Expertise
and insight
for the future

Mai Vu

Building Topic Modelling on Theses Abstracts Data

Thesis Supervisors Finder for Students

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

18 November 2021

Author	Mai Vu
Title	Building Topic Modelling on Theses Abstracts Data: Thesis Supervisors Finder for Students
Number of Pages	39 pages
Date	18 November 2021
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Smart Systems
Instructors	Aarne Klemetti, Researching Lecturer Janne Kauttonen, Staff Researcher

This thesis focuses on topic modeling on theses data from Finnish Universities of Applied Sciences students collected from Theseus. The main objective of this thesis is to extract valuable information and create useful applications using given data.

The thesis starts with the project background, mentioning the project's resources, including the data and the CSC's supercomputer used to run the algorithms. Next is the comprehensive theoretical background that describes NLP and its methodical approach, as well as topic modeling and preprocessing techniques. This section provides the foundation of knowledge for the implementation of the LDA and DTM algorithms in the next chapter. The implementation is done on Anaconda and supercomputer using Python language, and `NLTK`, `gensim` library. Thereafter, the results of tested models are reviewed and compared to find out the most suitable; two of those are noteworthy. Eventually, a discussion on improving the project is presented. Afterward, a small test is made to build the thesis supervisor finder for students as proof of concept using the model. The last section is the conclusion of this research.

In short, this thesis is written in hopes that it contributes to researches of applying AI to gain more insights to improve teaching quality and student experience in the higher education sector.

Keywords	Natural Language Processing, Python, Topic Modeling

Contents

List of Abbreviations

3UAS	3 Universities of Applied Sciences (Haaga-Helia, Laurea, and Metropolia).
AI	Artificial Intelligence.
ANN	Artificial Neural Network.
API	Application Programming Interface.
BERT	Bidirectional Encoder Representations from Transformers.
DL	Deep Learning.
DTM	Dynamic Topic Modeling.
GPT	Generative Pre-trained Transformer.
HDP	Hierarchical Dirichlet Process.
hLDA	Hierarchical Latent Dirichlet Allocation.
HTML	HyperText Markup Language.
IT	Information Technology.
LDA	Latent Dirichlet Allocation.
LSA	Latent Semantic Analysis.
ML	Machine Learning.
NaN	Not A Number.
NLP	Natural Language Processing.
NLTK	Natural Language Toolkit.
NMF	Non-negative Matrix Factorization.
SLURM	Simple Linux Utility for Resource Management.
SVD	Singular Value Decomposition.
TF-IDF	Term Frequency – Inverse Document Frequency.

1 Introduction

Information Technology (IT) are rapidly transforming the education industry, bringing about a complete revolution. Especially as the Covid-19 pandemic is still happening worldwide, social distancing restriction is applied to ensure society's health but prevent students from going to school physically. It boosts the existing accelerating trend of transforming education infrastructure and study material from offline to online, or in other words, deviating from the traditional access to digitalization. Having more and more data available, modern Artificial Intelligence (AI) technologies are added to exploit the potential of the enormous data. The movement continues with the estimated $6.1 billion spendings on AI in education in 2025 [1].

One of the growing interests is AI-utilized Natural Language Processing (NLP) products. For example, WriteLab, a NLP-based tool, helps students to improve their writing before submitting, gaining popularity in both K-12 and higher education institutions [2]. Another successful application is Gradescope, which instantly gives students feedback on their exams and supports teachers in the grading process [3]. Those development lessens manual operation and minimizes delays, allowing teachers to spend more time with students and improve teaching materials.

Similarly, the task in this thesis is to apply NLP in order to make use of the available theses data, mainly focusing on topic modeling. After the data exploration and preprocessing phase, the most well-known algorithms: Latent Dirichlet Allocation (LDA) and Dynamic Topic Modeling (DTM) are performed and then their results are compared. In addition, the ambition is to explore how this study can be employed to create a practical application in real life.

An empirical observation is that nowadays, thesis topics can hardly be classified uniquely into only one topic, for instance, IT applications in the healthcare section or laws in business. A theory is that the trained topic models can be used to distribute thesis supervisors into a mixture of different research areas. Thus, in the future, students can match with a suitable supervisor based on their topics of interest.

A potential goal of this thesis is to present a matching tool for students to find their thesis supervisors as an application of NLP and topic modeling. The use case contributes to further research development of 3UAS, desiring to increase study quality and student experience by utilizing AI in the higher education industry.

2 Project Background

2.1 Theses Data

Theseus is an open repository for theses and publications of the Finnish Universities of Applied Sciences, owned by Arene, the Rectors' Conference of Finnish Universities of Applied Sciences. Its purpose is to provide an accessible and common platform to upload and examine academic publications as a digitalized transformation. It is worth noticing that students uploading their papers can select a Creative Commons license. [4].

For both Open Access and Restricted theses, their abstracts, used later in this thesis, can be accessed openly through the system. The data used in this thesis is scraped from the Theseus website by supervisor Janne Kauttonen. Only theses information from 2009 to 2020 are collected. A part of this scraped data is shown in figure 1.

	url	stream_url	is_downloaded	pdf_name	txt_url
index					
19606	https://www.theseus.fi/handle/10024/19606	https://www.theseus.fi/bitstream/handle/10024/...	False	[jamk_1250169708_1.pdf]	[https://www.theseus.f
1810	https://www.theseus.fi/handle/10024/1810	https://www.theseus.fi/bitstream/handle/10024/...	False	[Pitkaranta_Joni.pdf]	[https://www.theseus.f
19612	https://www.theseus.fi/handle/10024/19612	https://www.theseus.fi/bitstream/handle/10024/...	False	[jamk_1246449108_2.pdf]	[https://www.theseus.f

Figure 1: Samples of the first collected theses data.

It includes many attributes, and most of them do not contain beneficial information. Thus, it is crucial to define which attributes are useful for the model training and start the cleaning phase. As topic modeling only deals with texts, those unrelated attributes such as `url` can be removed. Furthermore, a majority of theses are in Finnish, which makes the data processing stage much more complicated. As topic modeling only focuses on important words, not whole sentences, thus, precise translation is deemed unnecessary. Instead, abstracts in Finnish are automatically translated into English, thanks to Janne Kauttonen's works. This second metadata is designed for this topic modeling problem. Details of the dataset are covered in section 4.

2.2 CSC Puhti Service

CSC – IT Center for Science is a non-profit company owned by the Finnish state and Finnish Universities and Universities of Applied Sciences. CSC provides various solutions and services for data management; one of those is supercomputers for computation and analysis purposes. Puhti is one of their supercomputers, also intentionally for Machine Learning (ML) and AI workloads. More information about the Puhti service can be found on the CSC website [5].

3 Theoretical Background

The third section explains Natural Language Processing (NLP) in AI/ML, describes the foundation of used methods and techniques in this project, as well as goes through the interpretation of steps in a typical text preprocessing pipeline.

3.1 Natural Language Processing

Natural Language Processing (NLP) can be interpreted as an intersection of AI and linguistics, whose intention is to exploit natural language data to develop machines that understand and respond to human languages. Natural language data, including both text and speech, is unstructured data. This indicates that it cannot fit precisely into a relational database. Therefore, it is more complicated to handle and analyze. However, natural language data contains a significant amount of information as human languages are highly ambiguous, reflect cultural knowledge, and embody human evolution. Those are also the challenges of NLP since it is hard to properly represent the hidden, abstract rules and the meaning of languages for computers to understand. Scientists and researchers have been working on the NLP topic since the 1950s, aiming to create machines that can interact with humans. The last decade observes the evolutionary development of this topic in academic research, growing from heuristic to ML, and lastly, Deep Learning (DL) for NLP. [6].

Firstly, heuristic- or rule-based methods had been used by computational linguists to tackle NLP tasks. This approach requires sophisticated feature engineering, which strongly needs domain knowledge and intensive specialized study in particular problems

to extract useful information from the data and form rules to implement into the program. Simple examples are that words starting with a capital letter are proper nouns, although this is not always the case, or regular expressions to find substrings in the corpus. For more superior rule-based methods, additional resources such as dictionaries and thesauruses can be employed. For example, the lexicon-based sentiment analysis uses a predefined vocabulary to calculate the sentiment of a text by counting positive and negative words. Furthermore, the semantic relationships between words, such as synonyms, hyponyms, meronyms, etc., can be mapped to create an advanced database. The heuristic-based approach is beneficial in some uncomplicated systems and simple cases, such as for early text review or to assist the data preprocessing for ML and DL models. For a more comprehensive system, the handcrafted process of formulating rules takes incredible time and effort. Additionally, its accuracy can hardly increase over time. [6].

Secondly, different from explicitly programming NLP systems based on observational rules, ML approach takes advantage of the available data as learning materials for computers to learn rules and patterns. This system is superior to the heuristic one as it generalizes better and does not involve investigating hardcoded rules. Generally, there are three types of ML techniques: supervised learning, unsupervised learning, and reinforcement learning. The first type of learning utilizes labeled data to generate systems that predict data for the future. [7]. Examples for supervised learning NLP applications are the spam email filter [8] (classification task) or the stock's price predictor based on stock-related discussions on social media (regression analysis) [6]. Both examples need data that includes spam-or-not values and stock prices respectively for every sample to train the ML models. Dissimilarly, unsupervised learning is applied when there are no targets to predict but rather to explore the hidden structures of the data. Examples of unsupervised learning tasks are clustering and dimensionality reduction. Lastly, the reinforcement learning approach attempts to solve interactive problems by developing self-learn systems that take feedback or rewards to improve. [7]. This method is well-known in teaching computers to play games, for example, chess [9]. Besides, it is applied in other sections as well, such as automation [10], healthcare [11], trading [12]. In NLP, reinforcement learning is used for many different tasks, namely, question answering [13], article summarization [14]. Depending on the data and the goal, a suitable technique can be tailored and implemented.

Thirdly, the DL approach is the most advanced solution at the moment. It can be viewed as a subfield of ML since it also needs data to draw patterns. DL is different from ML in that DL is built from Artificial Neural Network (ANN) architecture, attempting to simulate the biological human neurons. [6]. DL in NLP research started to receive more intense interest from the early 2010s. Groundbreakings were in 2018 when Google released Bidirectional Encoder Representations from Transformers (BERT) [15], in 2019 with Generative Pre-trained Transformer (GPT) 2 from OpenAI [16], and lastly, emerging from its previous model, GPT-3 in 2020 [17]. [18]. Not only does the above system development mark significant milestones of the DL-based for NLP, but it also is used as the backbone of numerous further projects due to the fact that DL is flexible in the adaptability to different realms and applications. Specifically, a system can adopt a BERT or GPT pre-trained network to support solving downstream tasks, particular problems, for example, machine translation or article generation problems. This method is called transfer learning. As a result, the task is simplified; thus, significantly reducing training time yet delivering outstanding performance. [6].

The usage of the above approaches should be chosen on a case-by-case basis to ensure optimized results. For example, although DL is state of the art in solving various NLP problems, outperforming traditional ML methods, DL is not always preferred in all scenarios, especially in contemporary commercial applications, as companies often desire to fully understand the technology and its results. While traditional approaches based on rules and statistics are much more straightforward to interpret, DL models act like black boxes. Potential high cost to produce DL-based NLP solutions is the other primary reason. It is costly because, firstly, DL is such a complex network that it requires a significant amount of data, which is hard and expensive to collect. Training with small dataset results in model overfitting, meaning poor and inconsistent performance in the future with unseen data. Secondly, the cost of powerful and specialized hardware to train and maintain such heavy DL models is also huge. In contrast, the rule-based or ML approach is selected simply because its overall cost is smaller. Furthermore, it is faster to obtain results, is much more transparent, and has direct algorithms; still, the performance is acceptable or even comparable to DL models. [6]. Due to the above reasons, this thesis fully focuses on the ML approach for this particular NLP task.

The expansion of NLP in the academic research side also leads to an escalation in

social and commercial uses in various industries. For instance, as the Covid-19 pandemic has been a dominant topic in the past years, there are increasing numbers of Covid-19 related unreliable and fake news. NLP is used to identify those to control its harmful spread [19] [20]. Another example in social media is using NLP to detect early signs of cyberbullying [21]. Those applications greatly boost clean digital communities and increase awareness and caution of the healthy online environment for all users. In addition, NLP also is employed in the healthcare sector, such as to spot the onset of Alzheimer'sAlzheimer's disease in its early stage, giving more time to ease the disease with early treatment [22]. Moreover, other NLP applications like Google Translate, Google Search, Grammarly, and intelligent virtual assistants such as Google Assistant, Apple's Siri, Amazon's Alexa, Microsoft's Cortana are widely common and extremely useful for daily human tasks [6]. Especially in the business area, NLP are playing a critical role in fast and excellent customer services. It helps analyze and keep track of consumers' comments on the company's brand and provides customers with timely, personalized responses with chatbots [6]. From there, the company can improve its performance.

3.2 Topic Modeling

3.2.1 Definition

One of the most extensively applied of NLP applications in the industry for more than a decade is topic modeling. In a nutshell, topic modeling intends to discover latent topics or the global structure of a large, typically unlabeled, set of documents. [6] [23]. Especially when dealing with an enormous number of documents, as manually assigning them topics is costly, topic modeling can help gain great insights with less time and effort [6].

Topic modeling is developed based on the assumption that a topic can be interpreted as a probability distribution of a set of words or tokens. Likewise, a document does not fall into a unique topic but rather a statistical mixture of different topics. [23]. In other words, topic modeling can be considered as a soft clustering method. It is different from topic clustering in that the hard clustering technique is where a document can only belong to one cluster [24]. Noticeably, both tasks are also different from topic classification, which is a supervised learning one with labeled datasets (in particular, the topic of a document

is provided in the first place).

The result from topic modeling is a matrix of documents by topics. The matrix's cell indicates the probability that a particular document belongs to a specific topic. A topic here is not explicitly defined but rather a collection of themed words. Therefore, in the end, topics need to be defined by human intellectuals. The results can be considered as a dimensionality reduction product because the matrix is a kind of structured data (numerical) of the original unstructured data (texts). More specifically, a unique word needs one dimension to be represented. Therefore, the entire corpus needs millions of dimensions. On the other hand, topic modeling's matrix has significantly fewer dimensions as they represent a defined number of topics rather than numerous single words. [24].

The advantage of topic modeling is the potential of spotting remarkable hidden topics in the corpus. The downside is that the model performance cannot be precisely evaluated since there are no real targets to categorize documents. [24]. Nevertheless, topic modeling is an efficient method as it does not require much computing power. Consequently, it is used widely in NLP, especially in the first stage of analyzing corpora. [23].

3.2.2 Application

Topic modeling is actively used in various industries, primarily for automating laborious tasks. For instance, instead of having technicians collect and track customers' conversations and reviews of the company or its products over social media, these comments and trends can be automatically detected and classified by applying topic modeling. [6]. In particular, in the hospitality industry, by monitoring online reviews using topic modeling, accommodation owners can understand key drivers that affect customer satisfaction to improve them [25]. Similarly, a different study explores patients' thoughts of physicians to help doctors acquire a deeper understanding to enhance healthcare quality [26]. Moreover, topic modeling can assist in routing customer support tickets based on subjects, keywords, urgency, and in delivering them to the proper departments. Thus, the technicians benefit from the ease of manual workloads, which allow them to focus on more critical tasks and complicated problems. Hence, they can provide better services and products and help improve business results.

Extensively, topic modeling can be applied in other industries as well. For instance, chemistry-related topic modeling is developed to categorize large molecules into ”chemical topics” and illustrate their similarities [27]. Adopting the same approach, in biology, the technique groups different cell types into diverse ”chromatin topics” successfully [28]. Furthermore, topic modeling is employed in new technologies like blockchains to improve the analysis of its technological trend [29] [30]. Finally, topic modeling’s outputs can also be used as the inputs for topic classification [6].

In summary, topic modeling’s powerful impact on analyzing corpus makes it an attractive tool to computer engineers and business owners. In the last decades, this research section has witnessed a revolutionary transformation thanks to the novel approach: using probabilistic and ANN/DL [31] [32] [33] instead of linear-algebra-based methods to train topic models [23].

3.2.3 Machine Learning Methods

The fundamental principle of topic modeling is applying statistics to words in the corpus so that the latent topics can be discovered [23] [24]. Three widely used statistical-based algorithms, namely Non-negative Matrix Factorization (NMF), Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) are described below. All of them start with the document-term matrix as input. This matrix is initially generated from the given corpus by calculating the (scaled) word frequencies. Each row represents a word, while each column corresponds to a document in the corpus. Since a document often only holds a small portion of words in the dictionary of the corpus, this document-term matrix is sparse, meaning most of its cells’ values are equal to zero. [23]. A detailed description of the document-term matrix is provided in section 3.3.

Firstly, the most basic and most straightforward topic modeling algorithm is Non-negative Matrix Factorization (NMF). From the calculated document-term matrix (M), the hidden structure of the corpus (W) and relevance between words and topics (H) are computed. [23]. The document-term matrix M is a product of two matrices W and H:

$$M \approx W \cdot H \tag{1}$$

In terms of topic modeling, matrices on the right side of equation 1 can be interpreted as the document-topic matrix (W) and topic-term matrix (H). Matrix M has the same number of rows as matrix W and the same total of columns as matrix H. The number of topics, or the number of columns in matrix W aka the number of rows in matrix H, is initially chosen. Computing the exact solution for W and H from M is exceedingly expensive. The approximative solution is competent and adopted, which explains the approximate symbol. The bigger the number of total topics, the more accurate the approximate equation. [23].

Secondly, also advancing from linear algebra methods is Latent Semantic Analysis (LSA). Similar to the NMF approach, final outputs of LSA are document-topic and topic-term matrix. However, LSA employs a different mathematical technique called truncated Singular Value Decomposition (SVD) to get these 2 matrices. The truncated SVD equation is shown below, where M is the document-term matrix. U_t has the same interpretation and dimension as W matrix in formula 1, so forth V_t^* and H. Lastly, Σ_t is a diagonal matrix holding singular values. [23].

$$M \approx U_t \cdot \Sigma_t \cdot V_t^* \tag{2}$$

Generally, truncated SVD intends to reduce the dimension of the original matrix but to keep as much similarity as possible. In detail, this method picks the largest t singular values, representing in the middle matrix Σ_t of the equation. Only t columns of U and t rows of V^* corresponding to Σ_t matrix are kept. The rest of the matrices is dropped; hence, the dimension decreases. As for topic modeling, those singular values equate to latent topics. In the same way as NMF runs, for LSA, the number of topics needs to be determined in the beginning too. [23].

Thirdly, instead of using linear algebra methods like the two above formulas, Latent Dirichlet Allocation (LDA) algorithm is based on probability. Specifically, LDA algorithm is a generative model, meaning it creates documents by assembling words pulled out stochastically out of the Dirichlet distributions of topics. [23]. In the LDA algorithm, the two variables or hyperparameters, often called α and β, control the relevance of documents and topics, respectively. In detail, lower values of α and β result in a smaller set of topics that a document falls into and a smaller list of unique words for a topic accordingly or vice

versa. LDA is considered to be the most famous method for topic modeling [23]. More details on the algorithm can be found in the source material [34].

Furthermore, there are various variants of LDA [35] [36]. Among those, the most noticeable are Hierarchical Dirichlet Process (HDP) [37], DTM [38].

HDP is a revamped version of LDA, which can infer the number of topics through the learning phase, on the contrary of choosing the number of topics initially as the above three algorithms. More advanced than the LDA formula, HDP has an appended preceding level of Dirichlet processes for producing a nonparametric prior for the number of topics; hence, the ”hierarchical” word in its name. [39]. The method is beneficial considering that it might be tricky to foresee how many topics are in the unknown, enormous corpus. A variant of this is online variational inference for the HDP. It overcomes some drawbacks of the previous one, as this enhanced algorithm can easily handle huge datasets and streaming texts like web APIs. [37]. Because HDP is relatively new and has not been widely used yet [23], it is not considered in this project. Nevertheless, the complete details can be found in the original research here [39] [37].

DTM is superior to LDA because DTM observes how each topic evolved over a particular time interval. The original research paper can be found here [38]. As the collected data in this project are theses from 2009 to 2020, whether the main focus of a topic change over time and how different it could be are questions to be addressed. Thus, DTM is employed as an ideal technique for this problem.

3.2.4 Deep Learning for Topic Modeling

All ML-based approaches fundamentally treat the corpus as a bag-of-words to get the initial document-term matrix. It means those algorithms pay no attention to word order, synonyms, syntactic information, the semantics of words, etc. [31]. Attempting to conquer these difficulties, ANN/DL is applied as a cutting-edge technique for topic modeling, in particular, Top2Vec words embeddings [31], pre-trained transformer models such as BERT words embeddings to perform topic modeling [32]. This enhancement also allows totally omitting preprocessing phase (such as removing stop words, applying stemming or lemmatization), which are essential for ML methods. Similar to HDP, the number of topics

does not require before training. [31]. An excellent overview of neural topic models can be further studied here [33].

Despite the above advantages, DL is computationally expensive and requires extensive preparation. Additionally, this thesis is an early stage of the big project; therefore, it mainly focuses on implementing the two most intensively used and inexpensive methods for topic modeling: LDA and DTM.

3.3 Text Preprocessing

As mentioned earlier in the previous chapter, to apply ML-based topic modeling algorithms, corpus needs to be handled before training. All the steps of text preprocessing for topic modeling used in the later chapters are explained here. Most of them are also standard procedures when carrying out NLP tasks. The pipeline includes formalization, tokenization, stop word removal, stemming and lemmatization, n-grams, and, lastly, document-term matrix computation with word counts and Term Frequency – Inverse Document Frequency (TF-IDF).

Any ML models are built around the data. Indeed, data is the most invaluable asset at the core that developers have to ensure the data fed into the algorithms is cleaned. However, the raw data collected for the given task hardly meet this standard. For NLP, the original texts might includes HTML tags, Unicode, etc. Thus, the data or texts need to be formalized by removing unnecessary and irrelevant information, such as deleting punctuation, digits and applying lowercase to convert data into a uniform and usable form. [6].

The next important step in the pipeline is tokenization. Fundamentally, tokenization is the process of breaking raw strings into tokens. In the simplest cases, a token is a word that can be split easily by the whitespace character or by some other punctuation. However, tokenization might be different and more complicated depending on the collected data, area of expertise, the goal of NLP project, and the used language. Thus, custom tokenizer is more common in real-life applications. [6]. It can be created using regular expressions [23] or, more advanced, by applying ML methods [40].

After tokenization, an observation is that a chuck of tokens do not contain useful information [23]. As they normally appear with high frequencies, they even add noises into the data. Those tokens are called stop words. [40]. Specifically, they are articles (*a*, *an*, *the*, etc.), demonstratives (*this*, *that*, *these*, etc.), pronouns (*I*, *you*, *she*, *he*, etc.), auxiliary verbs (*am*, *is*, *are*, *was*, etc.), and so on. Most of the time, stop words need to be removed to expose the underlying content of the texts. In the most straight-forward cases, each token in the data is compared to a list of predefined stop words to mark it as a stop word and to delete it. [23].

Another observation is that, for English language, words can appear in different forms, for example *do*, *did*, *done*, or *nation*, *national*, *international*, to convey different meaning such as past tense, passive voice, etc. Those words are called inflected words: a group of words sharing the same root are named a word family. Even though inflected words are essential to interpret the content, they compose even more sparse data, or in other words, a bigger and loose dictionary. The cost of this is that extra resources and time are required to train NLP models. [23]. To reduce the dictionary size, stemming or lemmatization are executed on tokens. Those are techniques to transform a word to its root or form.

The main difference is that stemmed words might not be proper linguistic roots; however, lemmatized words are always [40]. For example, the stemmed word for *having* is *hav*, while lemmatized word is *have*. Special cases are, for instance, *am* and *better*. Stemmed word stays the same, but lemmatized ones are *be* and *good* respectively. Stemming technique simply trunks extra part of a word using predefined rules, such as removing *-es* or *-s* in plural forms, suffixes like *-able*, *-ness*, etc. Apart from that, lemmatization derives a word to its lemma or the root of its word family. [6]. Lemmatization is a little bit more complex as additional material such as a dictionary is needed [23]. In fine, the choice between stemming and lemmatization varies depending on the task [40]. For this project, lemmatization is used.

The tokenization process generally splits a single word into a token, which is technically called unigrams (n-grams with $n = 1$) [7]. However, single words might not clearly describe the content [23], for instance, *town* and *square* (*town square* is an open public place typically located in the center of a town), or *black* and *box* (*black box* describes a system that only the inputs and outputs can be inspected, without any information about the

internal process). Similar cases arise with other compound words and collocations as well. Thus, it would benefit the text analysis if tokens also include n contiguous words, meaning n-gram with $n > 1$). Nevertheless, as n increases, the number of n-grams tokens grows exponentially, raising the dictionary dimensions. Therefore, n is chosen to be less or equal to three in most cases. [6] [23].

Lastly, as described in the previous section, the document-term matrix is needed before training the topic models. In a way, this matrix is a format for converting unstructured data (texts) to numbers that computers can manage easily. Firstly, a custom dictionary is built by taking unique terms or tokens across all documents. Then, each token is assigned to a number or id, typically arranged in alphabetical order. From here, each document is vectorized using the dictionary in three ways: binary, word counts, TF-IDF as explained below. Finally, all the row vectors get stacked vertically accordingly to the term columns, forming the document-term matrix. [6] [23].

The most straightforward way to vectorized a document is by inspecting whether words appear in that document or not (binary stage, yes or no). A binary vector does not hold much information compared to the other two approaches. Advancingly, the word count vector takes into consideration the frequency of words in documents. For this method, the matrix is commonly called the bag-of-words model. [6] [23].

The above calculations have the drawback of not acknowledging the occurrence of words across the corpus [6]. For example, if the collection of documents is more about education, words like *teacher*, *student*, *school* are likely to appear in all documents, meaning those words are less important and valuable in analyzing the texts. They act like stop words but do not be removed in the previous steps. Term Frequency – Inverse Document Frequency (TF-IDF) is a method that considers this concern by multiplying $tf(t, d)$ and $idf(t, D)$ as in formula 3

$$tf\text{-}idf(t, d, D) = tf(t, d) * idf(t, D) \tag{3}$$

The first factor is the term frequency for term t in document d. There are several ways to calculate it, for instance, the raw counts as in the above approach [7]. A common formula for $tf(t, d)$ is shown below, which is the number of appearances of term t in document d

divided by the total of terms in the same document [6].

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{4}$$

The second factor is the highlight of the TF-IDF formula. It is used to reduce the weights of common words and increase the importance of rare words across the corpus [23] [24]. Same with the $tf(t,d)$, $idf(t,D)$ can be calculate with different formulas. Equation 5 is a standard one. Here, $idf(t,D)$ is the logarithm of the inverse fraction of the total number of documents that contain term t out of the total number of documents (N). [7].

$$idf(t,D) = log\frac{N}{|\{d \in D : t \in d\}|} \tag{5}$$

After all those preprocessing steps, the processed data is employed to perform the algorithms, entering the training stage. Finally, when the models are produced, the following measure is the evaluation phase, where the final models are examined and compared to conclude the impact of the used algorithms and the best models.

4 Data Preprocessing

Understanding the data is the first critical step of any machine learning project. Learning about the data distribution and statistics helps developers ensure the quality of the input dataset to achieve good final results. Specifically for NLP subject, frequency analysis of different (group of) words is an important stage. [23]. This section covers all matters related to handling the dataset. It gathers the early insights from the theses data during the exploration process, focuses on generated word clouds, and shows the text processing pipeline preparing for model development.

4.1 Data Exploration

The first metadata contains the scraped theses information from Theseus provided by supervisor Janne Kauttonen, consisting of more than 200 columns. The one examined here, modified and simplified from the first metadata, is smaller and tailored for the topic modeling purpose. The file is in `pickle` format. Samples are shown in figure 2.

	handle	year	original_language	organization	google_translated_en	en	google_translated_fi	fi
133602	10024/149869	2018	fi	Jyväskylä University of Applied Sciences	0	The aim of the thesis was to improve the spare...	0	Opinnäytetyön tavoitteena oli tehostaa varaosa...
150857	10024/291862	2019	fi	Turku University of Applied Sciences	0	The aim of this thesis was to clarify in which...	0	Tämän opinnäytetyön tavoitteena oli selvittää,...
96648	10024/108106	2016	fi	Haaga-Helia University of Applied Sciences	1	The aim of the work was to produce material on...	0	Työn tavoitteena oli tuottaa materiaalia vuoro...
176591	10024/342548	2020	fi	Turku University of Applied Sciences	0	The purpose of this thesis was to find out the...	0	Tämän opinnäytetyön tarkoituksena oli kartoitt...

Figure 2: Samples of the collected and processed theses data.

This tailored data consists of 174,955 theses published from Universities of Applied Sciences in Finland from 2009 to 2020. It contains 1 index column and 8 attribute columns. The `handle` acts as the primary key of the table. The following 3 columns show the year of publication, the original language of a thesis, and the organization (or can be interpreted as the author's university). The `en` and `fi` columns are the abstracts in English and Finnish respectively. Even though it is not always the case, Finnish theses are required to include English abstracts, while English theses do not have Finnish abstracts. Therefore, those abstracts are translated using Google API so that both columns are

filled. The `google_translated_en` and `google_translated_fi` columns indicate whether the abstracts is automatically translated in that language or original. Furthermore, the dataset has 10 Not A Number (NaN) values; 7 are in `original_language` and 3 are in `organization`. Since the focus of topic modeling is the texts, those NaN do not affect the model training.

Figure 3 illustrates the total number of theses throughout the years. In general, it rose steeply from 2009 to 2013 and increased gradually afterward. In 2020, the total number grew nearly 300% compared to 2009.

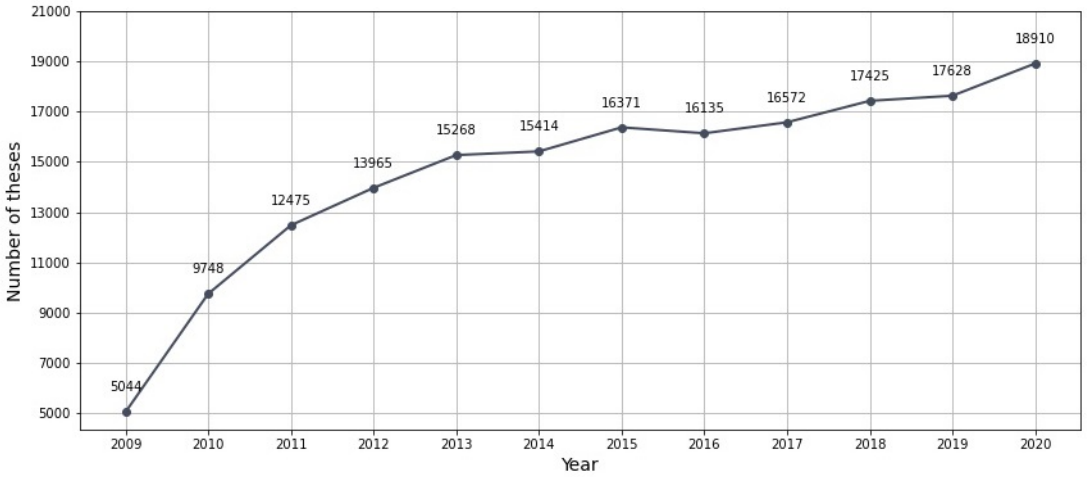


Figure 3: Diagram of the total number of collected theses in Universities of Applied Sciences in Finland from 2009 to 2020.

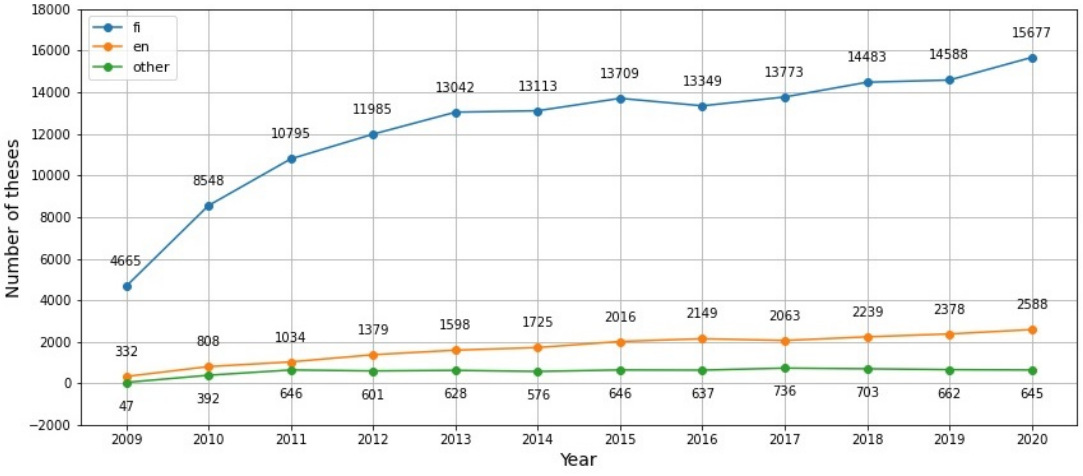


Figure 4: Diagram of the total number of collected theses in Universities of Applied Sciences in Finland from 2009 to 2020, based on Finnish, English, and other languages.

The line chart in figure 4 above shows in detail the distribution of the number of these in different original languages. The proportion of the theses written in English and Finnish increased over the 2009 - 2020 period. However, the English theses held a small portion

compared to the Finnish ones and did not significantly contribute to the overall trend. Indeed, reports initially in Finnish accounted for about 84.5%, while English was just only for nearly 12%.

For the `organization` attribute, as illustrated in figure 5, Metropolia University of Applied Sciences was the organization that published the most theses, nearly 21 thousand in 11 years. Additionally, there was a significant difference between top 1 and top 2 organizations, with more than 7000 theses.

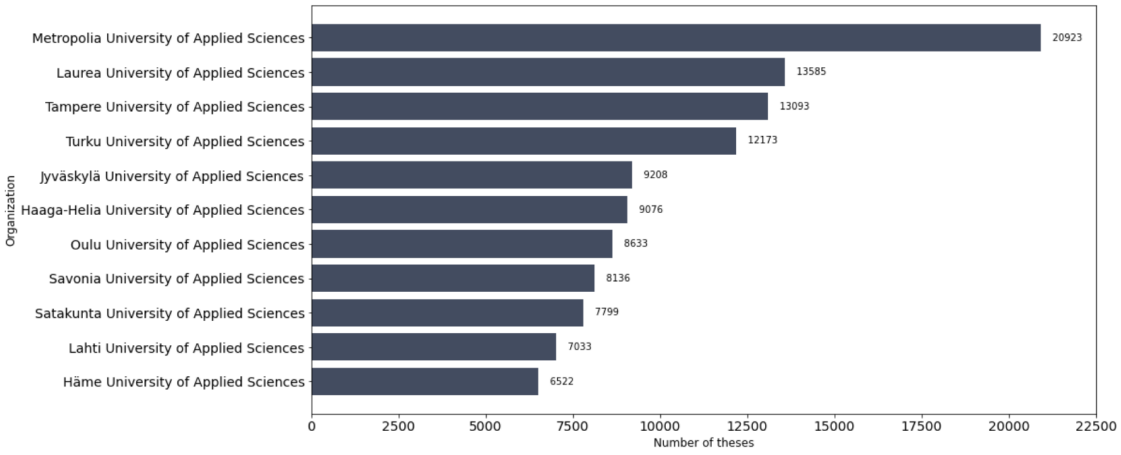


Figure 5: The top 11 most theses published organization sorted descendingly.

Another appealing aspect is the length of the abstract. The `en_fi_length` values are calculated by adding the total number of words in a thesis abstract in both languages, Finnish and English. As depicted in statistical summary table 1, the range of `en_fi_length` is from 136 to 3642 words. 50% of documents have from 338 to 510 words of both languages. Furthermore, the upper or 75% percentile value equals 510 words, showing that the 3642-words sample is quite extreme.

	mean	std	min	25%	50%	75%	max
en_fi_length	429.823	132.471	136	338	421	510	3642

Table 1: Statistical summary of the `en_fi_length` attribute.

Examining the `en_fi_length` values closely, as in figure 6, the dataset is nicely distributed. As the tailored data was modified to fit the topic modeling purpose, there are no outliers on the left side. Still, there are several outliers on the right one; two are quite larger than others. The more texts to work with, the better for training models; thus, those outliers on

the right are kept.

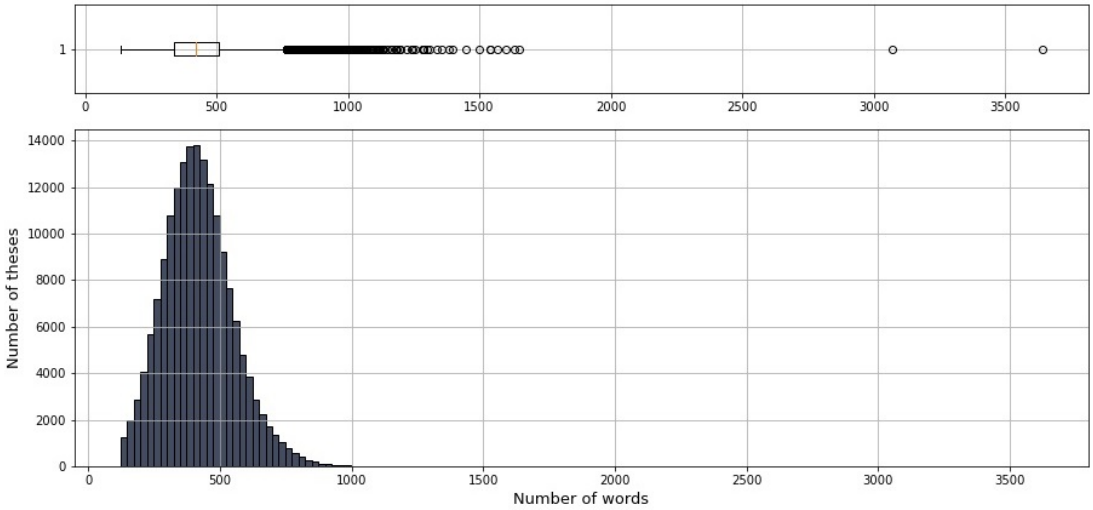


Figure 6: The box plot (above) and the histogram plot (below) showing the distribution of the `en_fi_length` attribute.

4.2 Text Preparation and Word Clouds

The last character to explore is the word frequency. Notably, Finnish is a typical agglutinative language with different uses of suffixal affixation. Because the text process for Finnish abstracts is completely different and much more complicated than English ones, only English versions are used for further analysis. The English abstracts are first tokenized. After that, punctuation and stopwords are removed, then the remaining words get lemmatized; both steps utilize `NLTK` (Natural Language Toolkit) library by using its stopwords dictionary and English lemmatizer. Figure 7 displays the top 11 most common words in all the English abstracts. Unsurprisingly, *thesis* is the most repeated term, up to more than 431 thousand times, meaning a document mentions *thesis* 2 to 3 times averagely. There are also some not-so-meaningful words, such as *also*, *result*, *used*, *part*, which will be deleted in upcoming steps.

A more common way to visualize word frequency is word clouds. Unlike the previous bar graph, word cloud illustrates the importance of a word by the size of the words and can show more words in one picture. [6]. Figure 8 shows a word cloud for the full vocabulary of the corpus. Words like *thesis*, *work*, *study*, *research* are ones of most frequency words, as it can also be seen from figure 7. Furthermore, with some modifications, a word cloud can present more detailed insights [6]. Thus, besides adding the stopwords dictionary using

`gensim` library, some further actions are performed as follows.

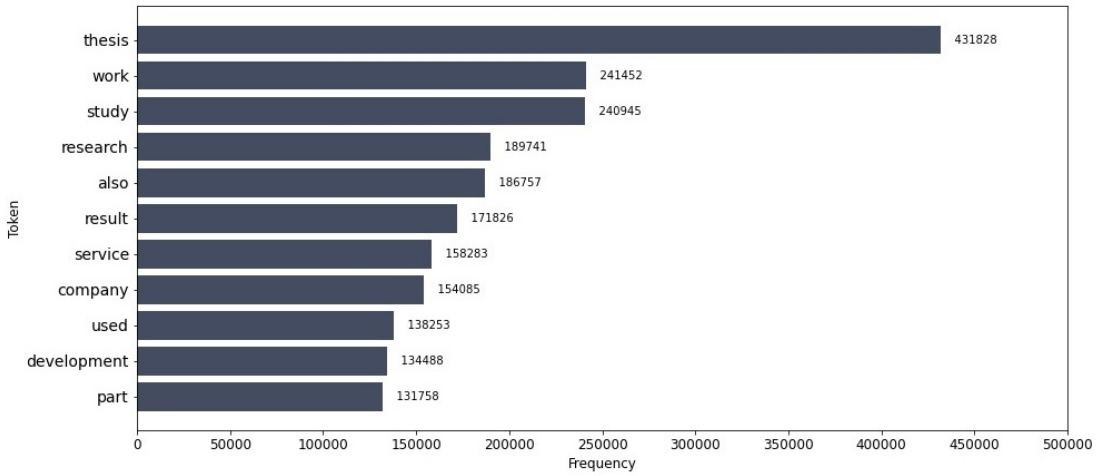


Figure 7: The top 11 most common words sorted.

Figure 8: Word cloud for the full vocabulary of the corpus.

Figure 9: Word cloud using same vocabulary, but excluding top 15 most frequency words.

The first issue to be fixed is abbreviations. In detail, since lowercasing is applied at the beginning, abbreviations like *IT* become *it*. This new form is considered to be a stopword and is deleted in later steps. To prevent this, an all-caps word does not be lowercased and is kept as a normal term. The most frequent abbreviations are shown in the below word cloud (figure 10). Some are related to universities (*UAS - University of Applied Sciences*, *JAMK - Jyväskylän Ammattikorkeakoulu*, *HAMK - Hämeen Ammattikorkeakoulu*), while others are highly associated to a industry (*IT - Information Technology*, *ICT - Information and Communications Technology*, *API - Application Programming Interface*). Thus, the training models would certainly benefit from including abbreviations in the dictionary.

Figure 10: Word cloud of abbreviations in theses in English from 2009 - 2020.

Secondly, the n-grams technique is applied with $n = 2$ as explained in 3.3. Its word cloud is given below (figure 11). Some of the compound words can be useful to distinct topics, such as *social_medium* (*medium* is the singular noun of *media* after lemmatization), *health_care*, *customer_satisfaction*. A combination of an abbreviation and bi-grams, for example, *SWOT_analysis* can be found in the dictionary too. From here, the corpus is processed well enough for the model implementation.

Figure 11: Word cloud of bi-grams in theses in English from 2009 - 2020.

An illustrative word cloud after applying two both techniques and TF-IDF weighting is shown in figure 9 above. Comparing these two word clouds, the one on the right shows a clearer view of different possible hidden topics in the corpus.

5 Implementation and Results

Creating the customized dictionary for the corpus is the first step of implementation. In this state, the preprocessed abstracts are such a dictionary and bag of words for all documents, plus words with outlier frequency (too rare or too common) and abstracts with too few tokens are removed. In addition, DTM algorithm requires an additional parameter:

`time_slice`, a list of integer. It indicates the number of documents in each selected period of time, which could be monthly, quarterly, or yearly. Here, the 2-year interval is chosen. However, the number of abstracts is limited for the first two years (5,044 in 2009 and 9,748 in 2010). Thus, abstracts written in the first three years are grouped together, so that each time frame can have an approximately equal number of theses. In short, the time frames are 2009 + 2010 + 2011, 2012 + 2013, 2014 + 2015, 2016 + 2017, 2018 + 2019, and 2020.

The second step is to find the optimal number of topics since both algorithms need it as input. This is accomplished by running the algorithms with different values of the number of topics and comparing their results, either by human interpretation/observation or coherence metrics. In short, coherence values indicate the semantic correlation between terms in a topic formed by the model. High coherence values mean that words in a particular topic are more similar and vice versa. Hence, the higher this value, the better the model. Coherence metrics are a common method to evaluate topic models. More details can be found in this research paper [41].

Figure 12 compares the run time to train LDA to that of DTM models. As shown, DTM run time is significant longer than the LDA ones. With the help from Puhti supercomputer from CSC, the DTM training process takes around one week to complete.

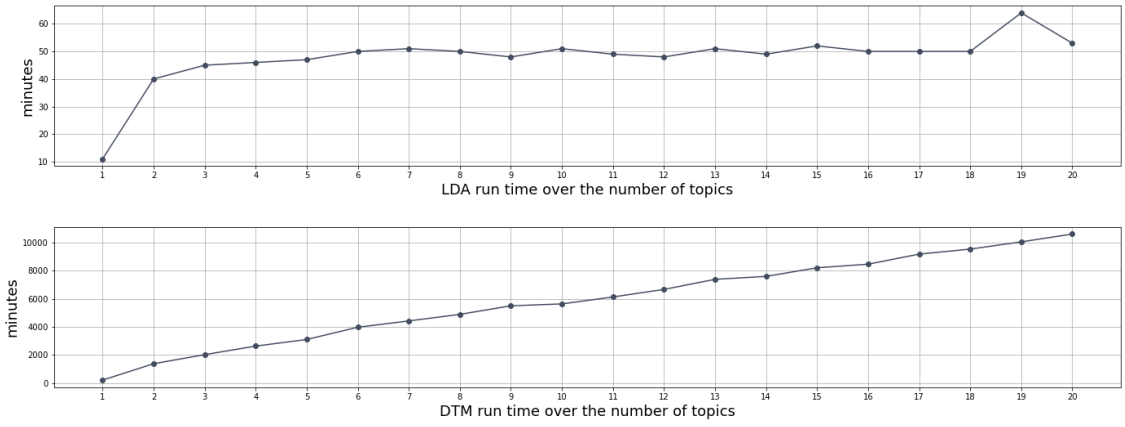


Figure 12: The run time of training LDA and DTM.

The library used to construct the dictionary, train models, and calculate the coherence metrics is `gensim` library, more specifically, `gensim.models.ldamodel`, `gensim.models.ldaseqmodel`, `gensim.models.coherencemodel`. To plot topics of a model, `pyLDAvis` libary is used. This libary provides an excellent and interactable topic modeling visualization.

5.1 LDA Model

As shown in figure 13, the optimal solution for the number of topics is the range of six and eight. The reason is below six, there are not enough room for topics to form clearly; while above eight, `c_v`, `c_uci`, and `c_npmi` values decrease or remain stable. Indeed, when plotting the 9-topic and 10-topic model as in figure 17 and figure 18 respectively, there are several partly-overlapping sections, and the common words generated for topics are trickier to interpret.

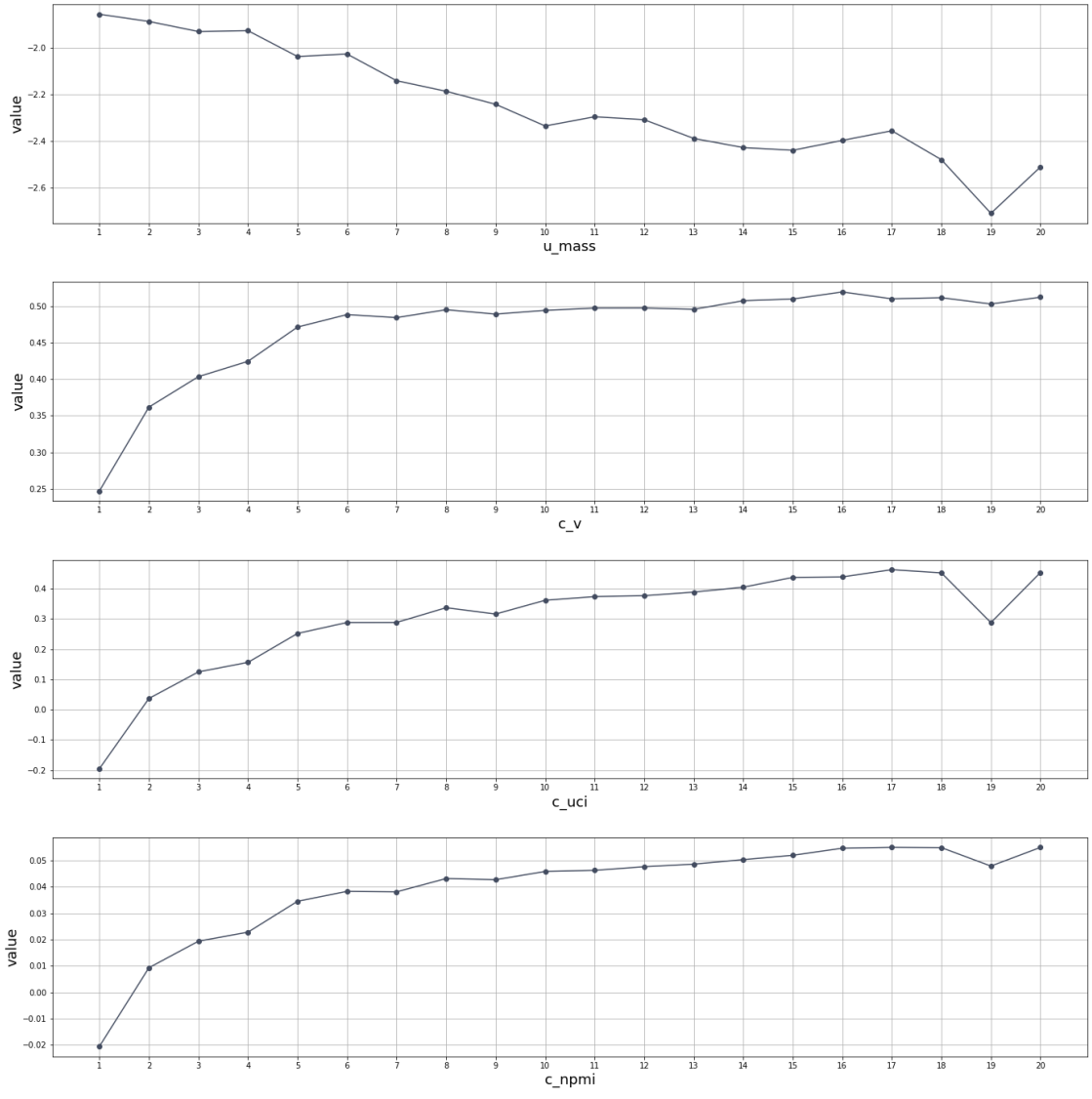


Figure 13: Four coherence metrics of LDA algorithms
across different values of the number of topics.

The LDA 6-topic (figure 14) and 7-topic (figure 15) models are quite alike, except for the existence of topic 7, which shares certain similarities, such as words like *customer*,

social_medium, to the cluster of topic 2 and topic 3. For the 7-topic model, the new topic 7 can be defined clearly as *marketing*; however, topic 2 becomes more difficult to identify. The word *business* appears in topic 2 and 7 with approximately equal frequency, partially in topic 3 as well.

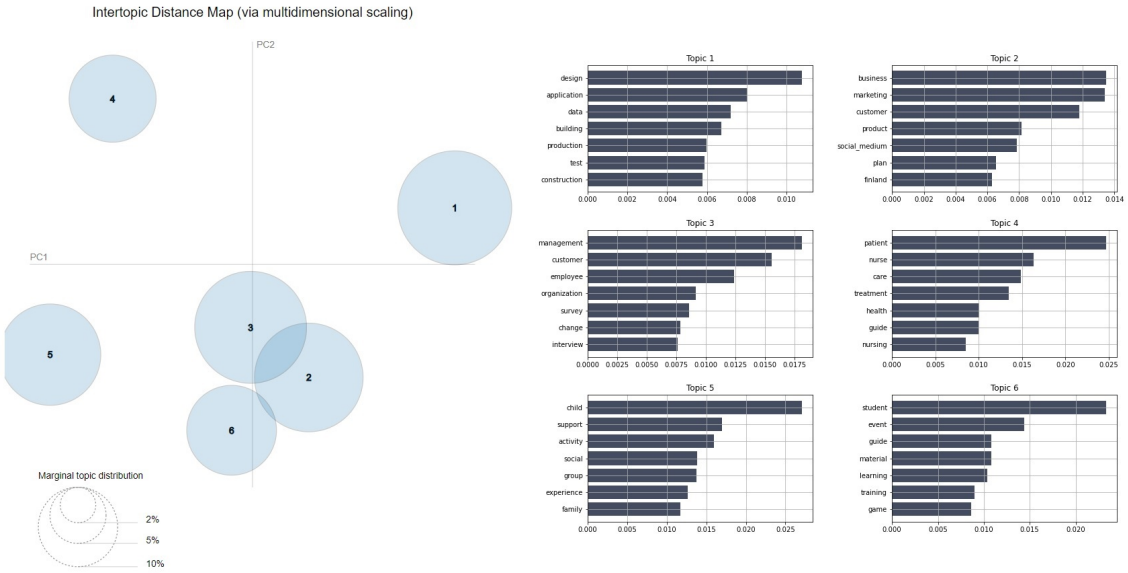


Figure 14: LDA model with 6 topics.

Figure 15: LDA model with 7 topics.

Similarly, topic 1 and 8 in the 8-topic model (figure 16) have obvious connections with topic 1 in figure 15. Indeed for the 6-topic and 7-topic model, topic 1 includes both *building_construction* and *information_technology*, which becomes much clearer in the 8-topic model. The topics of this model are interpreted as *building construction*, *Finnish/Finland*, *business management*, *health care*, *child development*, *education*,

marketing, *information technology* in numerical order. The 8-topic model plot shows that topics are nicely distributed with only two topics are partly overlapping. Thus, the optimal number of topics is eight.

Figure 16: LDA model with 8 topics.

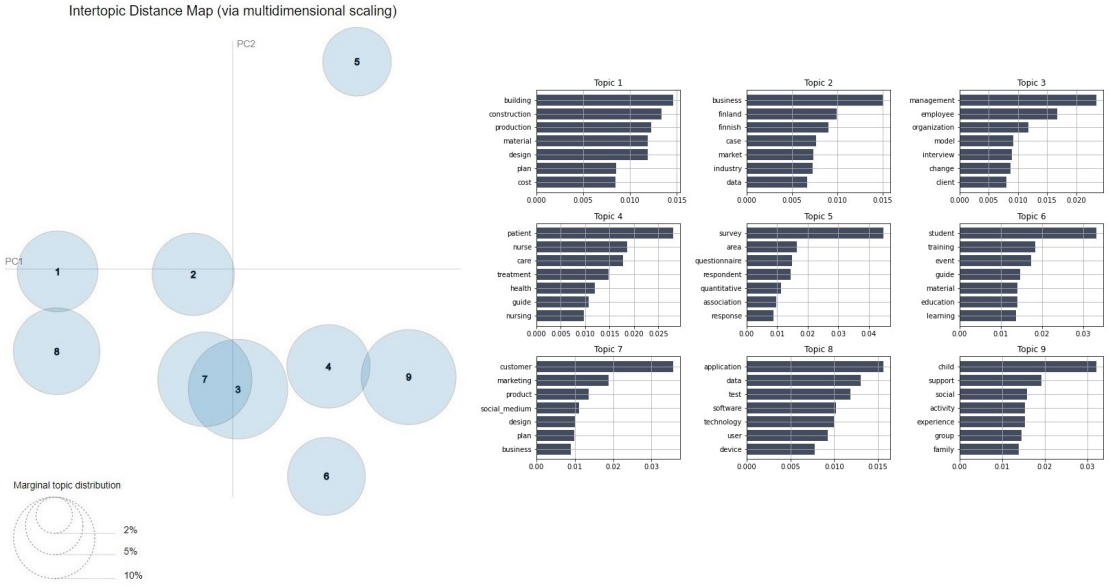


Figure 17: LDA model with 9 topics.

For double-checking, the 9-topic and 10-topic models are plotted in the above figures to compare with the 8-topic one. In the former model, topic 5 can hardly be considered a topic since surveys can fall in different fields such as business or education. Furthermore, for the 10-topic model, topics 6, 9, and 10 are arguably the same. In summary, models with nine and ten topics cannot identify hidden topics better than the 8-topic model.

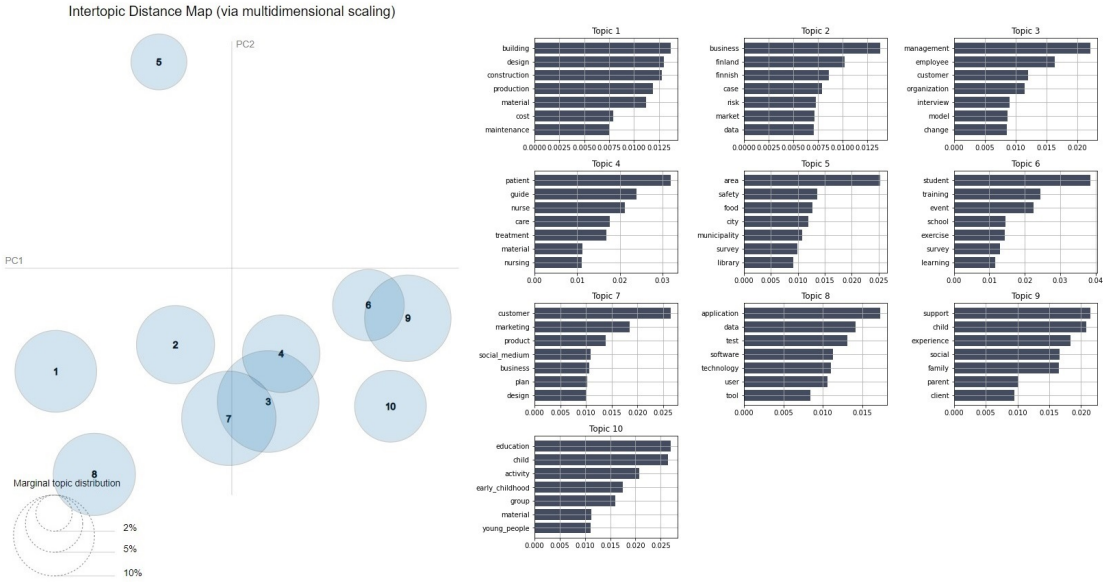
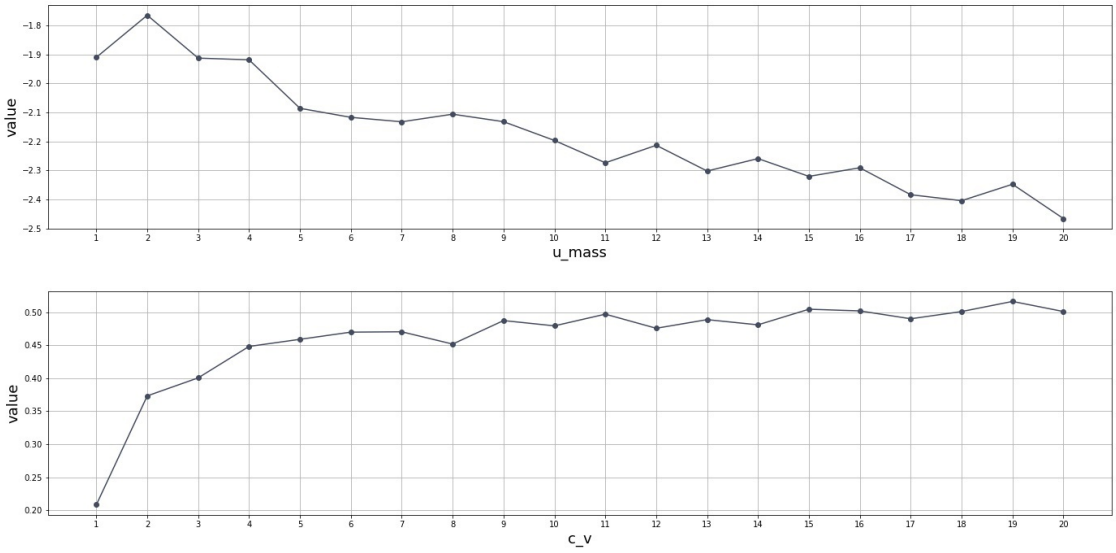


Figure 18: LDA model with 10 topics.

5.2 Dynamic Topic Model

Figure 19 plots four coherence metrics but with DTM models from one topic to twenty topics. The `u_mass` value drops dramatically from four topics to five topics, indicating the 5-topic model is favorable. For the next three models, those `u_mass`, `c_uci` and `c_npmi` values remain stable. The 8-topic model's `c_v` slightly drops compared to the last two. Models with more than nine topics have coherence metrics relatively the same; simultaneously, it is harder to make sense of numerous topics. For those reasons, models from five to nine topics are further analyzed below.



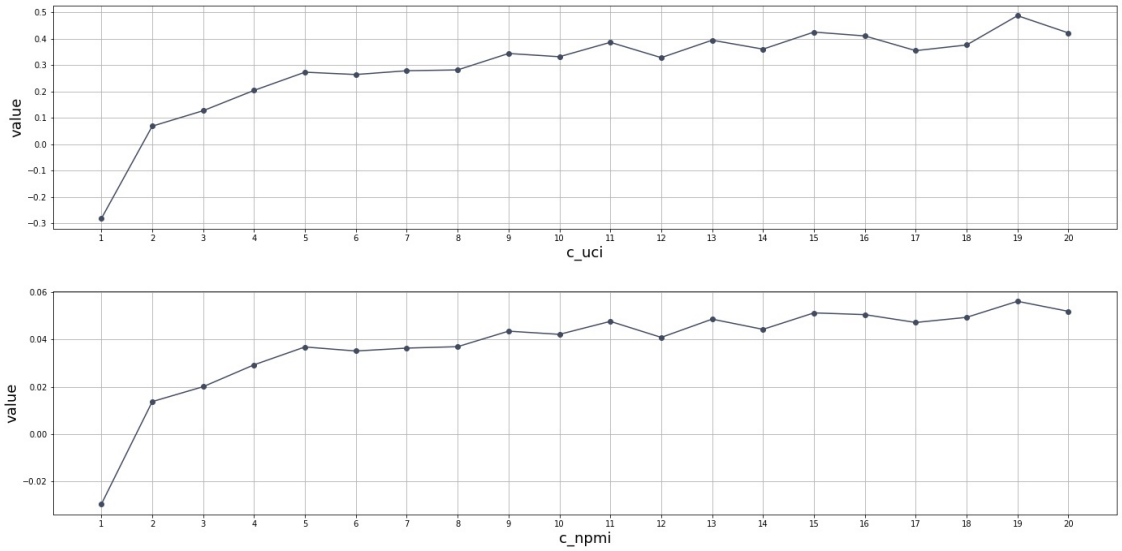


Figure 19: Four coherence metrics of DTM algorithms
across different values of the number of topics.

First glance at figure 20 shows that topics are separated from each other, which indicates it is a good model. The topics can be read as *information technology*, *business*, *building construction*, *education*, *health care* respectively. For topic 1 and 2, the word *design* and *customer* have an considerable impact of the corresponding topic, since the words' weights below drop significantly. Overall, this model classifies five topics properly in a general, satisfactory way.

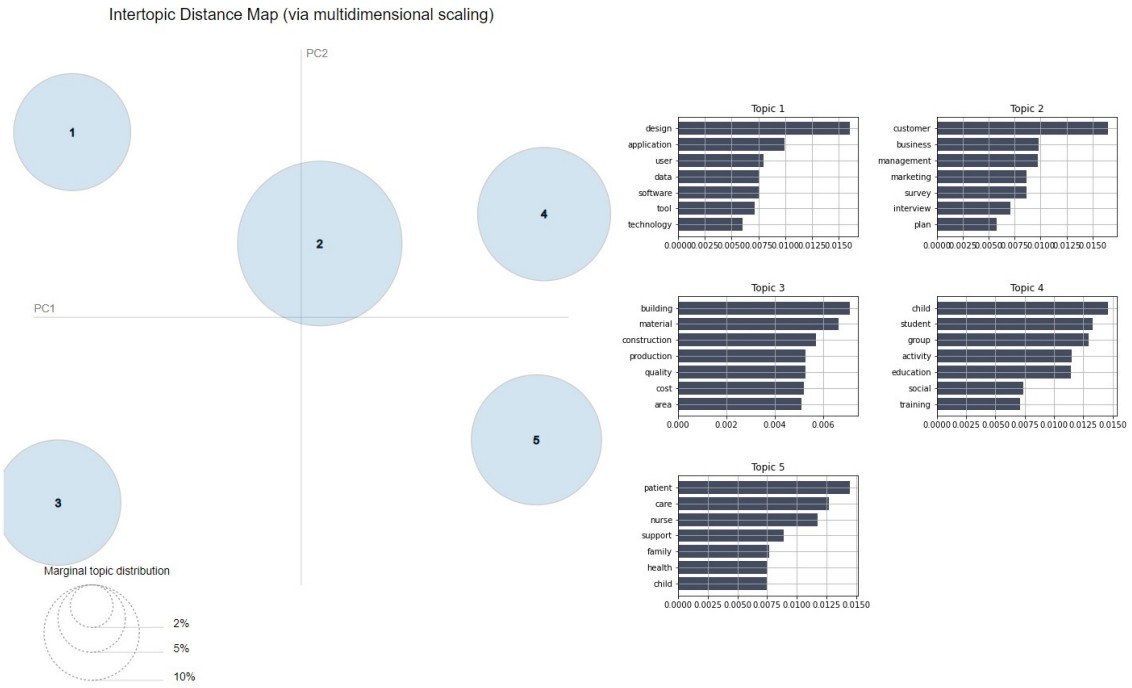


Figure 20: DTM model with 5 topics (last time frame).

Comparing the 5-topic model (figure 20) and the 6-topic (figure 21) model draws two major distinctions. First is the division of topic 2 of the first model, which becomes topic 1 and 6 in the second one; in other words, from *business* in general to *marketing* and *business management*. Furthermore, the *information technology* topic and a part of *building construction* topic of the 5-topic model integrate into topic 2 of the 6-topic model. The evident is some prominent words of *building construction* topic in the former, such as *building* and *construction*, are now belong to topic 2 of the latter model. The remaining part of this topic forms a topic by itself, the fourth one. The model hints that six topics are not sufficient to express all possible hidden topics.

Figure 21: DTM model with 6 topics (last time frame).

The 7-topic model is not to be an option either. This model has topics 4 and 7, interpreted as *child development*, *education* correspondingly, split from topic 3 of the 6-topic model. Secondly, the word *customer*, which is the most prominent word for the *marketing* topic in figure 21, becomes the dominant one in the *business management* topic in figure 22. This major change appears to clarify the *marketing* topic in the 7-topic model. However, topic 2 is relatively opaque. And the *information technology-building construction* still remains the same. Thus, the 7-topic model is not as definite as the 5-topic model.

Over and above that, the transform between seven and eight topics is considerable. First of all, the *building construction* topic and *information technology* topic are now clearly

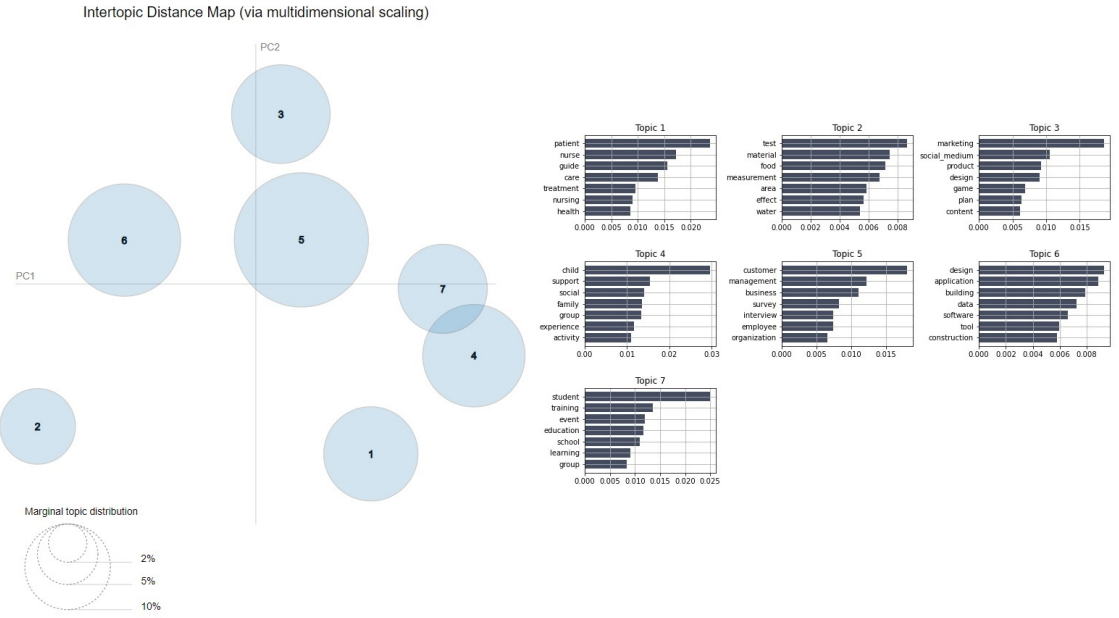


Figure 22: DTM model with 7 topics (last time frame).

defined in the 8-topic model (topic 4 and 5, respectively). Besides, the word *customer* and *marketing* are in the same category. Furthermore, surprisingly, topic 1 (*child development*) and topic 2 (*education*) are pretty distant in the plot (figure 23). On the other hand, topic 2 is closer to topic 8 (*customer/marketing*), despite the fact that the common words for both topics do not appear in the other one. Certainly, two topics have several words repeated, but not prominent ones, such as *medium*, *design*, *experience*; hence, the overlapping part. Topic 3 is somewhat difficult to describe. Topic 7 is even harder to define. Overall, this model is too vague to be used.

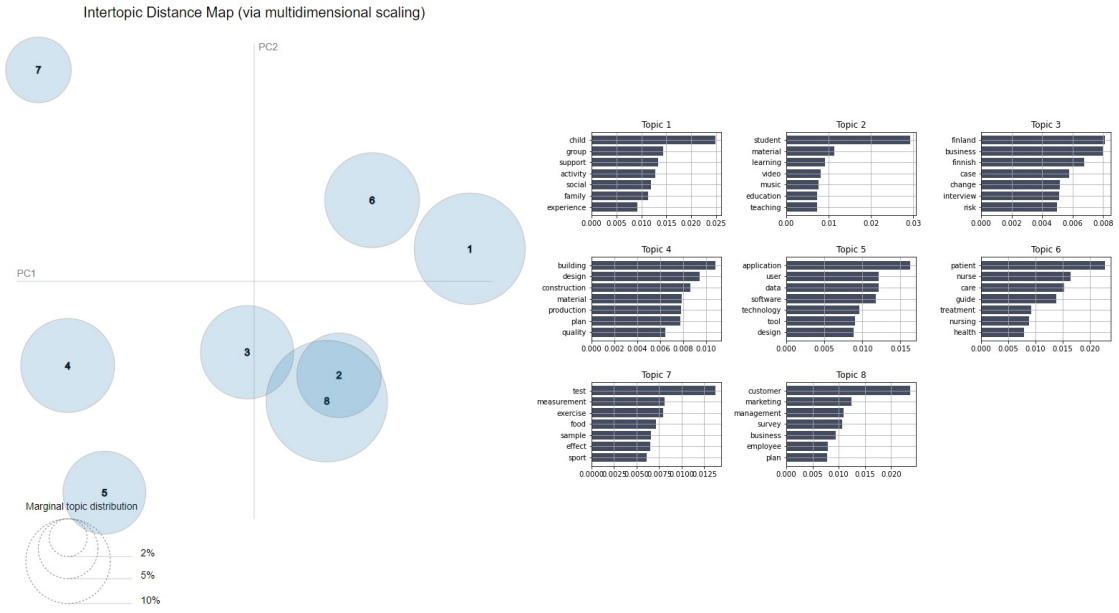


Figure 23: DTM model with 8 topics (last time frame).

Figure 24: DTM model with 9 topics (last time frame).

Last but not least, the 9-topic model share several similarities with the 8-topic one. Most of topics, specifically, *health care*, *customer/marketing*, *business management*, *information technology*, *building construction*, *education*, and *child development* remains relatively the same. Besides that, two new categories are formed: topic 3 and 9. The first one is hard to understand, while the second one can be read as *arts/entertainment*. This model is acceptable, but 5-topic model is more distinct.

Out of the inspected DTM models, the 5-topic one is the most promising, as all topics can be easily understood. This model is further analyzed of topics throughout the time. Figure 25, 26, 27, 28, and 29 below record the most frequency words for each topic in different time slides. As can be seen from those plots, those dominant words within the group hardly fluctuate, making it difficult to discover the topic's trend.

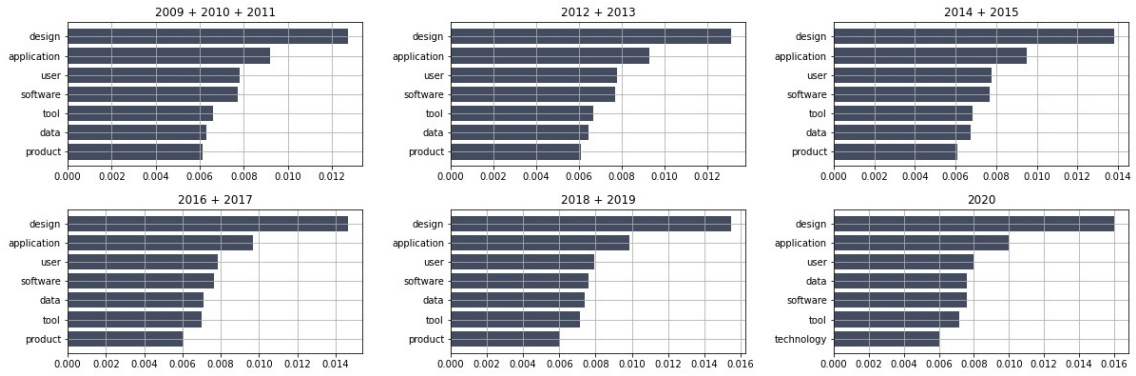


Figure 25: Information Technology - Topic 1 of the DTM 5-topic model through time.

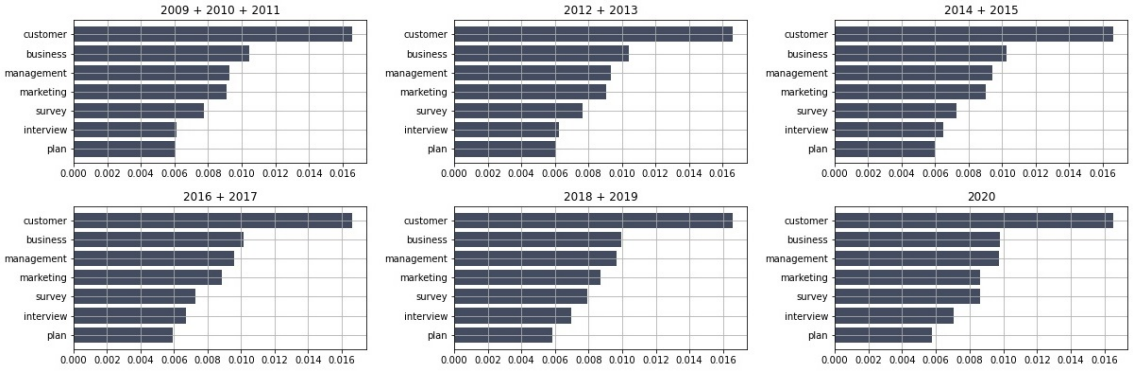


Figure 26: Business - Topic 2 of the DTM 5-topic model through time.

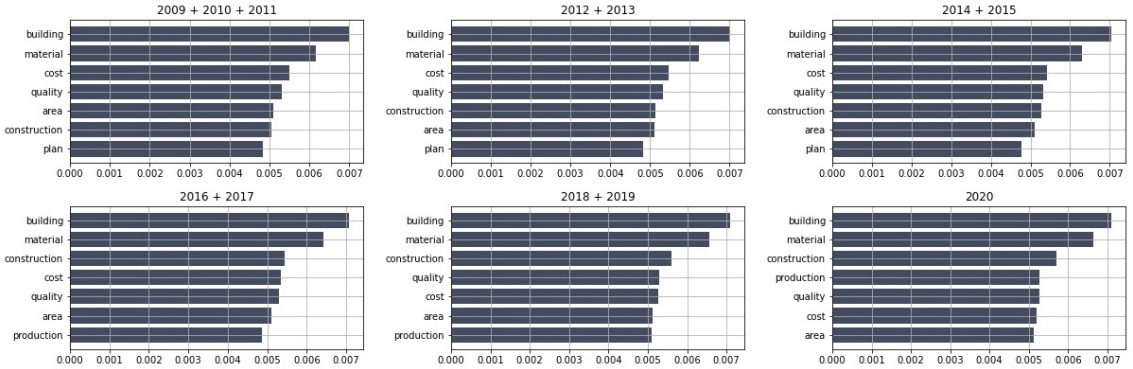


Figure 27: Building Construction - Topic 3 of the DTM 5-topic model through time.

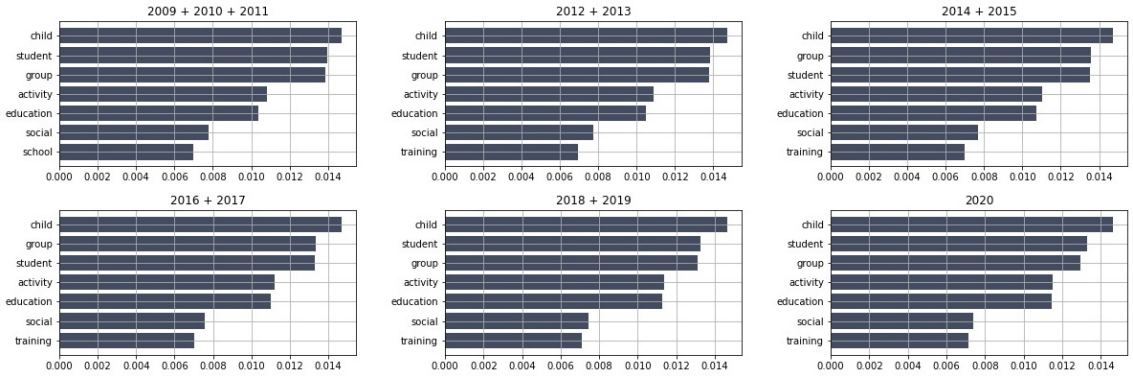


Figure 28: Education - Topic 4 of the DTM 5-topic model through time.

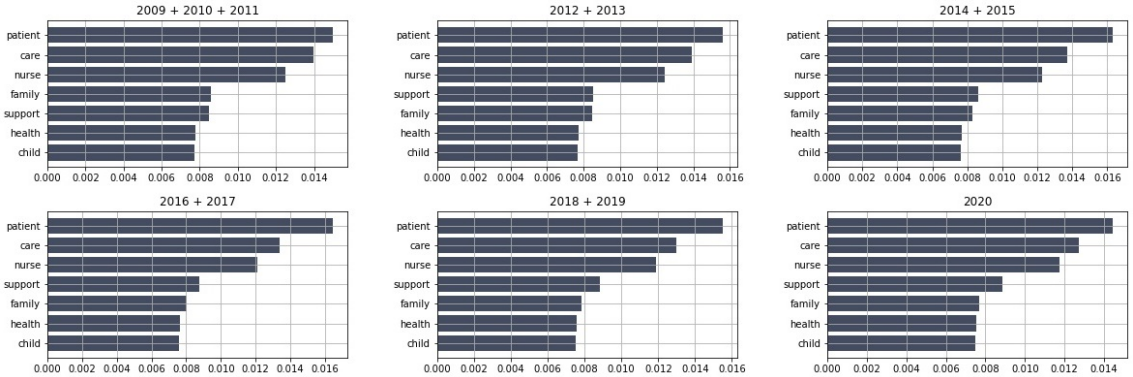


Figure 29: Health Care - Topic 5 of the DTM 5-topic model through time.

5.3 Comments

The bi-grams technique functions efficiently in supporting the models. Even though some bi-grams, such as *social_medium*, *health_care*, *early_childhood*, do not belong to the top common words and are not shown in the above graphs, they appear mildly frequent throughout most models, as recorded in the bi-grams word cloud (figure 11). Surprisingly, there are several intriguing bi-grams that appear fewer times but mainly belong to a topic: *artificial_intelligence*, *emergency_care*, *energy_efficiency*. Thus, those are extremely helpful for identifying the topic.

On the other hand, abbreviations rarely pop up during the whole process. Some that appear are *ERP - Enterprise Resource Planning*, *HR - Human Resources*, *ADHD - Attention Deficit Hyperactivity Disorder*, *VAT - Value-added Tax*, *AI - Artificial Intelligence*, *HVAC - Heating, Ventilation, and Air Conditioning*, etc. Similar to bi-grams, abbreviations are mostly related to one topic only, making them excellent indicators for forming and deciding topics themselves.

In the matter of topics, there are persistent classes whose common words stay relatively the same while the number of topics changes, such as, *health care*, *information technology*. On the contrary, topics like *business*, *education* usually gets diverged with higher number of topics. With the selected number of topics, there is not much overlapping area in the plots. Out of all, the LDA 8-topic and DTM 5-topic model are most promising.

6 Further Development and Conclusion

6.1 Further Development

To begin with, the text preprocessing part can be improved. For example, neutral words that seem not to belong to any topics, such as, *goal*, *people*, *time*, *change* can be filtered out. Besides, since bi-grams operate well, tri-grams can be considered. In one experiment, when only nouns are kept to build the dictionary, the results are not much different. Thus, it is not included in the report. The order of various methods in the text preprocessing can be changed as well; for example, do bi-grams before deleting stop words. Furthermore, worth noticing, some abbreviations duplicate with bi-grams, such as *AI - artificial_intelligence*, *HR - human_resource*, *VR - virtual_reality*. An additional dictionary can be added to recognize these replicates and uniform them, since there are a finite number of duplications. On the other hand, synonyms like *older_people* and *elderly_people* are plenty. The solution for this is to implement words embeddings, for example, the Top2Vec [31] or BERT [32] version.

Regarding the results, for the DTM models, the top-frequent words within a topic remain the same throughout different time frames. To highlight the change over time in one topic, TF-IDF can be applied for documents that mainly belong to that particular topic to delete common words. After the preprocessing, run the DTM again for that set of documents to further discover the trend of the topic in question.

Another challenge is to go out of the scope of analyzing given data to creatively build helpful applications. One proposal is to develop a tool for students to find their matching supervisors for topics they want to deliberate on.

To put this idea into practice, an extra attribute of supervisors' names of theses is required in the data. For a supervisor, all of the thesis abstracts that they once supervised are merged into one. After combining, the topic model is used to soft classify the passage, indicating which topics the supervisors are interested in and be responsible for. Go through the same process for the student's paragraph as well. Here, the similarity between a

supervisor and a student can be calculated using the Hellinger distance formula. To find the best supervisor for the given topics, compare the distance between a student and all available supervisors and choose the smallest one. This supervisor should be the best match for that student.

Here is a demo to briefly display how the tool might work with four artificial supervisors and an example student's short paragraph. The chosen model is the LDA 8-topic one (figure 16). Four supervisors are randomly assigned to theses, but ensuring supervisors devote to a particular topic. Four chosen dominant topics are *building construction*, *health care*, *education*, and *information technology*. Lastly, the paragraph is picked (from [42]) as the students' example as quoted below. After preprocessing and applying the model, the probability distribution of each topic for each subject is shown in figure 30.

"The control software is written in Python and runs on Raspberry Pi. It consists of 4 threads running simultaneously. The first and second ones are for reading weather data from the FMI Open Data service, the sun intensity data from pre-downloaded files from another FMI service called Ilmanet, and for transferring it to the simulator through REST API. Another thread is to apply AI algorithms to calculate the setpoint and send it to the simulator. The last thread is simply for the simulator to read and update the data. Each thread runs its own loop to do the task and sleeps until its next cycle. The length of the thread's cycle can be modified as well."

	1	2	3	4	5	6	7	8
supervisor_construction	0.604966	0.059281	0.081004	0.014471	0.014144	0.019094	0.039270	0.167771
supervisor_healthcare	0.015694	0.032550	0.066927	0.604568	0.124141	0.086014	0.019389	0.050717
supervisor_education	0.022546	0.078625	0.066176	0.079448	0.128524	0.527055	0.052795	0.044830
supervisor_it	0.116996	0.052235	0.061705	0.021204	NaN	0.034214	0.055436	0.648819
student_sample	NaN	NaN	0.033847	0.027934	NaN	0.105612	NaN	0.822165

Figure 30: Subjects - Topics probability distribution table.

From the table, considering a particular subject, the highest probability or value indicates that the subject is most related to that respective topic. Hence, the student paragraph is highly associated with topic 8, *information technology*. From here, it is evident that the fourth supervisor would be the best option. Still, in detail, the similarity between the student and supervisors can be accurately calculated using Hellinger distance; results are shown in figure 31. In short, in mathematics and statistics, this technique is employed to measure

the closeness of two probability distributions. The outcome of the formula ranges from 0 to 1. The lower the value, the more identical the two probability distributions are and vice versa. Thus, topics have 0 value of themselves and higher values of others. And for the student sample, the distance from it to *supervisor_it* is the lowest; therefore, they are a match statistically.

	supervisor_construction	supervisor_healthcare	supervisor_education	supervisor_it	student_sample
supervisor_construction	0.000000	0.699431	0.657728	0.428081	0.711343
supervisor_healthcare	0.699431	0.000000	0.475373	0.682130	0.719528
supervisor_education	0.657728	0.475373	0.000000	0.644965	0.687334
supervisor_it	0.428081	0.682130	0.644965	0.000000	0.359954
student_sample	0.711343	0.719528	0.687334	0.359954	0.000000

Figure 31: Hellinger distance between subjects.

6.2 Conclusion

The report presents the theoretical background of AI, ML, DL in NLP, topic modeling, as well as explains the NLP techniques in preprocessing texts. Essentially, the initial task of applying topic modeling in the given theses data is completed. Two algorithms LDA and DTM were employed. Their outcomes were examined and compared in detail. In consequence, the LDA 8-topic model performs best at separating topics compared to other models, and the DTM 5-topic model is worth to be analyzed further to discover the trend in different periods. Lastly, using the former model, an experience of simulating the supervisor-student matching tool was carried out. In the end, the demo proves the matching tool can be built, but with lots more effort. The tool would help to improve the student study experience. Finally, the code of the project mentioned in this thesis and the examined models can be found here [43]. Any comments are welcomed. Hopefully, the project will continue to discover all possibilities of applying AI, especially NLP into the higher education sector of 3UAS in the near future.

Bibliography

1 Holon IQ. Advanced technologies will embed into education delivery and learning processes.; 2021. Available from: https://www.holoniq.com/edtech/10-charts-that-explain-the-global-education-technology-market/ [cited June 11 2021].

2 Betsy Corcoran. Chegg Cuts $15 Million Check to Buy AI-Feedback Tool, WriteLab; 2018. Available from: https://www.edsurge.com/news/2018-05-16-chegg-cuts-15-million-check-to-buy-ai-feedback-tool-writelab [cited June 11 2021].

3 Business Wire. Gradescope by Turnitin Recognized by SIIA as Best Science and STEM Instructional Solution; 2019. Available from: https://www.businesswire.com/news/home/20190612005863/en/Gradescope-by-Turnitin-Recognized-by-SIIA-as-Best-Science-and-STEM-Instructional-Solution [cited June 11 2021].

4 Hietanen H, Sjölund AK. Theseus.fi: Open Access Publishing in the Finnish Universities of Applied Sciences. 2009. Available from: https://www.researchgate.net/publication/277751992_Theseusfi_Open_Access_Publishing_in_the_Finnish_Universities_of_Applied_Sciences [cited July 11 2021].

5 CSC. Docs CSC - Puhti and Mahti; 2021. Available from: https://docs.csc.fi/computing/overview/ [cited July 11 2021].

6 Vajjala S, Majumder B, Gupta A, Surana H. Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems. O'Reilly Media, Inc; 2020. Available from: https://learning.oreilly.com/library/view/practical-natural-language/9781492054047/ [cited June 18, 2021].

7 Raschka S, Mirjalili V. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt; 2019. Available from: https://learning.oreilly.com/library/view/python-machine-learning/9781789955750/ [cited June 28, 2021].

8 Christina V. Email Spam Filtering using Supervised Machine Learning Techniques; 2010. Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.1642&rep=rep1&type=pdf [cited June 28 2021].

9 Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. CoRR. 2017. Available from: http://arxiv.org/abs/1712.01815 [cited June 28 2021].

10 Mohammadhasani A, Mehrivash H, Lynch A, Shu Z. Reinforcement Learning Based Safe Decision Making for Highway Autonomous Driving; 2021.

Available from: https://arxiv.org/abs/2105.06517 [cited June 28 2021].

11 Zhou SK, Le HN, Luu K, Nguyen HV, Ayache N. Deep Reinforcement Learning in Medical Imaging: A Literature Review; 2021. Available from: https://arxiv.org/abs/2103.05115 [cited June 28 2021].

12 Hirsa A, Osterrieder J, Hadji-Misheva B, Posth JA. Deep Reinforcement Learning on A Multi-asset Environment for Trading; 2021. Available from: https://arxiv.org/abs/2106.08437 [cited June 28 2021].

13 Kaiser M, Roy RS, Weikum G. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs. CoRR. 2021. Available from: https://arxiv.org/abs/2105.04850 [cited June 28 2021].

14 Keneshloo Y, Ramakrishnan N, Reddy CK. Deep Transfer Reinforcement Learning for Text Summarization. CoRR. 2018. Available from: http://arxiv.org/abs/1810.06667 [cited June 28 2021].

15 Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR. 2018. Available from: http://arxiv.org/abs/1810.04805 [cited July 02 2021].

16 Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models are Unsupervised Multitask Learners; 2019. Available from: https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe [cited July 02 2021].

17 Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language Models are Few-Shot Learners. CoRR. 2020. Available from: https://arxiv.org/abs/2005.14165 [cited July 02 2021].

18 Patel AA, Arasanipalai AU. Applied Natural Language Processing in the Enterprise: Teaching Machines to Read, Write and Understand. O'Reilly; 2021. Available from: https://learning.oreilly.com/library/view/applied-natural-language/9781492062561/ [cited July 01, 2021].

19 Mazzeo V, Rapisarda A, Giuffrida G. Detection of Fake News on Covid-19 on Web Search Engines. CoRR. 2021. Available from: https://arxiv.org/abs/2103.11804 [cited June 20 2021].

20 Ayoub J, Yang XJ, Zhou F. Combat COVID-19 Infodemic Using Explainable Natural Language Processing Models. CoRR. 2021. Available from: https://arxiv.org/abs/2103.00747 [cited June 20 2021].

21 Safi Samghabadi N, López Monroy AP, Solorio T. Detecting Early Signs of Cyberbullying in Social Media. European Language Resources Association (ELRA); 2020. Available from: https://www.aclweb.org/anthology/2020.trac-1.23 [cited June 20 2021].

22 Eyigoz E, Mathur S, Santamaria M, Cecchi G, Naylor M. Linguistic Markers Predict Onset of Alzheimer's Disease. EClinicalMedicine; 2020. Available from: https://doi.org/10.1016/j.eclinm.2020.100583 [cited June 28 2021].

23 Albrecht J, Ramachandran S, Winkler C. Blueprints for Text Analytics Using Python: Machine learning-based Solutions for Common Real World (NLP) Applications. O'Reilly Media, Inc; 2020. Available from: https://learning.oreilly.com/library/view/blueprints-for-text/9781492074076/ [cited May 25, 2021].

24 Chopra R, Godbole AM, Sadvilkar N, Shah MB, Ghosh S, Gunning D. The Natural Language Processing Workshop: Confidently Design and Build Your Own NLP Projects with this Easy-to-understand Practical Guide. Packt Publishing Ltd.; 2020. Available from: https://learning.oreilly.com/library/view/the-natural-language/9781800208421/ [cited July 12, 2021].

25 Sutherland I, Sim Y, Lee S, Byun J, Kiatkawsin K. Topic Modeling of Online Accommodation Reviews via Latent Dirichlet Allocation. Sustainability. 2020. Available from: https://www.semanticscholar.org/paper/Topic-Modeling-of-Online-Accommodation-Reviews-via-Sutherland-Sim/01b7a25bea4d698e7feab403210006b83f49bf45 [cited July 15 2021].

26 Shah AM, Yan X, Tariq S, Ali M. What Patients Like or Dislike in Physicians: Analyzing Drivers of Patient Satisfaction and Dissatisfaction using a Digital Topic Modeling Approach. Information Processing Management. 2021. Available from: https://www.sciencedirect.com/science/article/pii/S030645732100025X [cited July 15 2021].

27 Schneider N, Fechner N, Landrum GA, Stiefl N. Chemical Topic Modeling: Exploring Molecular Data Sets Using a Common Text-Mining Approach. Journal of Chemical Information and Modeling. 2017. Available from: https://doi.org/10.1021/acs.jcim.7b00249 [cited July 15 2021].

28 Kim HJ, Yardımcı GG, Bonora G, Ramani V, Liu J, Qiu R, et al. Capturing Cell Type-specific Chromatin Compartment Patterns by Applying Topic Modeling to Single-cell Hi-C Data. PLOS Computational Biology. 2020. Available from: https://doi.org/10.1371/journal.pcbi.1008173 [cited July 15 2021].

29 Kim S, Park H, Lee J. Word2vec-based Latent Semantic Analysis (W2V-LSA) for Topic Modeling: A Study on Blockchain Technology Trend Analysis. Expert Systems with Applications. 2020. Available from: https://www.sciencedirect.com/science/article/pii/S0957417420302256 [cited July 15 2021].

30 Chung KY, Yoo H, Choe DE, Jung H. Blockchain Network Based Topic Mining Process for Cognitive Manufacturing. Wireless Personal Communications. 2019. Available from: https://www.semanticscholar.org/paper/Blockchain-Network-Based-Topic-Mining-Process-for-Chung-Yoo/3fb2efda6e17537cd20ac1b60d38a3604a78ed43 [cited July 15 2021].

31 Angelov D. Top2Vec: Distributed Representations of Topics. CoRR. 2020. Available from: https://arxiv.org/abs/2008.09470 [cited July 21 2021].

32 Grootendorst M. BERTopic: Leveraging BERT and c-TF-IDF to Create Easily Interpretable Topics. Zenodo; 2020. Available from:

https://doi.org/10.5281/zenodo.4381785 [cited July 15 2021].

33 Zhao H, Phung D, Huynh V, Jin Y, Du L, Buntine WL. Topic Modelling Meets Deep Neural Networks: A Survey. CoRR. 2021. Available from: https://arxiv.org/abs/2103.00498 [cited July 21 2021].

34 Blei DM, Ng AY, Jordan MI. Latent Dirichlet Allocation. J Mach Learn Res. 2003. Available from: https://dl.acm.org/doi/pdf/10.5555/944919.944937 [cited July 17 2021].

35 Wang X, Mccallum A. Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends; 2006. Available from: https://people.cs.umass.edu/~mccallum/papers/tot-kdd06s.pdf [cited July 17 2021].

36 Blei D, Griffiths T, Jordan M, Tenenbaum J. Hierarchical Topic Models and the Nested Chinese Restaurant Process. Advances in Neural Information Processing Systems. 2004. Available from: https://papers.nips.cc/paper/2003/file/7b41bfa5085806dfa24b8c9de0ce567f-Paper.pdf [cited July 17 2021].

37 Wang C, Paisley J, Blei D. Online Variational Inference for the Hierarchical Dirichlet Process. In: Gordon G, Dunson D, Dudík M, editors. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. PMLR; 2011. Available from: http://proceedings.mlr.press/v15/wang11a.html [cited July 20 2021].

38 Blei DM, Lafferty JD. Dynamic topic models. Proceedings of the 23rd international conference on Machine learning. 2006. Available from: https://www.semanticscholar.org/paper/Dynamic-topic-models-Blei-Lafferty/a1ca33025dc5c63486b1d6eb20c810008b513f8d [cited August 05 2021].

39 Teh YW, Jordan MI, Beal MJ, Blei DM. Hierarchical Dirichlet Processes. Journal of the American Statistical Association. 2006. Available from: https://doi.org/10.1198/016214506000000302 [cited July 20 2021].

40 Sabharwal N, Agrawal A. Hands-on Question Answering Systems with Bert: Applications in Neural Networks and Natural Language Processing. Apress Standard; 2021. Available from: https://learning.oreilly.com/library/view/hands-on-question-answering/9781484266649/ [cited July 24, 2021].

41 Röder M, Both A, Hinneburg A. Exploring the Space of Topic Coherence Measures. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. Association for Computing Machinery; 2015. Available from: https://doi.org/10.1145/2684822.2685324 [cited November 01 2021].

42 Vu M. Assemblin - Smart Building Control System; 2020. Available from: https://github.com/vnhm00/digisalama-assemblin/blob/main/Final%20Report.pdf [cited November 14 2021].

43 Vu M. Building Topic Modelling onTheses Abstracts Data; 2021. Available from: https://github.com/vnhm00/metro-topic-modeling/tree/main/Final%20-%20Thesis [cited November 16 2021].