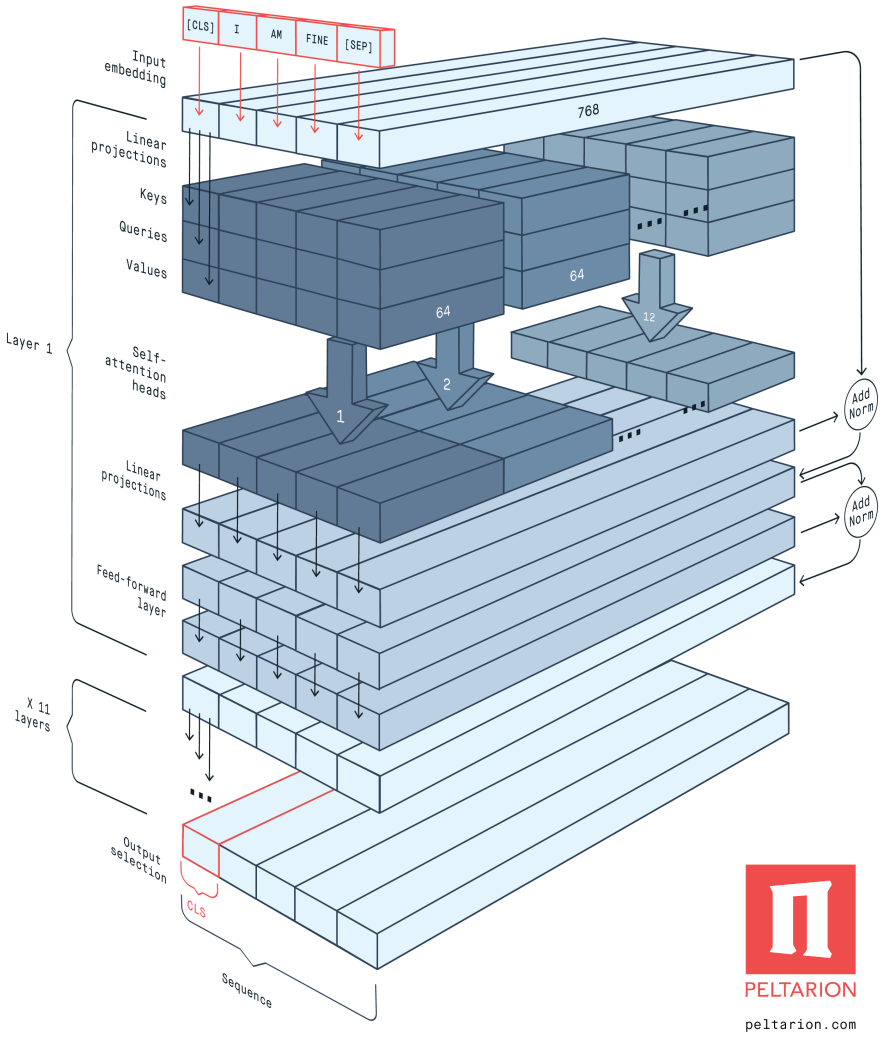


Figure 3.5: 3D representation of a transformer (BERT) [2]

[2] 3D representation of a transformer (BERT) by Peltarion. https://peltarion.com/blog/data-science/illustration-3d-bert

Their are two pre-trained models available for BERT, $BERT_{BASE}$ and $BERT_{LARGE}$. $BERT_{BASE}$ model has 12 transformer encoder layers, 768 hidden layers and 12 self-attention heads. $BERT_{LARGE}$ model has 24 transformer encoder layers, 1024 hidden layers and 16 self-attention heads. Figure [3.5] is a 3D representation of the $BERT_{BASE}$ architecture. It demonstrates one layer of the transformer encoder in detail and then shows that 11 more such identical layers exists in the architecture. Since it is a transformer encoder, one layer of encoder has 12 self-attention heads working in parallel and the output from these attention heads are concatenated, normalized, and passed to feed forward network layer.

BERT uses some special tokens to handle variety of tasks. [CLS] token is a special classification token that is used to indicate the start of every sequence. The embedding from this token is used as the aggregate sequence representation for classification tasks. [SEP] token is a special separator token that is used to separate two sentences in the sentence pair tasks. [CLS] and [SEP] tokens are used both while pre-training and fine-tuning. Another special tokens, [MASK] is only used during pre-training to mask (hide) the words. Figure [3.6] (adapted from the original paper [2]) gives visualization of how the input embedding is constructed for a given token using the special tokens and by summing up the respective token embeddings, segment embeddings, and position embeddings. The token embeddings in addition to embeddings for tokenized words also includes labels for the special BERT tokens like [CLS] and [SEP]. The segment embeddings includes labels for the different sentence segments passed in the input for sentence pair tasks to be able to distinguish between the segments. The position embeddings includes labels for the position of words in the input.

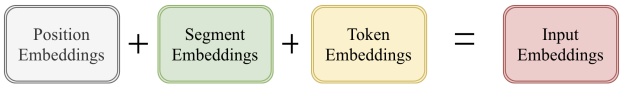


Figure 3.6: BERT input embeddings are the sum of the token, segmentation and position embeddings. Adapted from "Figure 2: BERT input representation." in [2].

BERT is pre-trained on two unsupervised NLP tasks, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, the model is pre-trained using the task of predicting the missing word in the given sequence. While training BERT, some words were replaced with [MASK] token and were treated as missing words. This was done randomly to 15% of the words, to prevent the model from focusing a lot on a some position or masked tokens. While training, 80% of the time the words were replaced with the token [MASK], 10% of the time the words were replaced with other random words and other 10% of the time the words were left unchanged [35]. With MLM method, embeddings capture the understanding of the relationship between words [2]. BERT is also pre-trained on the task of NSP so that the embeddings can also capture the understanding of the relationship between sentences. It is a binary classification task. The data is generated from a corpus by splitting it into sentence pairs. For half of the training pairs, the second sentence is actually the next sentence with label 'IsNext' and for the remaining half of the pairs , the second sentence is a random sentence from the corpus with label 'NotNext'. In this the [CLS] token is used to represent the classification task and start of the first sentence, [SEP] token is used to represent the separation between two sentences and the start of second sentence. To indicate the end of the second sentence again [SEP] token is used [2].

The way these pre-trained embeddings can be used is to fine-tune them for a variety of specific supervised-learning tasks or to use the contextual pre-trained word embeddings as it is without fine-tuning. BERT architecture is the same for the both pre-training and fine-tuning apart from the output layer. The same pre-trained model parameters are used to initialize models to fine-tune for different tasks. This is how BERT makes use of transfer learning. During the fine-tuning, all parameters are fine-tuned, and weights are updated. BERT can be fine-tuned for sentiment analysis, question answering tasks, or named entity

recognition (NER) tasks, as demonstrated in the original paper [2] using different datasets. The advantages of this kind of fine-tuning are that better accuracy can be achieved even when smaller labeled datasets are available for training as the pre-trained embeddings from the model already has a lot of language understanding. This existing knowledge also makes the fine-tuning faster as lesser epochs are required for training.

3.3.4 Sentence Embeddings using Siamese BERT-Networks (Sentence-BERT)

Sentence-BERT [28] is extended from pre-trained BERT model and provides semantically meaningful sentence embeddings rather than word embeddings. Since sentence embeddings are more suitable for the type of data used in this thesis, this model was explored and used. Sentence-BERT (also referred as SBERT) fine-tunes a pre-trained BERT network using siamese and triplet network structures to update weights. A pooling operation is then added to the output of fine-tuned BERT to produce a fixed-sized sentence embedding vector. This provides semantically meaningful sentence embeddings. These semantically meaningful sentence embeddings can then be compared using cosine-similarity for finding most similar pair tasks. These embeddings can also be used for clustering and semantic search. Using these techniques, Sentence-BERT is more computationally efficient than BERT while maintaining the accuracy standards of BERT. Sentence-BERT achieves the task of finding most similar sentence in 5 seconds for which BERT requires 65 hours as claimed in their paper [28]. BERT embeddings can also be used to derive fixed length sentence embeddings by either averaging the BERT embeddings or by using the output embedding of first token. But these BERT embeddings based techniques do not yield good sentence embeddings as shown in [28].

Sentence-BERT architecture for sentence similarity tasks using cosine-similarity is demonstrated in Figure [3.7] (adapted from the original paper [28]). This uses regression objective function. For regression objective function, sentence embeddings u and v are compared by computing the cosine similarity between them and mean squared-error loss is used as the objective function. These fixed-size sentence embeddings can also be compared using other similarity measures like Manhattan or Euclidean distance.

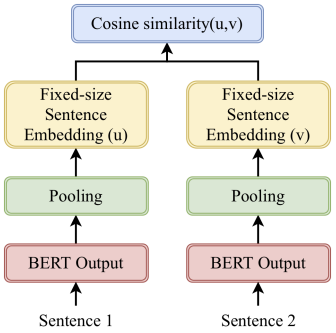


Figure 3.7: Sentence-BERT architecture to compute similarity scores between sentences. Adapted from "Figure 2" in [28].

Sentence-BERT uses different pooling techniques to derive fixed-size sentence embeddings. The default one is to compute mean of all output vectors from fine-tuned BERT embeddings. Sentence-BERT architecture for classification tasks uses softmax classifier. This is demonstrated in Figure [3.8] (adapted from the original paper [28]). Here the BERT networks have tied weights (siamese network structure). This uses classification objective function.

For classification objective function o, sentence embeddings u and v are concatenated with element wise difference $|u - v|$ and multiplied with trainable weight, $W_t \in \mathbb{R}^{3n \cdot l}$

$$o = \text{softmax}(W_t(u, v, |u - v|))$$

where n is the dimension of the sentence embeddings and l the number of labels. Here cross-entropy loss is optimized.

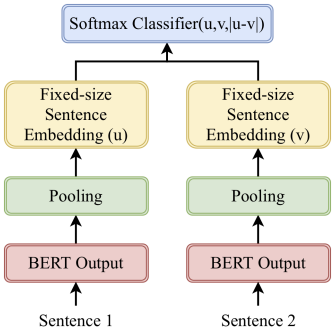


Figure 3.8: Sentence-BERT architecture with classification objective function. Adapted from "Figure 1" in [28].

3.4 Document Clustering

Document Clustering refers to the technique of gathering similar documents into subsets (called clusters) given a large set of documents with the aim that these subsets of similar documents are substantially different from each other. Document Clustering is performed by calculating distance between documents and then forming clusters of documents having least distance among themselves but yet substantial distance among the document clusters. Distance between documents is performed by calculating euclidean or cosine distance between some kind of numerical representations of the documents. These numerical representation could be TF-IDF score, word embeddings or sentence embeddings or any other numerical representation of the documents. There are many applications of document clustering like finding similar documents which can help in search or finding duplicates among documents, can also help in organizing large text corpus, better search engines and recommendation systems. Many different clustering algorithms can be leveraged for clustering documents as well like K-means, dbscan, hierarchical clustering like hdbscan and others.

3.4.1 K-means Clustering

K-means is one of the most popular and extensively used algorithm for clustering. It's simple and yet effective in finding good quality K number of clusters within a given larger dataset. In K-means algorithm, the aim is find user-specified K clusters which refers to the number of centroids in the dataset. First, K initial centroids are chosen and each point in the dataset is assigned to the nearest centroid. Each collection of points allocated to a centroid is a cluster. Then based on points assigned to a cluster the centroid of each cluster is updated. This iterative process carries on until centroid does not change any further. Algorithm 1 describes this simple description of the basic algorithm [36].

Algorithm 1 Basic K-means algorithm. [36]

Select K points as initial centroids.
repeat
  Form K clusters by assigning each point to its closest centroid.
  Recompute the centroid of each cluster.
until Centroids do not change.

In K-means algorithm, mean of the data points is used as a centroid. So, after points are assigned to the centroid, the centroid is again updated. After centroid is updated, points are again assigned to the closest centroid and centroids are again updated. Points are assigned to the closest centroid based different distances based on the data points in the problem set. Euclidean distance is used if data points are in Euclidean space. Whereas, cosine similarity is more suitable if our data points are documents or sentences. Other options are Manhattan distance and Jaccard measure [36].

Given a set of observations $(x_1, x_2, ..., x_n)$, where each observation is a v-dimensional real vector, K-means clustering aims to partition the n observations into $K(n)$ sets $S = S_1, S_2, ..., S_K$ so as to minimize the within-cluster sum of squares (i.e. variance) [37].

The objective is to find [37]

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \boldsymbol{\mu}_i||^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

where μ_i is the mean of points in S_i.

This is equivalent to minimizing the pairwise squared deviations of points in the same cluster

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\mathbf{x}, \mathbf{y} \in S_i} ||\mathbf{x} - \mathbf{y}||^2$$

4 Method

This project is implemented in Python programming language using scikit-learn [38] and huggingface transformer libraries [39]. For model training, GPU provided by Google Colab is used. The data was downloaded using a Python script and APIs provided by Science Direct.

Methods used to solve the problem set in this thesis are divided into traditional machine learning approaches (using LDA, document clustering, TF-IDF) and transformer-based approaches (using BERT and Sentence-BERT).

4.1 Data Preprocessing

In the pre-processing step, very common and frequent English stopwords were removed. No other data pre-processing was done as based on research papers in this area like this study by Schofield et al. [6], where it was found that removal of stopwords has little impact on quality of topic models i.e. model likelihood, topic coherence, or classification accuracy. They observed that removing determiners, conjunctions, and prepositions can improve model fit, but further removal has little effect on topic model inference. Also, stemming and lemmatization can, in fact, have an adverse impact on topic model inference as it often places words sharing morphological roots in the same topics. In their experiments, they found out that topic coherence also did not improve between pre-stemming and post-stemming. So these heavy and time consuming pre-processing tasks for textual data like such as stemming, lemmatization, and corpus specific lists for stopwords, seem to be having little impact on the quality of topic models and their coherence, specificity, and uniqueness. So in this project, very light pre-processing of the text was done to avoid discarding any useful word information.

4.2 Traditional Machine Learning Approaches

4.2.1 Topic Modeling using LDA

After downloading all the documents in the Journal, data pre-processing was done, which included cleaning up the data and removing common English stop words. Then, as a next step LDA was performed on the text from abstracts of all the 24000+ documents. This step did not involve any other data manipulation. For LDA, scikit-learn [38] libraries were used.

Limitations of using LDA are that number of topics and words per topic need to be mentioned apriori. And for infrequent word, LDA cannot learn its correct semantics from the observed distribution and will assign it to the dominant document topic [7].

Another main limitation when it comes to the dataset being used for this thesis is that it is a vast and varied dataset of all the publications in a journal. Using LDA topic modeling on abstract data (average length of 200 words per abstract) of 24000+ documents did not give very meaningful and intuitive topics. This could also be because these abstracts are on such a vast variety of research areas. Thus the approach decided to overcome this challenge (inspired by Venkatesaramani et al. [5]) is document clustering. The abstracts are clustered based on similarity and topic modeling is applied to these clusters individually. Based on topic modeling results, a topic (collection of words) is assigned to each cluster. This is further discussed in next section.

Another difficulty of this approach was to do trend analysis from these results. An approach of using the average TF-IDF score of the words in topics was used to assign a score to each topic and then trend over the years was extracted. This involved doing these steps:

- Step 1: Getting TF-IDF score of each word in the topic for that particular year.
- Step 2: Taking average of the TF-IDF scores of all the words in the topic to come up with a score for each topic per year.

4.2.2 Document Clustering and LDA

In this approach, after collecting all the documents in the Journal, data pre-processing was done, which included cleaning up the data and removing common English stop words. Post that, the wide range of documents were clustered based on a similarity measure. Once the clusters of the documents were created, then LDA topic modeling was applied on each of these clusters independently to find topics within those clusters. This approach of handling large text corpora is inspired by [5].

For Document clustering, the following steps were implemented using K-means with cosine similarity measure on TF-IDF score:

- Step 1: Calculated TF-IDF scores for the abstract text of each document.
- Step 2: K-means clustering with cosine similarity based on the TF-IDF score was done to create clusters.
- Step 3: LDA topic modeling was performed on each cluster independently to get the topics from that cluster.

Each cluster is associated with one topic where topic is collection of related words. Hence, all the documents belonging to one cluster are associated with the same topic.

4.2.3 Document Clustering and TF-IDF

In this method, the document clustering approach remains the same as above, i.e., pre-processed documents are clustered using K-means clustering with cosine similarity based on the TF-IDF score. The difference here is that once the clusters of similar documents are formed, they are analyzed based on the TF-IDF scores. In the previous approach, we had used LDA topic modeling for this purpose. So the clusters, in this case, are represented by their important and relevant words. Since the TF-IDF score can provide the most important words from the given document based on the theory explained in Section 3.2.1, we have leveraged this capability of TF-IDF to create a visualization of document clusters (in our case, these are clusters of similar abstracts). To visualize the content of these clusters, we used word clouds based on TF-IDF scores. TF-IDF is also more suitable for infrequent, relevant words.

Each cluster is visualized with one word cloud based on TF-IDF scores, which is representative of all the documents in that cluster.
The main limitation of these two methodologies using document clustering shared above in Sections 4.2.2 and 4.2.3 is that the clustering is done based on TF-IDF scores from the abstract texts. But, TF-IDF does not capture the contextual meaning of the word. To overcome this, pre-trained BERT word embeddings [2] and pre-trained Sentence-BERT sentence embeddings [2] are used, which provide embeddings based on the context of the word.

4.3 Transformer-based Language models

Transformer-based language models as discussed in Section 3.3.2 are good at generating word embeddings that captures the context of the word. And with advancements in the field of NLP, we now have access to such pre-trained embeddings that are already trained on a large and varied dataset.

4.3.1 Semantic Topic Modeling and Trend Analysis using BERT Embeddings

In this approach, pre-trained BERT (explained in section 3.3.3) word embeddings were used. Each individual sentence from abstract texts was input into pre-trained BERT model to derive fixed size embeddings and the output of the first token (the [CLS] token in BERT) was used. These embeddings were then fed to K-means algorithms to find clusters of similar abstracts. Once the clusters were generated, LDA topic modeling was done on each of these clusters independently to get the topics. For each cluster, TF-IDF based world cloud visualization was also done to understand the content of these clusters.

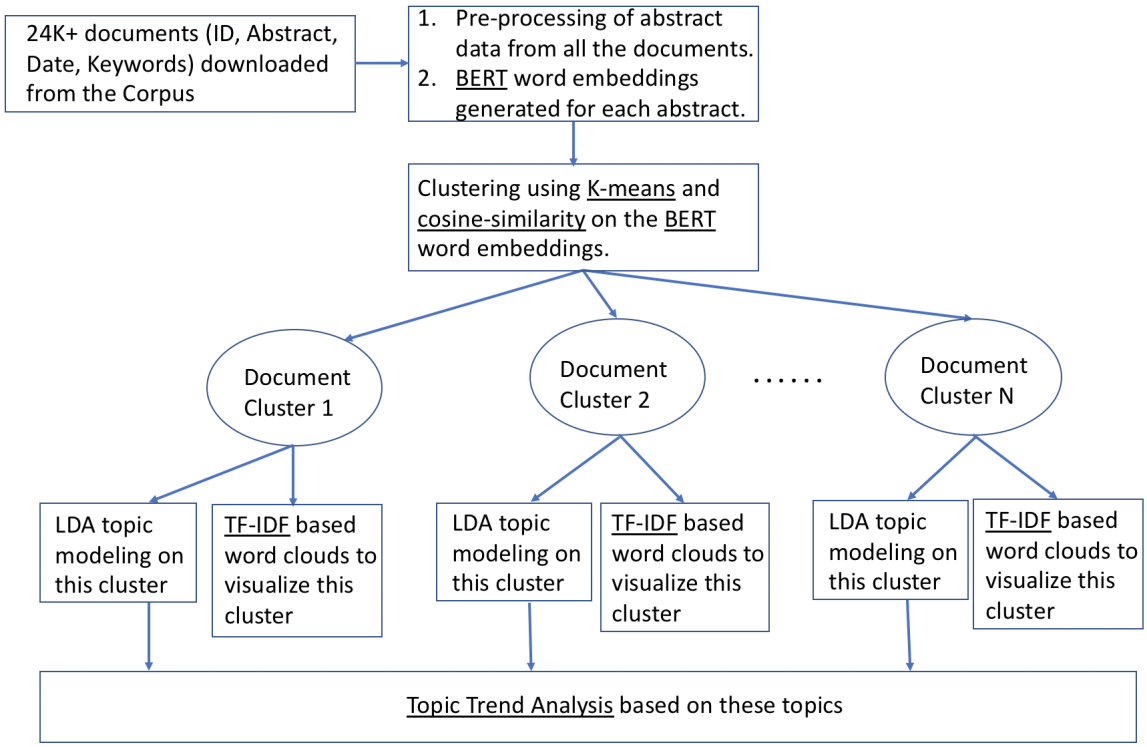


Figure 4.1: Semantic topic modeling and trend analysis using BERT word embeddings.

All the documents in a cluster were assigned to the same topic. Once all documents were associated with a topic, trend analysis of these topics was done using the published date information available for each document.

Figure 4.1 demonstrates this method of semantic topic modeling and trend analysis using BERT word embeddings in a chart format.

4.3.2 Semantic Topic Modeling and Trend Analysis using Sentence-BERT Embeddings

In this method, pre-trained Sentence-BERT (explained in Section 3.3.4) sentence embeddings were used. For each abstract text per document one fixed length embedding of size 768 was created using pre-trained Sentence-BERT. These embeddings per abstract were then used for clustering using K-means algorithms to find clusters of similar abstracts. Cosine-similarity measure was used for this clustering. Once the clusters were generated both LDA topic modeling and TF-IDF based world cloud visualization was done to understand each of these clusters. For all the documents belonging to each cluster, the same LDA topic was assigned and these LDA topics were further used for trend analysis as the temporal information for documents is also available (published date). Figure 4.2 demonstrates the method for semantic topic modeling and trend analysis using Sentence-BERT embeddings.

Using sentence embeddings was much more suitable for our data which is abstract text with average length of 202 words per abstract. And according to the paper [28] in which Sentence-BERT was introduced, it is much more efficient at calculating the similarity between two embeddings than BERT. Since BERT uses a cross-encoder, two sentences are passed to the full transformer network and the target value is predicted. In a corpus of 10000 sentences, finding similar pairs of sentences requires about 50 million inference computations taking approximately 65 hours on a modern V100 GPU. Whereas, Sentence-BERT reduces this complexity for finding the most similar sentence pair in a collection of 10,000 sentences is reduced to 5 seconds and computing cosine similarity to 0.01 seconds [28].

In this approach, top words based on TF-IDF score in each cluster (which are also used to generate word clouds) can also be considered as a topic representing that particular cluster. The trend analysis can then be done based on these most relevant and important words acting as a topic.

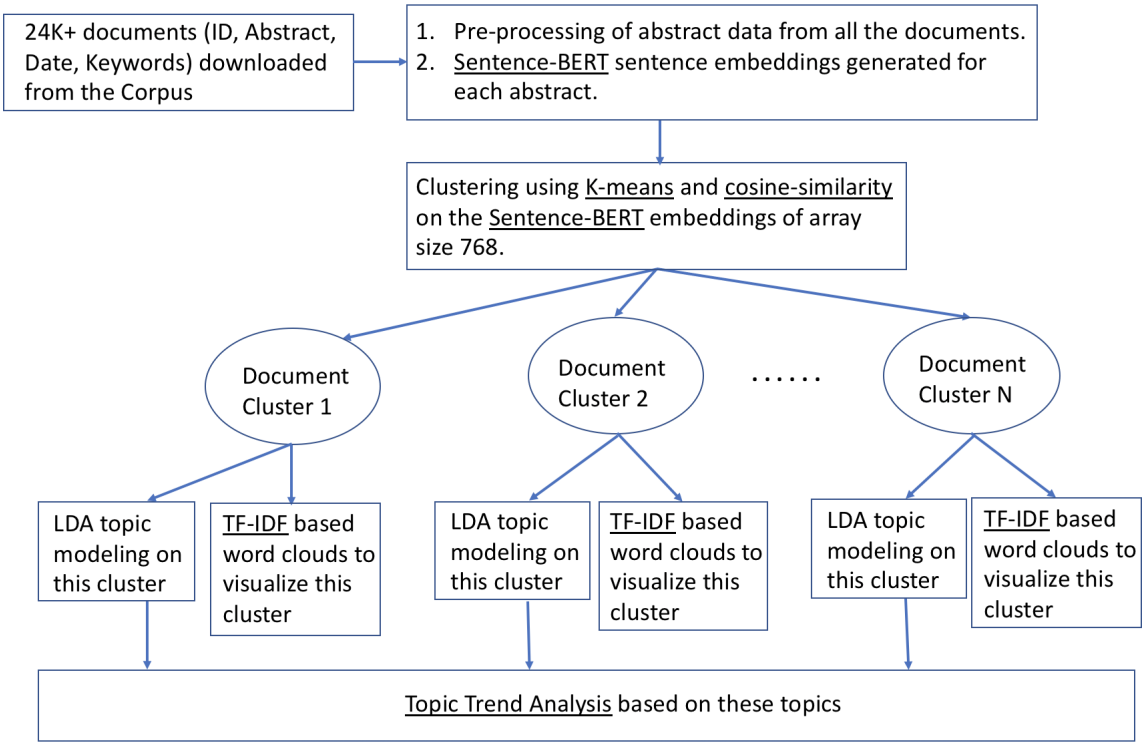


Figure 4.2: Semantic topic modeling and trend analysis using Sentence-BERT Embeddings.

5 Results

5.1 LDA Topic Modeling on Entire Data

Figure 5.1 shares the top 20 results from LDA topic modeling on the abstract text of all the documents with 10 words for each topic. From these results it is difficult to assign a topic to them as they are very varied and not related to each other. Another difficulty of this approach was doing trend analysis on these results. An approach to achieve trend analysis is discussed in Section 4.2.1, but as can be seen in Figure 5.2, this did not yield informative results.

```
Topic 0: water urban land use soil areas management urbanization cities area
Topic 1: energy emissions consumption efficiency cost reduction emission results study total
Topic 2: power fuel electricity solar coal gas heat thermal wind generation
Topic 3: oil extraction acid surface process cutting biodiesel method fiber particles
Topic 4: sustainable sustainability development research economy change circular business innovation future
Topic 5: carbon emission emissions low dioxide gas methane pyrolysis powder process
Topic 6: environmental green study performance management supply chain results companies research
Topic 7: cr vi leather stability neural artificial microwave hydrothermal rainwater contained
Topic 8: design risk process manufacturing health environment processes material product environmental
Topic 9: model sustainability proposed decision analysis based approach study models different
Topic 10: countries production cleaner trade states r united developing d european
Topic 11: china s development pollution economic industrial policy results environmental growth
Topic 12: production food process energy biomass plant recovery waste processing technology
Topic 13: c metal leaching asphalt iron m metals ash slag catalyst
Topic 14: concrete strength steel compressive aggregate fly resistance geopolymer board durability
Topic 15: results study treatment high water used showed properties using different
Topic 16: adsorption g removal n mg p ph surface x electron
Topic 17: environmental life cycle impacts impact assessment results production emissions study
Topic 18: waste recycling construction materials recycled material wood wastes rubber composite
Topic 19: biochar vehicles vehicle electric transport battery transportation ce variable logistics
```

Figure 5.1: Top 20 Results from LDA topic modeling on abstract text of all the documents with 10 words per topic.

5.2 Document Clustering (LDA and TF-IDF)

The results shared in Figure 5.3 demonstrate the division of documents from the corpus into clusters of similar documents when the TF-IDF score is used along with K-means clustering.

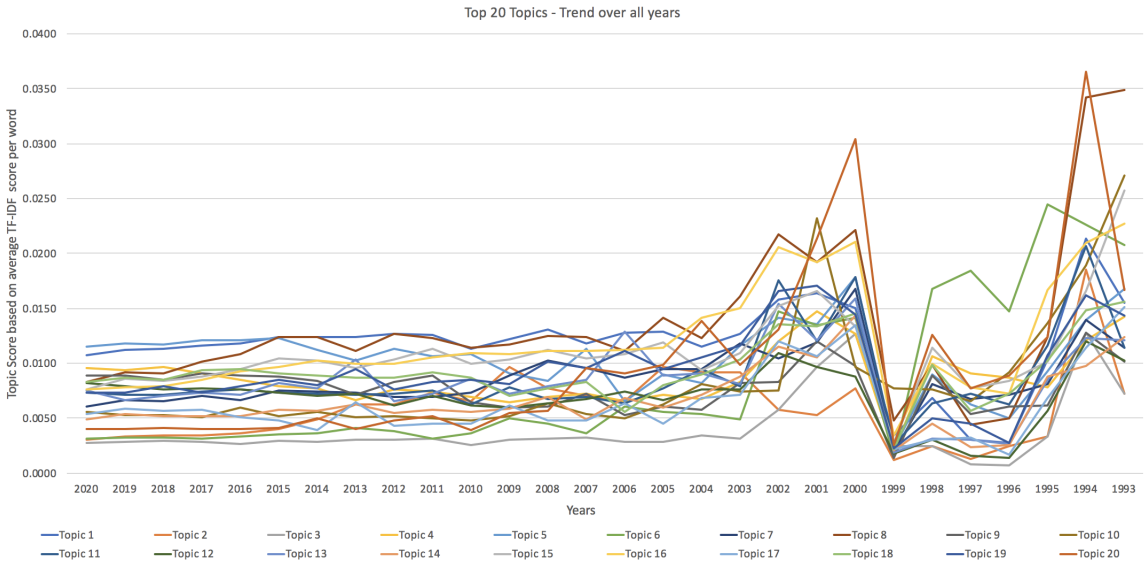


Figure 5.2: Topic trend analysis of top 20 Results from LDA topic modeling on abstract text of all the documents with 10 words per topic.

In this figure, the segments are the document clusters. The purpose of including this figure here is to give an idea of the percentage of documents falling within a cluster. It can be seen that based on TF-IDF scores, two big clusters (Seg7 and Seg8) constitute more than half of all the documents.

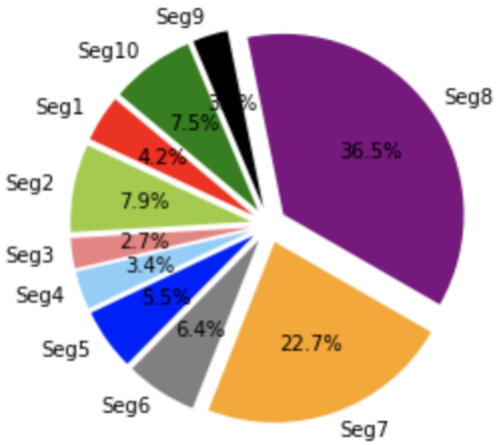


Figure 5.3: Document clusters based on TF-IDF scores of each abstract using K-means clustering

When LDA topic modeling is done on abstracts of all the documents within each of these clusters, the results were very reasonable and made sense according to the background of the journal. Figure 5.4 shares the results of topic modeling using LDA on some of the cluster segments from Figure 5.3. The boxes depict document clusters in this figure, and each line is an LDA topic. These results are better refined than the ones from doing LDA topic modeling on entire data in the previous section.

gross domestic product	carbon dioxide emission	supply chain sustainability
ch4 n2o emissions	low carbon economy	environmental social economic
iron steel industry	doped carbon dots	global reporting initiative
land use change	low carbon technologies	sustainable project management
based co2 emissions	residents low carbon	social responsibility csr
greenhouse gas emissions	low carbon products	education sustainable development
related co2 emissions	production based carbon	economic environmental social
greenhouse gas ghg	dissolved organic carbon	supply chain management
fossil fuel consumption	product carbon footprint	sustainable development goals
carbon dioxide co2	climate change mitigation	product service systems
fossil fuel energy	fossil energy carbon	corporate social responsibility
residential co2 emissions	carbon emission reduction	higher education institutions
gross domestic product	reduce carbon emissions	sustainable business models
coal fired power	carbon emission trading	future research directions
life cycle ghg	life cycle assessment	emergy sustainability index
co2 emissions reduction	carbon emissions china	higher education institutions
greenhouse gas ghg	total factor carbon	corporate social performance
consumption co2 emissions	gross domestic product	social environmental economic
energy efficiency improvements	multiwall carbon nanotubes	multi criteria decision
renewable energy consumption	carbon dioxide emissions	sustainable products services

Figure 5.4: Results from LDA topic modeling on all the abstracts per cluster for some of the clusters in Figure 5.3. The boxes represent a cluster and each line is a topic.

Based on these results, the documents were further assigned the topics, and trend analysis was done based on the available published date to achieve a year based trend analysis. All the documents in a cluster were assigned the same topic. Figure 5.5 demonstrates the trend of these topics through the years. Each topic here is representing a cluster of documents. The chart in Figure 5.5 is based on percentages rather than actual. As the number of documents before the year 2015 are very less compared to years after 2015 so, to be able to compare the results with each other, percentages were used. The trend chart based on actual can also be seen in Figure 5.6.

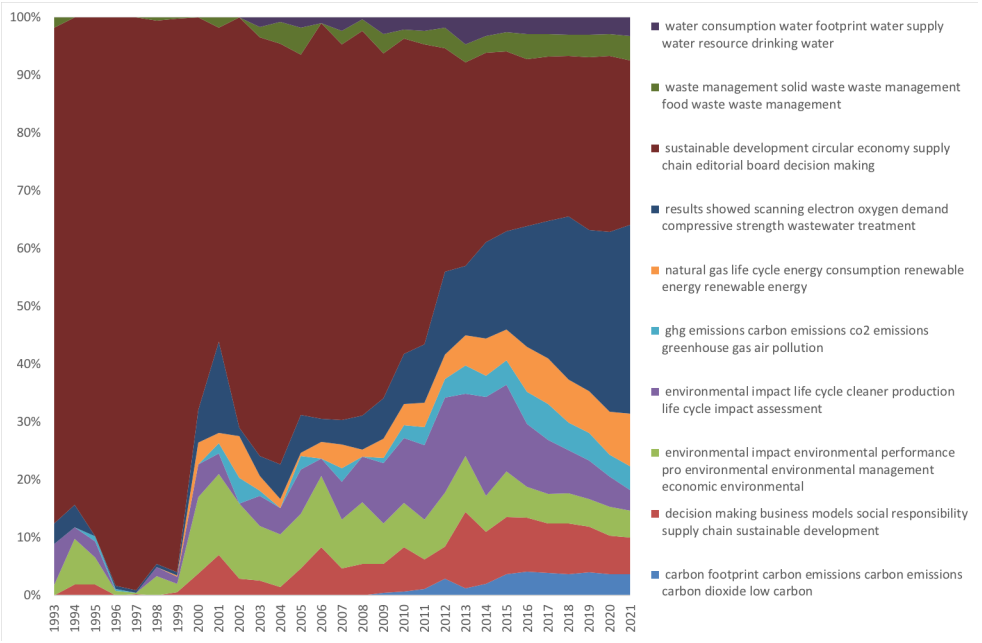


Figure 5.5: Topic trend analysis from TF-IDF embeddings based clustering and LDA

Figure 5.5 gives us an idea of changing trends. Some of the information which can be inferred from this trend analysis is:

- "Natural gas life cycle" topic started coming up around the year 2000 and has been constant area of interest since then.

- "ghg (green house gases) emissions" and "carbon emissions" as a topic first appeared in 1995 and then again from 2001-2010 but, the actual interest started building up around 2012.
- "Solid waste management" has been a topic of research since 2000 with continued interest.
- "Carbon footprint" started appearing in research documents around 2009.
- There has been increased interest in "waste water management" in last couple of years.

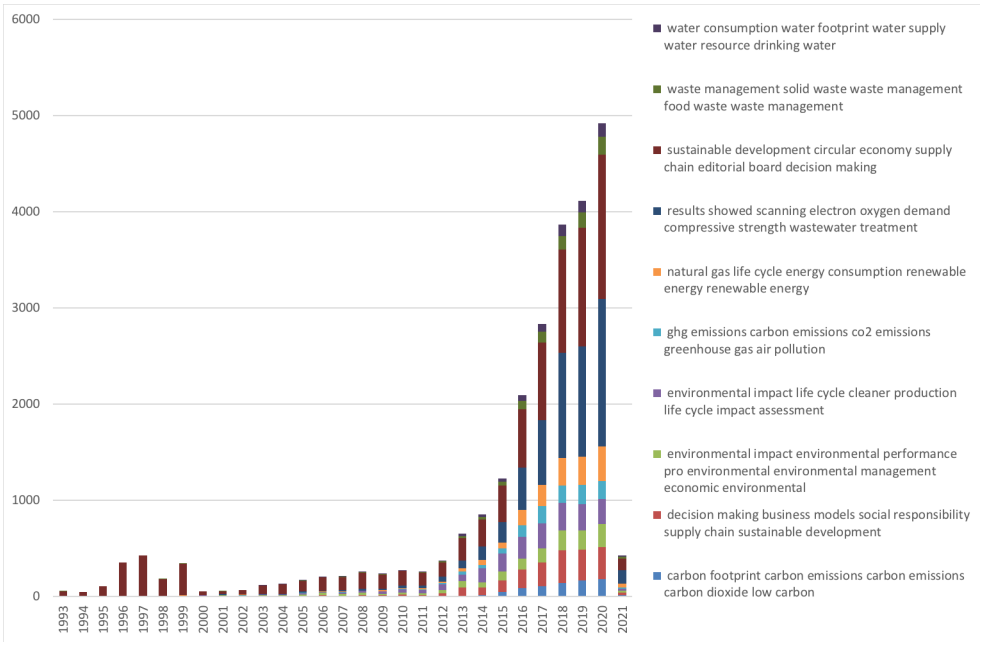


Figure 5.6: Topic trend analysis from TF-IDF embeddings based clustering and LDA

5.3 Document Clustering using Sentence-BERT Embeddings

After trying both BERT word embeddings and Sentence-BERT sentence embeddings based approaches as discussed in the methods section above, Sentence-BERT sentence embeddings were found more suitable and efficient for the kind of data we have. Hence, for the purpose of not making the results section overwhelming, the results from Sentence-BERT sentence embeddings are only shared here.

The results shared in Figure 5.7 demonstrates the division of documents into clusters when Sentence-BERT embeddings are used along with K-means clustering to find the document clusters. Segments depict document clusters in this figure. This figure is included to show the difference between clustering based on TF-IDF score as shown in figure 5.3 and based on Sentence-BERT embeddings on the same dataset. Since Sentence-BERT embeddings capture the context of the word, this could be the reason for far more evenly divided clusters.

Figure 5.8 shares the details of topic modeling using LDA on some of the clusters from Figure 5.7. The boxes represent clusters here and each line in the box is a topic from LDA topic modeling.

Based on these sentence embeddings, which capture the context of the words and topic modeling, other trend analysis results are shared in Figures 5.9 and 5.10.

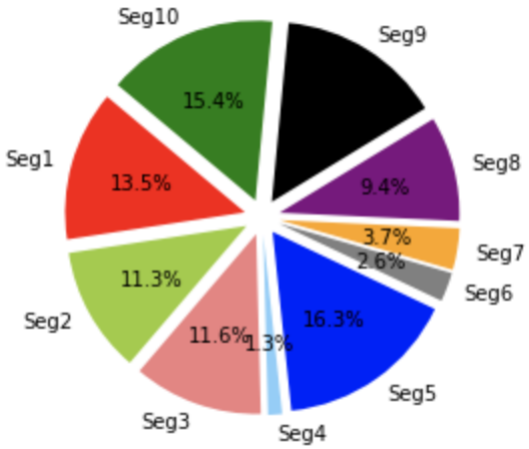


Figure 5.7: Document clusters based on sentence-Bert embeddings using K-means clustering

reduce carbon emissions reducing co2 emissions corporate social responsibility greenhouse gas emissions greenhouse gas ghg carbon dioxide co2 beijing tianjin hebei renewable energy consumption total energy consumption energy saving emission play important role total factor productivity input output analysis mean divisia index small medium sized input output model data envelopment analysis life cycle assessment iron steel industry foreign direct investment	corporate social responsibility higher education institutions greenhouse gas emissions municipal solid waste life cycle cost sustainable development goals sustainable supply chain product life cycle life cycle assessment new product development circular business model decision making process semi structured interviews waste cooking oil closed loop supply energy performance contracting wastewater treatment plants small medium sized eco industrial parks decision making trial evaluation	renewable energy systems chemical oxygen demand energy exergy efficiencies maximum power point internal combustion engine cost carbon emissions municipal solid waste life cycle assessment abstract paper presents carbon dioxide emissions fuel cell hybrid greenhouse gas emissions mixed integer linear multi criteria decision total energy consumption greenhouse gas ghg phase change material carbon emission reduction renewable energy sources computational fluid dynamics

Figure 5.8: Results from LDA topic modeling on abstracts from some of the clusters in Figure 5.7. The boxes represent clusters here, and each line in the box is a topic.

Figure 5.9 demonstrates the trend of topics over the years. Here each topic represents each document cluster. This chart is based on percentage so as to make it easy to have comparison over the years. Since the number of over all documents in initial years of the publication are very different from recent years so it is not possible to compare the actual numbers. It is a stacked area chart.

Some of the inferences that can be made from 5.9 are:

- Interest in "water footprint food waste life cycle" is increasing since 1999
- There is constant interest in "energy consumption supply chain" but percentage wise it has reduced in the recent years with the increase of interest in other topics.
- It also correctly captures the patent documents correctly during years 1998-2005. There are some patented documents in the dataset for which abstract is not available. These have been correctly identified here.
- Interest in "energy consumption" has relatively decreased as more areas of interest are emerging.
- There is an increasing research in "waste water management". Also, "fly ash management".

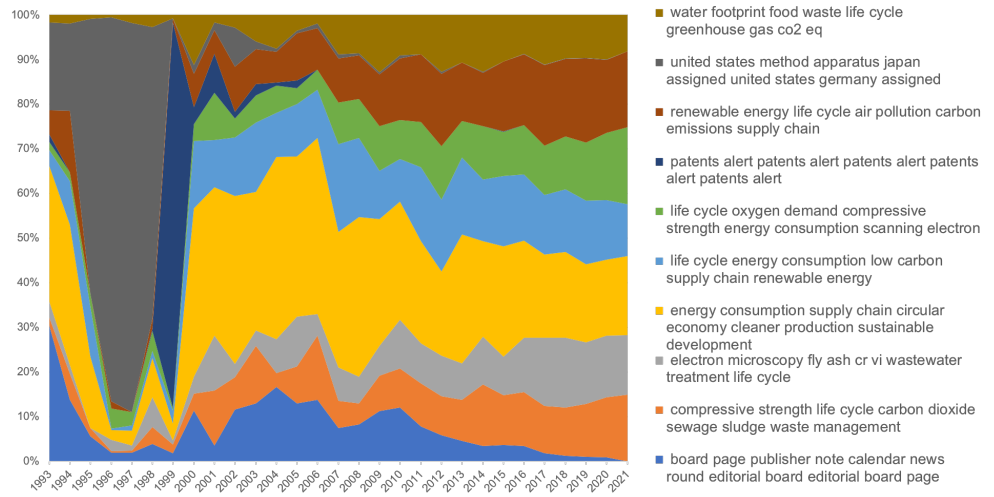


Figure 5.9: Topic trend analysis from Sentence-Bert embeddings based clustering and LDA over the years

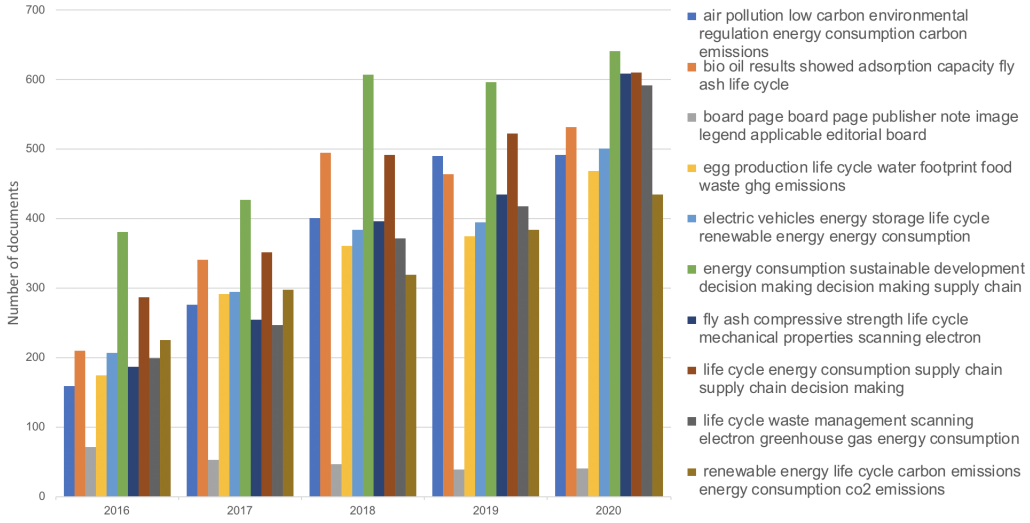


Figure 5.10: Topic trend analysis over the past 5 years

Figure 5.10 demonstrates the trend analysis of last 5 years. This comparison is useful as before that the number of documents in the journal were significantly less and it is challenging to compare them with each other. Some insights which can be inferred from these results are:

- Continued interest in "energy consumption sustainable development" in last 5 years.
- Increased interest in "air pollution low carbon" in 2020.
- This also gives idea of popular topics in last 5 years.

5.4 Topic Visualization

For topic visualization, we started with word clouds based on the word frequency of the words in all the abstract text belonging to a cluster. This resulted in word clouds that did not

include the words which were less frequent but were unique and important to the cluster. So for this purpose, we generated word clouds based on the TF-IDF scores of words belonging to a cluster (combining the abstract text of all the documents in that cluster).

Figure 5.11 demonstrates the TF-IDF based word clouds of 4 different clusters. From these results, we can infer that they are about different topics, e.g., one of the clusters consists of documents on the supply chain and another cluster is about carbon and CO2 emissions. Similarly other two clusters are about waste management and fly ash life cycle. These results were discussed with domain experts of the dataset and the results seemed to be correctly identifying the patterns.

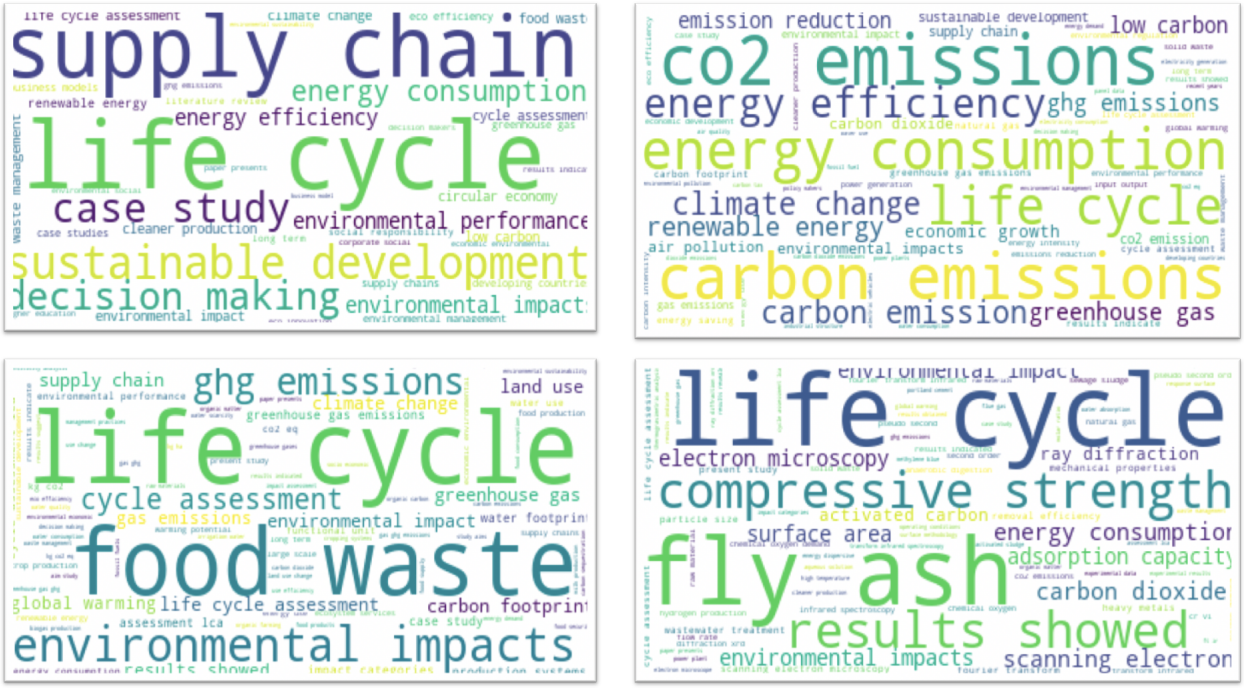


Figure 5.11: TF-IDF based word clouds representing one cluster each

5.5 Comparison

Some of the results from the TF-IDF method are compared with the Sentence-BERT method to see the change in patterns. This was done based on similar looking topics from both approaches. The purpose of doing this was to answer one of the research questions, whether it is possible to compare the results from these different methodologies. Figure 5.12 shows the comparison in number of documents attributed to the similar topics under both the TF-IDF and Sentence-Bert approach. The topics being compared using the TF-IDF approach is "carbon footprint carbon emissions carbon emissions carbon dioxide low carbon" with Sentence-Bert topic " life cycle greenhouse gas air pollution ghg emissions carbon emissions"

And Figure 5.13 shows the comparison in number of documents attributed to the similar topics under both the TF-IDF and Sentence-Bert approach. The topics being compared using TF-IDF approach is "sustainable development circular economy supply chain editorial board decision making" with Sentence-Bert topic "sustainability performance life cycle circular economy energy consumption sustainable development".

These charts show that such a comparison of results from two different approaches can be made, but the comparison results are not very consistent.

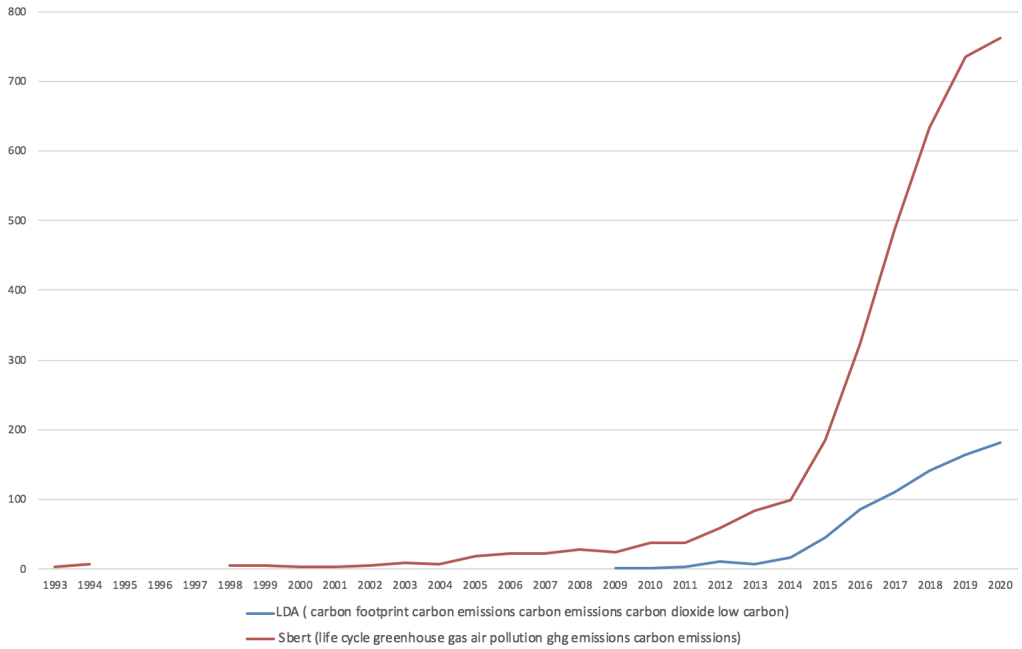


Figure 5.12: Comparison of topic trend analysis of carbon related topics from both the approaches

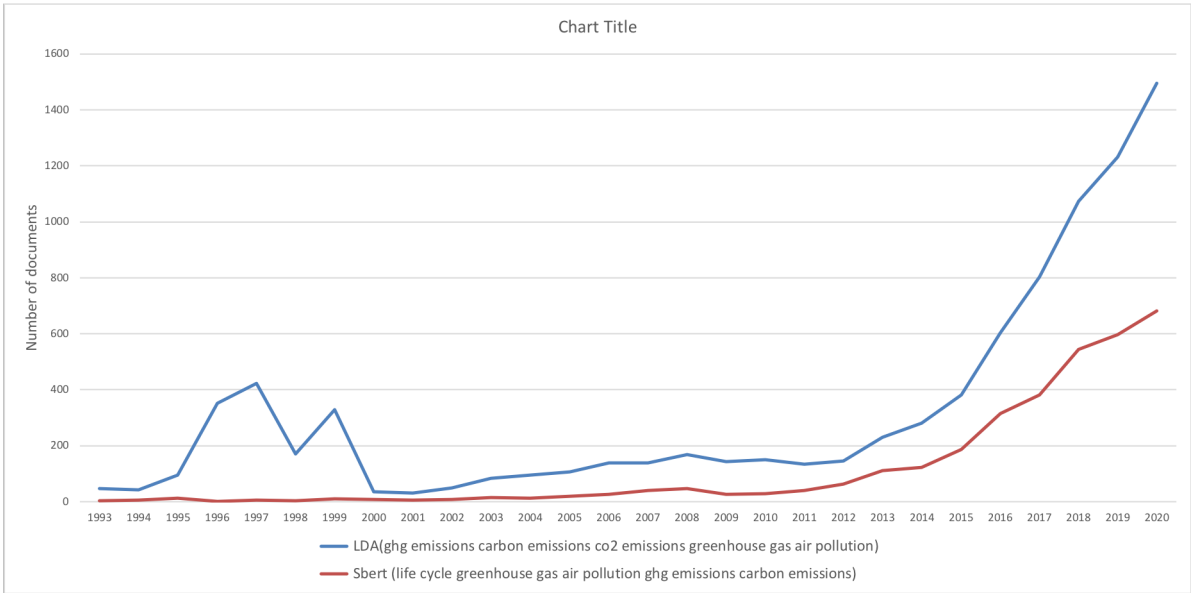


Figure 5.13: Comparison of topic trend analysis of sustainable development and circular economy related topics from both the approaches over the years

6 Discussion

Different methods and approaches were discussed in chapter 4 to achieve the aims of the thesis, i.e., to find an unsupervised solution to get semantic topics from a large text corpus and then do a trend analysis of these topics over the years. And the results from these approaches were also discussed in 5. In this chapter, we will focus on having a conclusive discussion based on the results and methods and will also discuss the scope of the work done in this thesis in a wider context.

6.1 Methods and Results

To achieve the first aim of this thesis project, i.e., An unsupervised semantic topic modeling, we started with traditional machine learning approaches like LDA and then moved to leverage transformer-based pre-trained language models. Regarding data pre-processing, we referred to this paper [6], which had done many experiments to conclude that apart from removing common English keywords, any other kind of corpus specific word removal does not improve the quality of topic models and even pre-processing tasks like stemming and lemmatization can, in fact, lead to losing important information when it comes to topic modeling. This information was very useful for us as having minimum pre-processing tasks helped us in keeping our methodology as unsupervised as possible.

In the traditional machine learning approach, the most popular method for topic modeling, which is LDA [19], was used. LDA topic modeling was first done on the complete data, which included the abstract text of all the documents. This approach gave an idea of the dataset but did not yield very meaningful results. Also, it was not feasible to generate an interpretable trend analysis using this method. Looking at the vast size of the text corpus and the number of varied research areas included in the corpus, it made sense to perform some kind of clustering among the documents and then analyze those individual clusters independently for topics. This thought process was confirmed by the paper [5]. While focusing on traditional approaches, the document clustering was done based on TF-IDF score and once the clusters were formed, the topic modeling was done using LDA and visualization of the cluster was done using TF-IDF based word clouds. This technique of document clustering resulted in much more defined and informative results. The clusters were defined by LDA topics and these topics were further used for trend analysis over the years based on the published date available for each document. Hence, a complete topic trend analysis approach

was devised. Some trend analysis results from this approach are shared and discussed in section 5.2. The trend charts achieved from this approach provided insightful results.

The traditional approaches have some limitations. TF-IDF score does not capture the context of the word and LDA needs number of topics and words per topic to be provided apriori. To move away from these limitations, the contextual pre-trained embeddings by Google's BERT were leveraged [2]. Since our task focuses on using these pre-trained embeddings for document clustering based on similarity another algorithm Sentence-BERT [28], which is build upon BERT was used. Sentence-BERT provides better sentence embeddings than BERT which is more suitable for our data and is also lot more computationally faster.

In our final approach, the abstracts of documents are pre-processed and then Sentence-BERT embedding are generated for each abstract. These embeddings are arrays of size 768. These embeddings are then clustered using K-means clustering with cosine similarity measure. Once the clusters are created, they are represented using words with highest TF-IDF score and LDA topics are extracted from each cluster. These topics were further used to analyze trends over the years in the corpus. Hence deriving a complete end-to-end unsupervised solution for semantic topic trend modeling. This method is discussed in detail in section 4.3.2 and results from this methodology are shared in section 5.3. Using these results many meaningful inferences about the corpus can be made. In addition to providing an overview of the corpus and serving as a technique to summarize the large text corpora, these trend charts can answer questions like when a particular topic started emerging in the corpus? How the popularity of a topic is changing over the years? How are some of the topics performing compared to last year? What are the most popular topics in last 5 years? .

We got some very meaningful results from the discussed approaches. Both traditional approaches and pre-trained language model based approaches had their own pros and cons.

6.2 The Work In A Wider Context

This work can be implemented to any kind of large text corpus to get an overview of topics in the corpus. It can be used on temporal textual data in any domain and can be used on short or large text. It can be used to summarize large dataset of research texts or news articles or social media posts or reviews.

7 Conclusion

This thesis project concludes with providing a solution based on the goals of the project. Given a large text corpora, this thesis had a two-step goal, of first doing an unsupervised semantic topic modeling which should generate semantically meaningful human-understandable topics and a technique to visualize them. The Second goal was to do a topic trend analysis. We conclude with an approach to achieve this using latest pre-trained transformer-based language models and hence leveraging the power of transfer learning in machine learning.

7.1 Conclusion on Research Questions

In this section, we discuss the conclusions on the research questions we started this project with.

1. Is it possible to find human-interpretable topics from a large text corpus using traditional statistical and machine learning approaches and get meaningful trend analysis of these topics over time?

   It is possible to get topics and their trend analysis using traditional machine learning techniques but these topics are not semantically meaningful. The traditional approaches come with some other limitations as well but it is possible to achieve these results using techniques like TF-IDF, LDA and K-means clustering. This method/approach is discussed in detail in section 4.2.3 and the results from this approach are discussed in section 5.2

2. Can latest pre-trained language models (e.g. BERT and it's variants) be leveraged to find semantic topics? What kind of impact having/including the word/sentence contextual embeddings from the transformer-based pre-trained language models have on the quality of topic models? Does leveraging the knowledge of these models improve the quality of extracted topics or not?

   Yes, pre-trained language models like BERT and Sentence-BERT be used to extract topics. Though very little research has been done and is available which leverages these models for topic modeling. Another challenge which was faced during the implementation was evaluating the generated topics. There are few techniques which exist for

evaluating probabilistic topic models but not much options are available to evaluate how semantic a topic is. The results of this thesis project were discussed with the domain expert and based on their feedback, they were found to be reasonable and made sense. This method/approach is discussed in detail in section 4.3 and the results from this approach are discussed in section 5.3

3. Can the results from these methodologies be compared with each other and an optimal approach be decided. Can some conclusion be reached on comparative analysis of results from these two approaches.

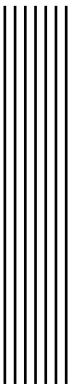
   These results can be compared based on similar topics. The comparison of results is discussed in section 5.5 but it is difficult to do a comparative analysis of the two approaches based on the results.

4. What could be the solution for visualizing topics and visualizing the trend changes in the topics and analysis of results given the large temporal textual data?

   For visualizing topics TF-IDF based word clouds can be an effective tools and for visualizing the trend changes the approach based on dividing documents into clusters and then assigning topics to these clusters (either LDA based topics or top TF-IDF words considered as topics) was used. Once each document is associated with a topic then we can derive trend analysis as documents have published date associated with them. These results are demonstrated in section 5.3

7.2 Future Work

We plan to devise a technique which does not use TF-IDF and LDA. Also we plan to investigate other clustering techniques so that the number of cluster do not have to be provided apriori. It will also be useful to investigate the clusters based on pre-trained word embeddings and why and how they are different from TF-IDF based clusters. More work can be done on finding an evaluation process to evaluate the quality of topics i.e. the interpretability and coherence of topics. This will need detailed experiments and analysis based on various kinds of text corpus and comparing the results from them.



Bibliography

[1] *Journal of cleaner production*, https://www.journals.elsevier.com/journal-of-cleaner-production.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] D. M. Blei and J. D. Lafferty, "Topic models," *Text mining: classification, clustering, and applications*, vol. 10, no. 71, p. 34, 2009.

[4] I. Vayansky and S. A. Kumar, "A review of topic modeling methods," *Information Systems*, vol. 94, p. 101 582, 2020.

[5] R. Venkatesaramani, D. Downey, B. Malin, and Y. Vorobeychik, "A semantic cover approach for topic modeling," in *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, 2019, pp. 92–102.

[6] A. Schofield, M. Magnusson, L. Thompson, and D. Mimno, "Understanding text preprocessing for latent dirichlet allocation," in *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics*, vol. 2, 2017, pp. 432–436.

[7] W. Guo and M. Diab, "Semantic topic models: Combining word distributional statistics and dictionary definitions," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 552–561.

[8] X. Deng, R. Smith, and G. Quintin, "Semi-supervised learning approach to discover enterprise user insights from feedback and support," *arXiv preprint arXiv:2007.09303*, 2020.

[9] A. B. Dieng, F. J. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020.

[10] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, 2010, pp. 100–108.

[11] D. Sahrawat, D. Mahata, M. Kulkarni, H. Zhang, R. Gosangi, A. Stent, A. Sharma, Y. Kumar, R. R. Shah, and R. Zimmermann, "Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings," *arXiv preprint arXiv:1910.08840*, 2019.

[12] D. Hall, D. Jurafsky, and C. D. Manning, "Studying the history of ideas using topic models," in *Proceedings of the 2008 conference on empirical methods in natural language processing*, 2008, pp. 363–371.

[13] J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, and D. Blei, "Reading tea leaves: How humans interpret topic models," *Advances in neural information processing systems*, vol. 22, pp. 288–296, 2009.

[14] T. Y. Lee, A. Smith, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater, "The human touch: How non-expert users perceive, interpret, and fix topic models," *International Journal of Human-Computer Studies*, vol. 105, pp. 28–42, 2017.

[15] W. contributors, *Topic model — wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Topic_model&oldid=1000434813, 2021.

[16] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[17] L. Liu, L. Tang, W. Dong, S. Yao, and W. Zhou, "An overview of topic modeling and its current applications in bioinformatics," *SpringerPlus*, vol. 5, no. 1, p. 1608, 2016.

[18] Y. Feng and M. Lapata, "Topic models for image annotation and text illustration," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 831–839.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[20] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50–57.

[21] M. Gormley, *Lecture notes in topic modeling*, https://www.cs.cmu.edu/~mgormley/courses/10701-f16/slides/lecture20-topic-models.pdf, Nov. 2016.

[22] W. contributors, *Word embedding — wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Word_embedding&oldid=998250451, 2021.

[23] ——, *Cosine similarity — wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Cosine_similarity&oldid=1000382256, 2021.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, pp. 3111–3119, 2013.

[25] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[26] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[27] B. Stecanella, *What is tf-idf*, https://monkeylearn.com/blog/what-is-tf-idf/.

[28] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[29] D. Jurafsky and J. H. Martin, *Speech and language processing*. https://web.stanford.edu/~jurafsky/slp3/3.pdf, 2020.

[30] J. L. Boyd-Graber, Y. Hu, D. Mimno, *et al.*, *Applications of topic models*. now Publishers Incorporated, 2017, vol. 11.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[32] J. Alammar, *The illustrated transformer*, `http://jalammar.github.io/illustrated-transformer/`.

[33] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[35] M. S. Z. RIZVI, *Demystifying bert*, `https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/`.

[36] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to data mining (second edition)*, `https://www-users.cs.umn.edu/~kumar001/dmbook/ch8.pdf`.

[37] W. contributors, *K-means clustering — wikipedia, the free encyclopedia*, `https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=1001316455`, 2021.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[39] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, arXiv–1910, 2019.