Using Machine Learning Algorithms for Finding the Topics of COVID-19 Open Research Dataset Automatically

by

Donya Hamzeian

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Statistics

Waterloo, Ontario, Canada, 2021

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The COVID-19 Open Research Dataset (CORD-19) is a collection of over 400,000 of scholarly papers (as of January 11th, 2021) about COVID-19, SARS-CoV-2, and related coronaviruses curated by the Allen Institute for AI. Carrying out an exploratory literature review of these papers has become a time-sensitive and exhausting challenge during the pandemic. The topic modeling pipeline presented in this thesis helps researchers gain an overview of the topics addressed in the papers. The preprocessing framework identifies Unified Medical Language System (UMLS) entities by using MedLinker, which handles Word Sense Disambiguation (WSD) through a pre-trained Bidirectional Encoder Representations from Transformers (BERT) model. The topic model used in this research is a Variational Autoencoder implementing ProdLDA, which is an extension to the Latent Dirichlet Allocation (LDA) model. Applying the pipeline to the CORD-19 dataset achieved a topic coherence value of 0.7 and topic diversity measures of almost 100%.

Acknowledgements

First and foremost, I would like to express my sincerest appreciation to my supervisors, Professor Helen Chen and Professor Ali Ghodsi. Thank you Professor Chen for introducing me to the interesting area of NLP in Health sciences and supporting me in this path. Thank you Professor Ghodsi for teaching me attention to the details and thank you for your helpful insights.

I also wish to dearly thank my two Ph.D. mentors, Aref Jafari and Mojtaba Valipour who were always a big help whether it was technical issues, coding difficulties, preparing for a presentation, or even brainstorming for my thesis.

Finally, I would like to thank my husband, and my mother, who were all ears and always supportive when I had to encounter obstacles in this journey.

Dedication

This is dedicated to my love, Maziar. Thank you for being a wonderful teacher. I would not be here today without you.

Table of Contents

Abbreviations

AVITM Autoencoded Variational Inference for Topic Models vii, viii, 29–31, 33, 63

BERT Bidirectional Encoder Representations from Transformers 5, 10, 11, 47, 48, 55, 56, 58

BoW Bag of Words 8, 26

CTM Correlated Topic Model 17

CUI Concept Unique Identifiers 4, 9, 26, 31–34, 47, 57

LDA Latent Dirichlet Allocation 3–5, 12, 14–19, 28–31, 64

LSA Latent Semantic Analysis 13

NLP Natural Language Processing 5, 7, 8, 10, 35, 55

NMF Non-Negative Matrix Factorization 13

pLSI Probabilistic Latent Semantic Indexing 14, 15

ProdLDA LDA with a Product of experts 4, 5, 12, 30, 31, 47

SVD Singular Value Decomposition 13

TM Topic Modeling 1–5, 47

UMLS Unified Medical Language System 4, 5, 10, 11, 18, 25, 26, 31, 35, 45, 47, 48, 56–58

VAE Variational Autoencoder viii, 10, 18, 29, 30, 60, 62, 63

WSD Word Sense Disambiguation 47

Chapter 1

Introduction

1.1 Motivation

With the rising number of COVID-19 cases, more academic papers are published [1], making it almost impossible for a researcher or even a group of researchers to do a manual literature review on the COVID-19 topic. To make matters worse, the literature review requires prior knowledge in the area for which organizing a group of trained researchers for timely evidence synthesis and knowledge dissemination is not always feasible. COVID-19 is not the only field of research and body of literature that is rapidly growing. Therefore, it is desirable to seek tools, like Topic Modelling (TM) to assist in topic identification during scoping or systematic review process. TM not only reduces the length of time for researchers but also enables them to do the exploratory literature review of a large number of texts in a short period (Asmussen and Møller, 2019). In this thesis, we present a TM framework that helps researchers, especially biomedical researchers, to identify the topics in documents and group them accordingly. Although the procedure still requires helps from the experts, it significantly reduces their time and effort by considering semantics and contextual meanings of words in a text. TM is an unsupervised learning task for which, especially in domain-specific tasks, there is no benchmarking dataset for the evaluation. This is one of the most important challenges in this field.

[1] The CORD-19 dataset size has changed from 0.3 GB on 2020-03-13 to 6.4 GB on 2020-12-12. For more information, please visit here

1.2 Problem Definition

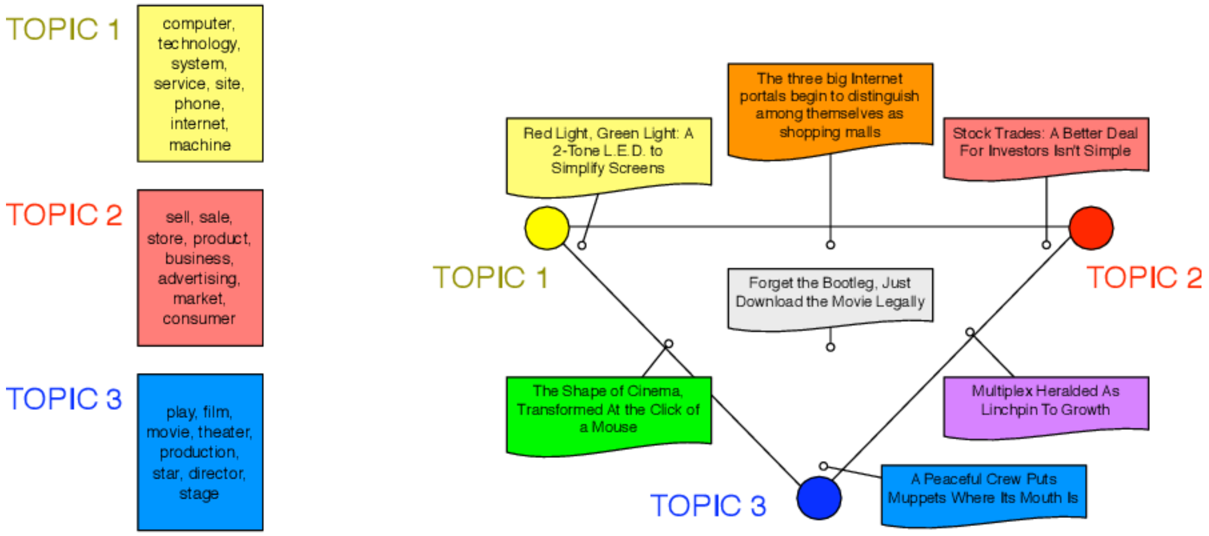
What is Topic Modeling?
Topic modeling is a statistical method for extracting *topics* from a collection of *documents*. *Topics* can be represented by *words*. For instance, one can assign the *topic* “technology” to a document by seeing the words, like “computer”, “phone”, “system”, “internet”, etc. A document may have more than one topic. For example, a document can also be assigned to the topic “Sale” because of the words like “sell”, “shopping”, etc. As such, a document can be described by a *distribution* over the topics and a topic can be described as a *distribution* over vocabularies (a pre-specified dictionary of terms). Figure 1.1 shows an example of the results from applying TM on some documents. On the left, each of the topics are shown with their top probability words. On the right, documents are shown on a simplex [2] formed by document-topic distributions. Each document can therefore be described as a combination of topics. Some documents are close to only one topic (the yellow document), some are between two topics (the orange one), and some are in the middle of all the topics (the gray one). Consequently, the yellow document can be attributed to TOPIC 1, the orange document can be attributed both to TOPIC 1 and TOPIC 2, and the gray one is related to all of these topics. Thus, it is not possible to assign a single topic to the gray document.
The goal of TM is to find these two following distributions.

1. θ: a matrix of document-topic (distributions) of size $M \times K$ (where M is the number of documents and K is the pre-defined value representing the number of topics)

2. Φ: a matrix of topic-word (distributions) of size $K \times N$ where N is the size of the *vocabulary* set.

When these two matrices are obtained using any of the TM models, an expert must use his or her domain knowledge to *name* each of the topics. Although TM can eventually be used to identify similar documents, it is not a clustering method in nature, i.e. clusters may not be considered as topics. First, in clustering, each document will be assigned to at most one cluster, whereas, in topic modeling, a document can have multiple topics. Second, topic modeling tries to cluster the documents and the words at the same time. Thus, it can be considered as a bi-clustering technique.

[2] See Chapter 2 for the definition of simplex.


TOPIC 1
computer, technology, system, service, site, phone, internet, machine
TOPIC 2
sell, sale, store, product, business, advertising, market, consumer
TOPIC 3
play, film, movie, theater, production, star, director, stage
The three big Internet portals begin to distinguish among themselves as shopping malls
Red Light, Green Light: A 2-Tone L.E.D. to Simplify Screens
Stock Trades: A Better Deal For Investors Isn't Simple
TOPIC 1
TOPIC 2
Forget the Bootleg, Just Download the Movie Legally
The Shape of Cinema, Transformed At the Click of a Mouse
Multiplex Heralded As Linchpin To Growth
TOPIC 3
A Peaceful Crew Puts Muppets Where Its Mouth Is

(a) Topics

(b) Document Assignments to Topics

Figure 1.1: (a) Each topic is represented by its top probability words. (b) Each document relies on the simplex (triangle) of topics which means it is represented by a probability distribution over the topics. [Source: Chang et al. (2009)]

1.3 Applications

Natural Language Processing (NLP) is not the only area in which topic modeling was used. TM, to be more specific, Latent Dirichlet Allocation (LDA) [3], has been used in other domains. According to Blei (2012), these domains include “survey data, user preferences, audio and music, computer code, network logs, and social networks.”
Another area that has benefited from topic modeling is the Recommender Systems. As an example, Wang and Blei (2011) built a framework which can be used in recommending a similar article to the researcher following a TM. After applying a TM algorithm, the features for documents are obtained, e.g. “biology”, “statistics”, etc. Also, each of the other documents is presented by a vector describing the distribution of the *features* in that document. Finally, the most similar documents can be found by calculating a similarity measure, like cosine similarity, on the document vectors. Similarly, since the model presented in this thesis mainly focuses on the biomedical texts, it could therefore be used by literature databases and search engines, such as PubMed, for recommending the most relevant articles to the user.
In computer vision, TM was used to classify images. Fei-Fei and Perona (2005) connected im-

[3] We will explain LDA in more details in Sections 3.4 and 5.2.2

ages and captions, Blei and Jordan (2003) built image hierarchies, and Bart et al. (2011), Li et al. (2010), and Sivic et al. (2008) used TM for other applications. One application is in Image Analysis where the idea is that each image is a combination of visual patterns and these sets of patterns are fixed among the collection of images (Blei, 2012). TM is useful to find how each individual image can be described using the combination of patterns and how each pattern can be characterized.

1.4 Thesis Contributions

In this thesis, we applied topic modeling on a (subset of a recent) collection of papers from COVID-19 Open Research Dataset, or CORD-19. Using a carefully designed preprocessing framework, with the Variational Autoencoder implementing ProdLDA (Srivastava and Sutton, 2017), we managed to achieve a high topic coherence value (C_v) of 0.7.

We noticed an imperfection in LDA/ ProdLDA that the model treats synonyms differently and does not necessarily assign the same probability to them. This becomes more problematic when the model wants to find the topic probability distribution for a previously *unseen* document. To illustrate, a new document may contain a word that is not present in the training data; however, its synonym was present in the training data. Then, the model cannot use that word for making inference about the topic distribution(of the new document. This problem calls for an enormous training corpus or a model that takes care of the synonyms. The former is more challenging when it comes to domain-specific topic modeling tasks. In this thesis, we leveraged the medical metathesaurus in the Unified Medical Language System (UMLS) (Bodenreider, 2004) to find medical entities and we replaced them with Concept Unique Identifiers (CUI). Since all the synonyms of a medical entity are replaced by one CUI, the model can treat them equally. When it comes to a new document, the mentioned problem is solved because the new word is also replaced by the same CUI. Another advantage of replacing the medical concepts with CUIs is to limit our topic model to using uni-grams [4]. Medical texts contain lots of n-grams, which is challenging because incorporating even bi-grams (n=2) leads to the *vocabulary* size grow exponentially, and yet not all the bi-grams are meaningful. Using CUIs instead of n-grams keeps the *vocabulary* size at a minimal number while capturing only the meaningful n-grams.

The reason why we only *replaced* the medical entities with CUIs and did not remove the non-entity words– similar to what Otmakhova et al. (2020) did– is that there are some words that UMLS does not capture, even though they are important in this domain. To name a few, “out-

[4] An n-gram means a sequence of n words. For example, ‘black’ is a 1-gram (uni-gram) and ‘black hole’ is a 2-gram (bi-gram) and so on.

break", "spread", "threat", and so on can relate to the topic "global pandemic", and determining this topic is very important for researchers.

Finding the entities in a text and linking them to the most appropriate UMLS entity is not a trivial task. MedLinker (Loureiro and Jorge, 2020) is a deep learning model that has shown a significant improvement in the task of concept (biomedical entity) linking. It uses BERT (Devlin et al., 2019) language model with Approximate Dictionary Matching for Mention Recognition and Entity Linking.

1.5 Thesis Layout

In Chapter 2 we briefly explain some mathematical and NLP terms that we use in this thesis and provide a short introduction about the UMLS and the deep learning models that we mentioned in this thesis such as BERT and MedLinker. In Chapter 3, we present the relevant works in topic modeling. We will then give an introduction to the most common TM algorithm, LDA, and its extensions. In Chapter 4, we present the common measures used for evaluating topic models which were also used in this work. In Chapter 5, we explain the preprocessing steps, and the variation-based LDA and ProdLDA networks that we used for training. In Chapter 6, we explain the dataset – CORD-19– that we applied the proposed framework to and show the results and discuss the calculated scores and the derived topics.

Chapter 2

Technical Background

In this chapter, we briefly explain some of the technical terms and models that we are going to use throughout this thesis. More detailed explanation is provided in the appendix.

2.1 Mathematical Terms

In this section, we briefly explain the mathematical terms that we frequently use throughout this thesis.

Simplex

In geometry, a **simplex** is a generalization of the notion of a triangle to other dimensions. The k-simplex (or the k-dimensional simplex) is the convex hull of its k +1 vertices $u_0, \dots, u_k$. That is the set of points satisfying

$$C = \left\{ \theta_0 u_0 + \cdots + \theta_k u_k \;\middle|\; \sum_{i=0}^{k} \theta_i = 1 \text{ and } \theta_i \geq 0 \text{ for } i = 0, \dots, k \right\}.$$

The **standard simplex** is the simplex whose vertices are the k standard unit vectors and the origin, which means

$$\{x \in \mathbb{R}^k : x_0 + \cdots + x_{k-1} = 1, x_i \geq 0 \text{ for } i = 0, \dots, k-1\}.$$

Softmax

The standard softmax function $\sigma : \mathbb{R}^K \to \mathbb{R}^K$ is defined by the formula

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K.$$

Dirichlet Distribution

Dirichlet distribution is a conjugate prior to the multinomial distribution and its probability density function is

$$Dir(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1},$$

where $B(\alpha) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)}$ and $\{\theta_k\}_{k=1}^{k=K}$ belongs to the standard $K-1$ simplex i.e. $\sum_k \theta_k = 1$ and $\theta_k \in [0,1]$.

2.2 Natural Language Processing Terms

In this section, we provide a short explanation of the NLP terms that we use in this thesis.

Corpus

In linguistics and Natural Language Processing, **corpus** refers to a collection of texts.

Lemmatization

According to *Collins English Dictionary*, "**lemmatization** in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form." As an example, the words "better", "went", and "bags" have "good", "go", and "bag" as their lemma respectively.

Document-Term Matrix

A **Document-Term Matrix** is a matrix that describes the frequency of terms that occur in a collection of documents.

Bag of Words (BoW)

The most common way to build the document-word matrix is to use the **Bag of Words** (BoW). In this representation, the collection of all vocabularies present in the corpus is called the *Dictionary* or the *vocabulary* and is of size N. Each document is then represented by a vector of size N that shows the frequency of each word (in the dictionary) in the document.

Term Frequency–Inverse Document Frequency (TF-IDF)

Since the raw counts cannot specify the significance of each word in the document, the **TF-IDF** (or Term Frequency-Inverse Document Frequency) score is often used instead of BoW for creating the document-term matrix. TF-IDF assigns a weight for term j in document i given as

$$w_{i,j} = tf_{i,j} \times \log \frac{M}{idf_j}.$$

Where $tf_{i,j}$ is the number of occurrences of term j in document i, M is the number of documents, and idf_j is the number of documents containing word j in the corpus.

Stop words

Stop words (stopwords) are the most commonly used words in any language that their removal does not change the meaning of a sentence, such as ‘the’, ‘a’, ‘he’, ‘on’, etc.

Word Sense Disambiguation (WSD)

Word Sense Disambiguation, or **WSD**, is a problem in NLP concerned with identifying which **sense** (meaning) of a word was used in the sentence. In other words, it is a task to resolve the ambiguity caused by similar words having different meanings in different contexts. For example, consider the two following examples:

1. I want to take a loan from a *bank*.
2. Rainfall caused Rhine river to overflow its *bank*.

The word *bank* in the first sentence refers to a ‘financial institution’, while in the second sentence refers to ‘river bank’.

2.3 Unified Medical Language System (UMLS)

The Unified Medical Language System, or **UMLS**, is a repository of biomedical vocabularies. The UMLS contains more than 2 million names from more than 60 biomedical vocabulary families for some 900,000 terms, as well as 12 million relationships between these concepts (Bodenreider, 2004). UMLS includes tools for extracting UMLS concepts from texts called MetaMap. The NCBI taxonomy, Gene Ontology, Medical Topic Headings (MeSH), OMIM and the Interactive Anatomist Symbolic Information Base are the source vocabularies incorporated into the UMLS Metathesaurus. Each concept (meaning) in the UMLS Metathesaurus is identified by a Concept Unique Identifier, or **CUI**. According to the *National Library of Medicine*, “A meaning can have many different names (aliases). A key goal of Metathesaurus construction is to understand the intended meaning of each name in each source vocabulary and to relate all the names from all of the source vocabularies that are synonyms[1].” CUI contains the letter C followed by seven numbers. For instance, the concept ‘Headache’ is identified by CUI = C0018681.

2.4 Neural Network Models

In this section, we provide a short introduction to the most important deep learning models that we mentioned or used in this thesis.

Autoencoder

An **Autoencoder** is an artificial neural network used to learn the efficient latent representation of the data– usually for the dimension reduction– in an unsupervised manner. The architecture of a typical autoencoder is composed of an *encoder* and a *decoder*. The encoder converts the original input into a latent representation. Then, the decoder reconstructs the data from the

[1] https://www.nlm.nih.gov/research/umls/new$_{u}sers/online_{l}earning/Meta_{0}05.html$

latent representation. The architecture of a simple autoencoder is depicted in Figure A.4. In Appendix A.3, more details are provided about autoencoders and why they cannot be used for data generation.

Variational Autoencoder (VAE)

Kingma and Welling (2014) presented Variational Autoencoder (VAE) that– unlike the vanilla autoencder– can be used for generating new data points. In this network, the inputs are mapped to a *distribution* rather than a *fixed* vector. In Figure 2.1, the VAE that encodes the inputs to the normal distribution is depicted. Since the normal distribution is characterized by its mean and variance, the encoder maps the input into mean, μ, and variance, σ, vectors. Then, a random sample $\mathbf{z}$ is drawn from this distribution using an auxiliary random variable sampled from the standard Gaussian distribution, ε, combined with the deterministic variables μ and σ.
Training the VAE (backpropagation [2]) involves taking gradients with respect to the sampled vector $\mathbf{z}$. However, taking derivative with respect to a stochastic variable is not possible. Reparameterization Trick (RT) is introduced to resolve this issue (Kingma and Welling, 2014). For further details about the VAE and the variational lower bound– or the Evidence Lower Bound (ELBO)– which is the loss function in this network and the Reparameterization Trick, please see Appendix A.4.

Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (Devlin et al., 2019), or BERT, is a deep learning model that generated state-of-the-art results in many of the NLP tasks. BERT, similar to Word2Vec (Mikolov et al., 2013), provides embeddings for words i.e. the words are mapped to vectors of real numbers. Unlike Word2Vec that generates a unique embedding for words in different contexts, BERT incorporates the contexts of the words into the embedding. To make things clear, consider the word *bank* in the above sentences. Since *bank* has different meanings in these two sentence, we expect two different embeddings for this word. However, Word2Vec does not consider the context words and generates a unique embedding for the word *bank* in the two sentences. BERT, on the other hand, generates different word embeddings based on the context words. As a result, BERT was used to disambiguate word senses (**?**). In this thesis, BERT is used in MedLinker– discussed in the next part. MedLinker uses BERT to find the UMLS concepts in a text and disambiguate them.

[2] Backpropagation is an algorithm for supervised learning of neural networks using gradient descent.

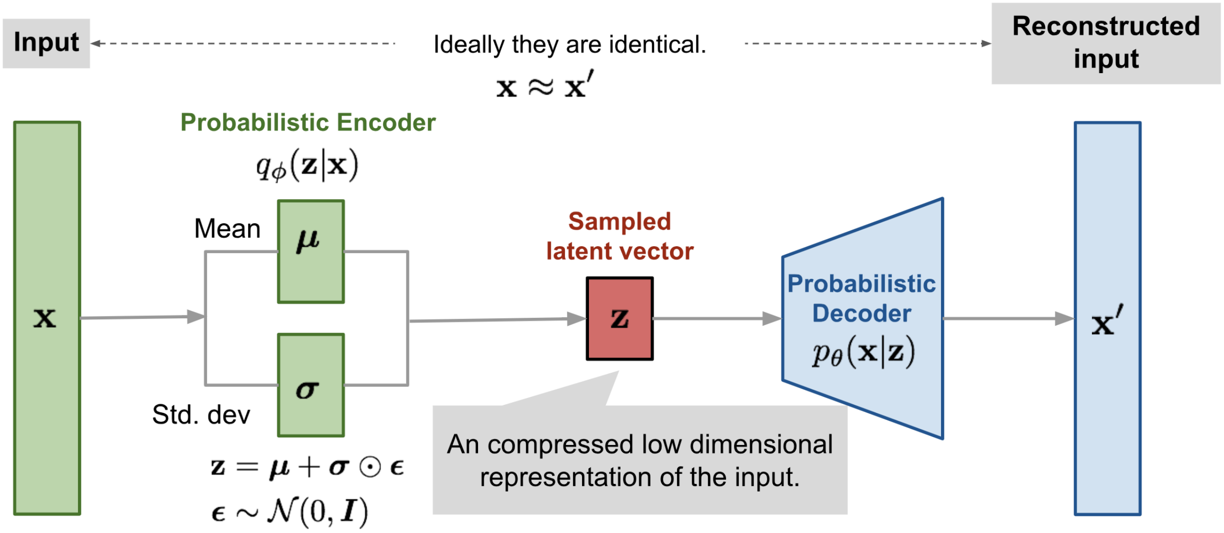

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
Std. dev
$\boldsymbol{\sigma}$
$\mathbf{x}$
Sampled latent vector
$\mathbf{z}$
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
An compressed low dimensional representation of the input.

Figure 2.1: Architecture of a variational autoencoder
Source: https://lilianweng.github.io

MedLinker

Loureiro and Jorge (2020) introduced MedLinker that handles medical entity linking with neural representations and dictionary matching. MedLinker uses BERT model both for identifying biomedical entities from texts and linking them to the most relevant concept in the UMLS metatheasurus. Because of using BERT as a contextual language model, MedLinker considers contextual similarity as well as string level similarity in linking the entities. An example of the output of MedLinker applied to a sample text and further details about its architecture are provided in A.2

Chapter 3

Relevant Works

All topic models have the same assumptions:

- Each document contains multiple topics
- Each topic can be represented by a collection of words

That is to say, the idea of topic modeling is that each document can be described by some latent variables– topics– and the goal is to find these hidden variables.
The topic models that will be explained in the rest of this section can be classified into two categories: **Non-Probabilistic** and **Probabilistic**. The first category that is based on matrix algebra includes Latent Semantic Analysis (LSA) and Non-Negative Matrix Factorization (NMF). The most important model in the field of topic modeling, LDA, relies in the second category along with its ancestor model, Probabilistic Latent Semantic Indexing (pLSI) (T.Hofmann, 1999), and the extensions to the LDA, including Correlated Topic modeling (CTM) (Blei and Lafferty, 2007), ProdLDA (Srivastava and Sutton, 2017), hierarchical LDA (Griffiths, 2004), Pachinko Allocation Topic Model (Li and McCallum, 2006), and the deep learning-based models.

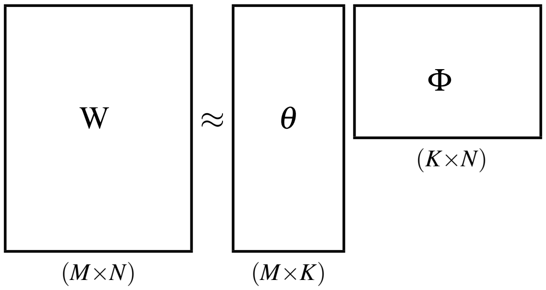
3.1 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA), sometimes called Latent Semantic Indexing (LSI) is an algebraic method based on truncated Singular Value Decomposition (SVD). This is one of the first methods in the field of topic modeling presented by Deerwester et al. (1990). Given the

document-word matrix (W), the main idea of LSA, is to decompose it into document-topic (θ) and topic-word matrix (Φ).
Given the document-word matrix, W, truncated SVD is applied to it to find the latent topics. SVD factorizes any matrix into 3 matrices $A = U\Sigma V^T$ where Σ is a diagonal matrix of the singular values of A. Truncated **SVD** approximates A by taking the k largest singular values and keeping the corresponding columns of U and V, where k is the pre-defined number of topics. Thus, the goal of LSA is to find the following:

$$A \approx U_K \Sigma_K V_K^T$$

$U_k \in \mathbb{R}^{M \times K}$ is the document-topic matrix, θ and $V_K^T \in \mathbb{R}^{K \times N}$ is the topic-word matrix, Φ.



The topic-word matrix, Φ, may have negative elements which make the topic embeddings hard to interpret. The inefficiency and lack of interpretability in LSA, made the path to other models in this category, such as the methods based on Nonnegative Matrix Factorization (NMF) (Lee and Seung., 1999).

3.2 Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF) was proposed by Lee and Seung. (1999) as a tool for data mining. Like SVD, NMF is a dimension reduction technique. However, it resolves the issue of negative elements in the topic-word matrix by imposing the non-negativity constraint. In text mining, NMF is used to decompose the nonnegative document-word matrix W into two low-rank nonnegative matrices θ and Φ such that $W = \theta\Phi + C$ where $\theta \in \mathbb{R}^{M \times K}$, $\Phi \in \mathbb{R}^{K \times N}$, $K \leq min(M,N)$ is a pre-defined parameter, and C is the noise matrix. NMF has been used in many applications including segmentation, dimension reduction, pattern recognition, image processing, language modeling, and so forth. In Årup Nielsen et al. (2005), they applied NMF on abstracts from PubMed. Biggs et al. (2008) proposed an algorithm for computing NMF called rank-one downdate (R1D), which is motivated by SVD. According to Biggs et al. (2008), "R1D

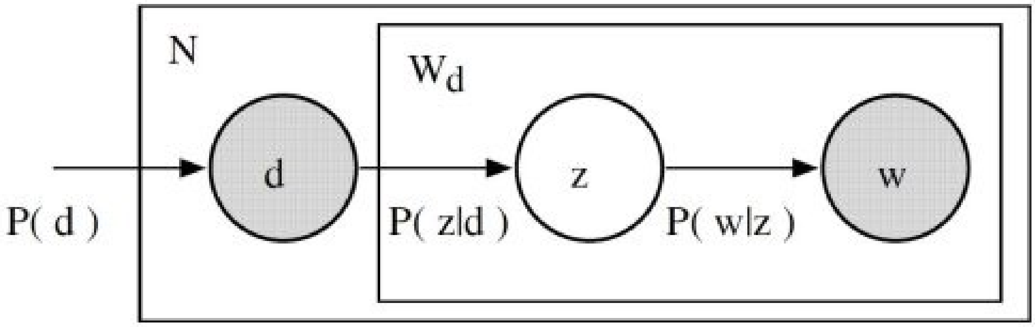


Figure 3.1: Plate notation of pLSI model

is based on the simple observation that the leading singular vectors of a nonnegative matrix are nonnegative."

3.3 Probabilistic Latent Semantic Indexing (pLSI)

Probabilistic Latent Semantic Indexing (pLSI), or pLSA, was first presented by T.Hofmann (1999), and it uses the probabilistic instead of algebraic methods. The main idea is to provide a generative process that can generate the observed document-word matrix. Using the two basic assumptions of topic modelling, pLSI provides the following generative process.

- For document d, topic z appears in the document with probability $P(z|d)$.
- For topic z, word w appears in the document with probability $P(w|z)$.

The plate notation of this model is depicted in Figure 3.1. The result of the above process is $P(d,w)$ which is the probability of each word w in document d that corresponds to the entries in the document-word matrix. The joint probability of $P(d,w)$ is therefore

$$p(w,d) = p(d)\Sigma_z p(z|d)p(w|z). \tag{3.1}$$

Where p(d) can be estimated from the corpus, but $p(z|d)$ and $p(w|z)$ are obtained using the E-M algorithm.
Because of the following problems, pLSI is no longer being used; however, it has laid the foundations for LDA.

- There is no parameters to model p(d), so pLSI does not assign probabilities to a new document; making it impossible to do the inference for a new document.
- The number of parameters, p(d), grows linearly with the number of documents; making pLSI inefficient and prone to overfitting.

3.4 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is based on pLSI. However, as it assumes that the topic-document and topic-word distribution matrices follow Dirichlet priors, it solves the problems of pLSI, described in the previous section. Figure 3.2 shows the plate notation of the generative process in LDA. $w_{m,n}$ are the words that are the only variables being observed, and the other variables are hidden.
To keep the consistency in the notations— in the following algorithm— M, N, K represent the number of documents (in the corpus), words (in the dictionary), and the topics respectively. In addition to K, α and β are also the hyperparameters of the model. They control the sparsity of topics per document and words per topic respectively.

Algorithm 1: LDA as a generative model

```
for each topic $k \in [1, K]$ do
    Randomly choose a distribution over words: $\Phi_k \sim$ Dirichlet($\beta$)
for each document $w_m$ do
    Draw topic distribution $\theta_m \sim$ Dirichlet($\alpha$) ;
    for each word at position n, $w_{m,n}$ do
        Sample topic $z_{m,n} \sim$ Multinomial(1, $\theta_m$);
        Sample word $w_{m,n} \sim$ Multinomial(1, $\Phi_{z_{m,n}}$);
```

Since Φ_k and θ_m, are distributions and build simplexes, the Dirichlet distribution— as priors— is a natural choice. They identify Dirichlet priors on the per-document topic, and the per-topic word distributions respectively.

The goal of LDA is to find the document-topic, θ, and the topic-word matrix, Φ, which maximize the following joint probability distribution over the hidden and observable variable.

$$p(\Phi,\theta,z,w) = \prod_{k=1}^{K} p(\Phi_k|\beta) \prod_{m=1}^{M} p(\theta_m|\alpha) \prod_{n=1}^{N} p(z_{m,n}|\theta_m) p(w_{m,n}|z_{m,n},\Phi_k). \tag{3.2}$$

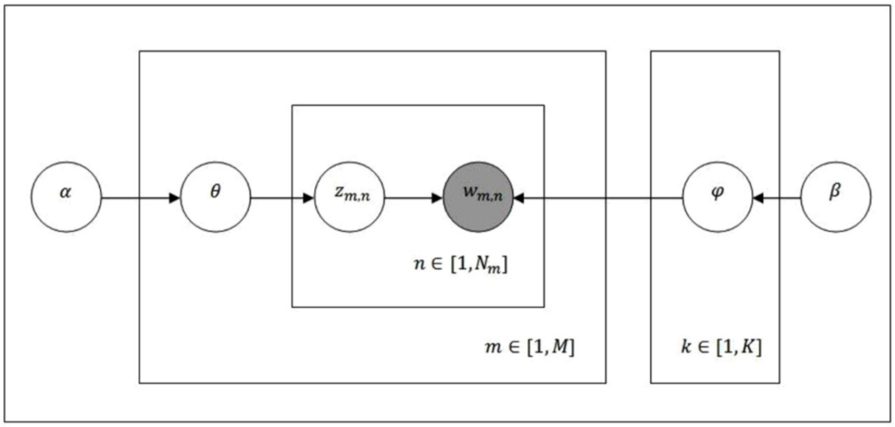


Figure 3.2: The plate notation of the LDA algorithm. The outer plate (on the left) shows documents and the inner plate (on the left) shows the topics and the words within a document. The right plane shows how topic distributions are generated. Figure from Ekinci and İlhan Omurca (2020)

3.4.1 Posterior Inference for LDA

Various inference techniques have been introduced for LDA, including E-M algorithm, (Collapsed) Gibbs sampling, and variation-based methods. In Section 5.2.2, we will explain that our model is using variational inference suggested in Srivastava and Sutton (2017) to solve the problem.

3.4.2 Extensions of LDA

LDA in its original form, proposed in Blei et al. (2003), makes some basic assumptions that the extension models target these assumptions and relax them.

1. Because of using Bag-of-Words (BoW), LDA assumes that the words in a document are exchangeable. Note that Equation 3.2 is invariant to the ordering of the words. Wallach (2006) relaxes this assumption by assuming the topics generate words conditional to the previous words.

2. LDA assumes documents are also exchangeable in the corpus. Again, Equation 3.2 is invariant to the order of documents. However, this is an unrealistic assumption since the topics, even the dictionary of the corpora, and the related distributions change in the long-run. As an example, the early documents in CORD-19 may be mostly about the origins

of the virus, whereas the recent documents may be mostly about the effectiveness of the vaccines. Even the vocabularies representing the study area must have changed; "Pfizer" must be just lately appeared in the context. An example of research in this area is the Dynamic Topic Models (Blei and Lafferty, 2006), where they used time series in modelling topics.

3. In LDA, the number of topics, K, is fixed and pre-defined. Hierarchical LDA (Griffiths, 2004) addresses this issue and sets different *number of topics* for different levels. The result is a hierarchy of topics where abstract topics are in the root and concrete topics appear near the leaves.

4. The Correlated Topic Model (CTM) (Blei and Lafferty, 2007) and Pachinko Allocation Topic (PAM) (Li and McCallum, 2006) are also based on LDA that assume correlation between topics. For example, if a document is talking about "Chemistry", it is more likely to be also talking about "Physics" rather than "Arts". CTM assumes the topics are related to each other through a probability distribution with mean and variance, and it tries to find the parameters of this distribution. PAM connects words and topics with an arbitrary Directed Acyclic Graph (DAG) and explores correlation in the *topics* through this DAG.

3.5 Incorporating Metadata

There has been a great amount of research that incorporated some additional information about the data– like author, title, geographic location, links, entities, etc– into topic modeling. The author-topic model (Rosen-Zvi et al., 2012) enables the inference both about the author and the document. Chang and Blei (2009) assumed "that the links between documents depend on the distance between their topic proportions", and developed a new topic model. Named entities are other forms of metadata that have been used in topic models. Newman et al. (2006) presented four extensions to the LDA model. In all these models, it is assumed that a document consists of words and entities. As such, topics include entities as well as words. Therefore, when a document is being generated, the entities are also chosen from a distribution that must be learned. The topic-document distribution must also be learned, similar to the LDA.

Using entities is especially helpful in domain-specific corpora– like in medical texts– because in these contexts, n-grams play a vital role in perceiving a topic. For example, "Single Nucleotide Polymorphism" is an entity in medical contexts that is identified by a unique identifier (ID); however, it may be hard, or even impossible to generate topics with having this tri-gram as its top words. Also, without incorporating entities, the topics may not be specific enough. For instance,

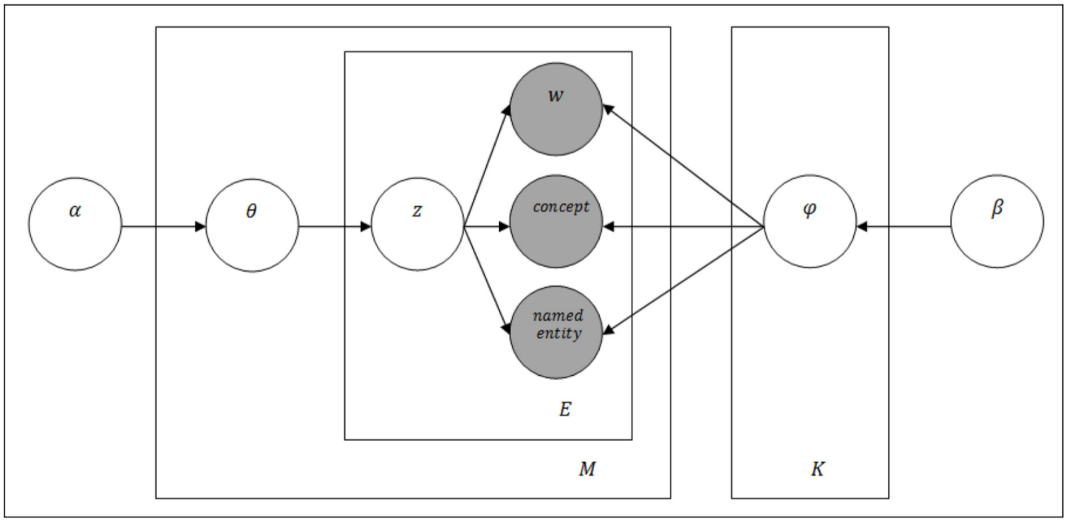


Figure 3.3: The plate notation of Concept-LDA where the bag of words were concatenated with the concepts and named entities (extracted from Babelfy). Ekinci and İlhan Omurca (2020)

the general word “single” would appear in the topics, instead of the specific entity “Single Nucleotide Polymorphism”. Otmakhova et al. (2020) applied LDA on the UMLS concepts (entities) extracted from the CORD-19 texts. Ekinci and İlhan Omurca (2020) enlarged the bag of words of the texts by adding UMLS concepts and named entities[1]. The model they used is a combination of Bag of Words (BoW) [2], Bag of Concepts (BOC) [3], and Named Entities. The plate notation of their model is depicted in Figure 3.3.

3.6 Using Deep Learning

In deep learning-based topic models, people have used neural networks either to implement the original form of LDA or to combine LDA with well-known models, like Word2Vec (Mikolov et al., 2013), reinforcement learning, and Generative Adversarial Networks (GAN) (Goodfellow et al., 2014). Miao et al. (2016) used Variational Autoencoder (VAE) (Kingma and Welling, 2014) to implement LDA with gaussian priors. Srivastava and Sutton (2017) also used VAE to implement LDA. However, they approximated the Dirichlet prior in LDA with a Logistic-Normal

[1] The (UMLS) entities are found using external tools.

[2] BOWs can be obtained by tokenizing the documents

[3] BOCs are composed of UMLS entities extracted from the document.

distribution. Gui et al. (2019) used Srivastava and Sutton (2017)'s model in a reinforcement learning framework with the topic coherence (See Section 4.3) as the reward. Moody (2016) built lda2vec that combines the idea of Word2Vec and LDA to simultaneously learn the word embeddings as well as topic representations and document representations. Wang and Zhou (2019) and Wang et al. (2020b) used GAN for modelling topics using Dirichlet prior.

Chapter 4

Topic Evaluation

Topic modeling is an unsupervised task and despite many of the clustering tasks that can be evaluated using gold labels, not every corpus has benchmarking datasets due to the time-consuming task of preparing them. Consequently, many *measures* have been introduced for topic evaluation. These measures either compute *topic diversity*, *perplexity*, or *coherence*. In the rest of this chapter, we are going to briefly explain some of the *measures* in each category.

4.1 Topic Diversity

The following measures, compute the diversity of the top words in the topics. The parameter τ which determines the number of the words with the highest score to be considered must be specified for calculating these scores. Note that K is the number of topics in the rest of this chapter.

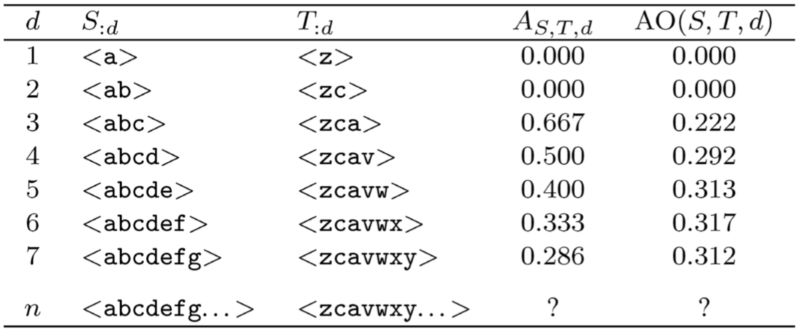
4.1.1 Naive Measure (TD)

In the most diverse case, the top words of all the topics must be unique, i.e. we must have $\tau \times K$ unique words. Therefore, a naive measure could be the number of the unique words in the union set of top τ words from all the topics divided by $\tau \times K$, formulated as

$$TD = \frac{\text{\# unique words}}{\tau \times K}. \tag{4.1}$$

4.1.2 Rank-Biased Overlap (RBO)

Webber et al. (2010) presented a measure for the similarity of two ranked sets called Rank-Biased Overlap (RBO). Given that the top τ words of each topic are in fact a ranked set, we can calculate the RBO of the word sets for each pair of the topics and take the average of the pairwise RBO scores. In comparing the ranked lists, like the sets of the top words, the main challenge is that an item may exist only in one of the sets. RBO tackles this challenge by extending the idea of the *set-based measure*, which does not care about the ranks. It compares the two lists at different depths in order to take the rank into consideration as well [1]. Figure 4.1 sketches the idea of using RBO for comparing ranked lists S and T by computing first their agreement $A(S,T,d)$ in different depths d and then their cumulative agreement (Average Overlap $AO(S,T,d)$) where $A(S,T,d) = \frac{|S_{:d} \cap T_{:d}|}{d}$ and $AO(S,T,\tau) = \frac{\sum_{d=1}^{\tau} A(S,T,d)}{\tau}$. Due to the unboundedness of AO for indefinite lists, they changed this measure to a bounded measure by using the geometric series. The detailed explanation of this score is out of the scope of this thesis and interested readers can see the original paper.



d	$S_{:d}$	$T_{:d}$	$A_{S,T,d}$	$\mathrm{AO}(S,T,d)$
1	<a>	<z>	0.000	0.000
2	<ab>	<zc>	0.000	0.000
3	<abc>	<zca>	0.667	0.222
4	<abcd>	<zcav>	0.500	0.292
5	<abcde>	<zcavw>	0.400	0.313
6	<abcdef>	<zcavwx>	0.333	0.317
7	<abcdefg>	<zcavwxy>	0.286	0.312
n	<abcdefg...>	<zcavwxy...>	?	?

Figure 4.1: Source: Webber et al. (2010)

4.2 Perplexity

Perplexity is one of the oldest measures in the topic modelling literature. This predictive measure, which is borrowed from the language model area, computes the log-likelihood of the held-out set of documents $\mathbf{w} = (w_1, \ldots, w_d)$. Perplexity is calculated based on the Φ matrix that shows the

[1] Set $A = \{a,b,c,d\}$ in depth=1 is $\{a\}$, in depth=2 is $\{a,b\}$, etc.

distribution of words in each of the topics. The topic-document matrix, θ, cannot be used for modelling the held-out set since they are unseen documents. The formula will therefore be [2]

$$\text{perplexity}(\mathbf{w}) = \exp\left\{-\frac{\sum_d \log(p(w_d|\Phi,\alpha))}{\text{count of tokens}}\right\}. \tag{4.2}$$

Perplexity has been rarely used since Chang et al. (2009) showed that perplexity is negatively correlated with human evaluation of the topics. For this reason, we did not calculate this measure.

4.3 Coherence

If a set of statements support each other, we call them coherent. Applying the same idea in topic modeling, we call a topic *coherent*, if the top words of it make sense together. An example of a coherent topic is {*game, sport, ball, team*}, whereas an incoherent topic could be {*game, sport, ball, penguin*}. A human or an expert in specific domains can distinguish the *intruder* word(s) [3]. However, it can be a challenging task for computers. The measures introduced for *coherence* are using a corpus, the WordNet (Fellbaum, 1998), or the word embeddings. We will only discuss the corpus-based scores and among them Coherence Value (**$\mathbf{C_V}$**), **NPMI**, **UCI**, and **UMASS** are the ones Newman et al. (2010) and Röder et al. (2015) showed they have the highest correlation with the human evaluation of the topics. Because of this, we only calculated these measures.

4.3.1 Term Co-occurrence

According to Röder et al. (2015), “The topic coherence measures take the set of top τ words of a topic and sum a confirmation measure over all word pairs.” Newman et al. (2010) showed that the coherence based on Pointwise Mutual Information (PMI) and Normalized Pointwise Mutual Information (NPMI) [4] have the highest correlation with human judgment in the topic evaluation, and later Röder et al. (2015) found the most correlated measures in this category by exploring the space of topic coherence measures. These measures are C_V, UMASS, UCI, and NPMI [5], where C_V yielded the highest correlation with the human judgment. In the rest of this section, we briefly explain how these scores are calculated [6]. We first need to explain PMI and NPMI

[2] Since $p(w_d|\Phi,\alpha)$ is intractable, the whole perplexity is also interactable.

[3] The intruder words are the words that do not belong to that topic. In this case, *penguin* is the intruder.

[4] The formula is in 4.4

[5] These measures are available in the Gensim package in python.

[6] For more details see Röder et al. (2015).

between two words.

The PMI between two words w_i and w_j is calculated by

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j) + \varepsilon}{P(w_i) \cdot P(w_j)}. \tag{4.3}$$

Probabilities are estimated based on word co-occurrence counts. Usually, these counts are estimated from documents that are constructed by a sliding window which moves over the Wikipedia, or another external corpus (The corpus is called *external* since it was not used for training the topic model). Aletras and Stevenson (2013) introduced topic coherence based on context vectors for every top word of a topic. Therefore, word w is represented using word co-occurrence counts determined by context windows that contain all words located ±5 tokens around the occurrences of the word w. The elements of these vectors are NPMI, where the j-th element of the context vector $\vec{v}_i$ is defined as below:

$$v_{ij} = NPMI(w_i, w_j)^\gamma = \left(\frac{\log \frac{P(w_i, w_j) + \varepsilon}{P(w_i) \cdot P(w_j)}}{-\log(P(w_i, w_j) + \varepsilon} \right)^\gamma. \tag{4.4}$$

where higher values of γ gives more weights to higher NPMI values. To find the similarity between the word pairs, different confirmation measures have been used, like cosine, Dice, or Jaccard.

1. **UCI**

   UCI was proposed by Newman et al. (2010). The UCI coherence for a topic based on its top τ words is calculated by:

$$C_{UCI} = \frac{2}{T \cdot (T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} PMI(w_i, w_j). \tag{4.5}$$

2. **UMASS**

   UMASS was proposed by Mimno et al. (2011). UMASS is very similar to UCI except that it uses the conditional probability between top word pairs. It is given by

$$C_{UMASS} = \frac{2}{T \cdot (T-1)} \sum_{i=2}^{T} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \varepsilon}{P(w_j)}. \tag{4.6}$$

3. **Coherence Value C_V**

By exploring the space of topic coherence measures, Röder et al. (2015) found that instead of defining the probability over word pairs, considering each top word of a topic, $W^{'}$, with the set of the context words of W' leads to a better correlation with the human evaluation. To make things clear, two words may not have high co-occurrence with each other, but frequently appear with the *context* words of each other. As an example, *Nike* and *Adidas* are two brands of athletic shoes that may not appear together, yet they both appear with context words *shoes* and *athletic*. Thereby, since indirect measures (context-based) may capture semantic supports that direct measures (non-contextual) miss, they introduced a novel way to segment the word subsets as below

$$S_{set}^{one} = \left\{(W^{'}, W^{*}) | W^{'} = \{w_i\}; w_i \in W; W^{*} = W\right\}, \tag{4.7}$$

where W^* is the set of context words [7] of W' . Then, the confirmation measure– M– is applied on a single pair $S_i = (W', W^*)$ and the final coherence value C_V of a topic is the arithmetic mean of the single scores on S_i's. The coherence measure of a single pair S_i is the cosine similarity between the two context vectors $\vec{u} = \vec{v}(W)$ and $\vec{w} = \vec{v}(W^*)$ of size $|W|$ (the size of the context words set). These vectors are defined as

$$\vec{v}_{m,\gamma}(W^*) = \left\{ \sum_{w_i \in W^*} m(w_i, w_j)^{\gamma} \right\}_{j=1,\ldots,|W|}. \tag{4.8}$$

Since W' only contains one word, i.e. $W' = \{w_i\}$, we have

$$\vec{v}_{m,\gamma}(W') = \left\{ m(w_i, w_j)^{\gamma} \right\}_{j=1,\ldots,|W|}, \tag{4.9}$$

where $\gamma = 1$ for C_V and the coherence measure M is NPMI:

$$m_{nlr}(w_i, w_j) = \left(\frac{\log \frac{P(w_i,w_j)+\varepsilon}{P(w_i)\cdot P(w_j)}}{-\log(P(w_i, w_j) + \varepsilon} \right). \tag{4.10}$$

Finally, the similarity between the two context vectors $\vec{u}$ and $\vec{w}$ are calculated as

$$s_{cos}(\vec{u}, \vec{w}) = \frac{\sum_{i=1}^{|W|} u_i \cdot w_i}{||\vec{u}_i||_2 \cdot ||\vec{w}_i||_2}. \tag{4.11}$$

[7] The best window size that Röder et al. (2015) found equals 110.

Chapter 5

Proposed Pipeline

The pipeline that we propose in this thesis consists of these two modules: **Preprocessing** and **Topic Model**. The former is explained in Section 5.1, and the latter in Section 5.2. These steps are done after splitting the given documents into train and test data.

5.1 Preprocessing

We designed a preprocessing framework that can be used for any medical paper. Our framework incorporated expert domain knowledge by using UMLS (Bodenreider, 2004). Using MedLinker (Loureiro and Jorge, 2020) [1], biomedical entities were identified and linked to the UMLS metathesaurus. The preprocessing steps are explained below.

1. The documents are all lower cased.
2. Since the topics of a document are almost irrelevant to the numbers, decimal numbers, ratios, and dates, we decided to remove them[2]. As a result, in this pipeline, they are removed using regular expressions [3]. The incorporated regular expression removes numbers, decimal numbers, and ratios like '7234', '12.5', '35,000', and '3:1'. However, the the digits that are inside the words are not removed, like '1' in *h1n1*, or '19' in 'COVID-19' because they affect the meaning of the words that embrace them.

[1] For more detailed explanation of the MedLinker pipeline please refer to Appendix A.2

[2] Nevertheless, one may argue that when a document is for example mostly about *statistics*, the probability of numbers in that document is higher. However, we decided not to consider these special cases.

[3] A regular expression is a sequence of characters used in programming languages to identify a search pattern.

3. The punctuation marks, and all non-alphanumeric symbols, except *dash*, are removed.
4. The stopwords are removed using the stopwords list available in the NLTK [4] package in Python. We added some extra stopwords to this list. [5]
5. Extra white spaces are removed.
6. All words are lemmatized [6] using the lemmatizer in Scispacy package in python [7]Neumann et al. (2019). The reason for lemmatization is that we do not want our model to be penalized if it produced *experiment* instead of *experimenting*. In other words, we do not care about the parts of speech when the model is reconstructing the BoW. We are only interested in their concepts and the meaning that they convey.
7. The tokenization [8] is done using Scispacy [9].
8. MedLinker is applied on the sentences and the spans in the text with the most relevant UMLS entity are obtained. Then, the tokens within the spans are replaced by the relevant CUI. For example, in Table 5.1, 'Inflammatory diseases' spans from token 0 to 1 and it was replaced by CUI= C0021368.
9. As a result, the *vocabulary* and the bag of words consist of both CUIs and words.

An example of the original vs. preprocessed document is shown in Table 5.1. Some changes are as follow. All words are lower cased. 'Inflammatory diseases' was replaced by C0021368. Stopwords 'of', 'the', 'are', and so on were removed. 'respiratory tract' was replaced by 'C0035237', punctuations except '-' were removed. 'elevated' was lemmatized.

[4] The Natural Language Toolkit (NLTK) is an open source Python library for Natural Language Processing.

[5] The following additional stopwords were added to the stopwords list. These are capitalized words that we identified them using regular expressions. PATIENTS, STATISTICAL, ANALYSIS, SETTING, REVIEWERS, QUESTIONS, PURPOSES, PURPOSE, EXPERIMENTAL, DESIGN, RELEVANCE, CLINICAL, ENHANCED, VERSION, REGISTRATION, MATERIALS, ELECTRONIC, MATERIAL, PRESENTATION, OBJECTIVE, METHODS, CONCLUSION, RESULTS, BACKGROUND, INTRODUCTION, DISCUSSION, SIGNIFICANCE, DESCRIPTION, ABSTRACT, METHODOLOGY, PRINCIPAL, FINDINGS, INFORMATION, SUPPLEMENTARY, SUMMARY

[6] **Lemmatization** is explained in Chapter 2.

[7] The ScispaCy is an open source Python package for processing biomedical, scientific or clinical texts.

[8] Tokenization is the process of splitting a large text into words.

[9] MedLinker handles the tokenization by using the tokenization module in Scispacy

original	Inflammatory diseases of the respiratory tract are commonly associated with elevated production of nitric oxide (NO•) and increased indices of NO• -dependent oxidative stress. Although NO• is known to have anti-microbial, anti-inflammatory and anti-oxidant properties, various lines of evidence support the contribution of NO• to lung injury in several disease models. On the basis of biochemical evidence, it is often presumed that such NO• -dependent oxidations are due to the formation of the oxidant peroxynitrite, although alternative mechanisms involving the phagocyte-derived heme proteins myeloperoxidase and eosinophil peroxidase might be operative during conditions of inflammation. Because of the overwhelming literature on NO• generation and activities in the respiratory tract, it would be beyond the scope of this commentary to review this area comprehensively. Instead, it focuses on recent evidence and concepts of the presumed contribution of NO• to inflammatory diseases of the lung.
processed	C0021368 C0035237 commonly associate elevate production C0028128 increase index C0030106 dependent oxidative stress although C0028128 know anti-microbial C0243095 anti-oxidant property various line evidence support contribution C0273115 several C0012634 model basis biochemical evidence often presume C0028128 dependent C0030011 due formation C0003402 C0136157 although alternative mechanism involve C2936482 C0033684 C0027021 C0059407 might operative condition inflammation overwhelm C0023866 C0028128 generation activity C0024109 would beyond scope C0584947 review area comprehensively instead focus recent evidence concept presume contribution C0028128 C0021368 lung

Table 5.1: An example of a preprocessed document

5.2 Topic Model

In this section, we describe the LDA-based model that we used for finding the topic-word (distribution) matrix, Φ and the topic-document (distribution) matrix, θ. As we discussed in Section 3.4, the goal of the LDA model is to optimize the joint probability distribution 3.2. Several inference models have been introduced for this aim, such as Gibbs Sampling, E-M algorithms, Variation-based methods, and so on. The model that we have used is based on variational Bayes and is called Autoencoded Variational Inference for Topic Models (AVITM), which was presented by Srivastava and Sutton (2017). In the rest of this section, we will briefly explain how inference is made in this model.

5.2.1 Variational Bayesian approach to LDA

In Section 3.4, we explained the extended model of LDA where the topic-word matrix, Φ, is also Dirichlet-distributed. However, the model that Srivastava and Sutton (2017) proposed and we have used is the simpler version of LDA. In this simple version, it is assumed that only the document-topic matrix, or θ, is Dirichlet-distributed and no distribution is assumed on Φ. However, a softmax function is applied so that each row of Φ denotes a probability distribution over the words of the *vocabulary*. In other words, the rows are constrained to be in a simplex. The modified algorithm is shown below and the plate notation is depicted in Figure 5.1.

Algorithm 2: LDA as a generative model (modified)

```
for each document $w_m$ do
    Draw topic distribution $\theta_m \sim$ Dirichlet($\alpha$) ;
    for each word at position n, $w_{m,n}$ do
        Sample topic $z_{m,n} \sim$ Multinomial(1, $\theta_m$);
        Sample word $w_{m,n} \sim$ Multinomial(1, $\Phi_{z_{m,n}}$);
```

5.2.2 Inference

The goal in the LDA algorithm is to find the document-topic matrix $\theta_{M\times K}$ and the topic-word matrix $\Phi_{K\times N}$. Thus, the key inferential problem is to find the hidden variables by maximizing the posterior below

$$p(\Phi_{1:K}, \theta_{1:M}, Z_{1:M} | w_{1:M}) = \frac{p(\Phi_{1:K}, \theta_{1:M}, Z_{1:M}, w_{1:M})}{p(w_{1:M})} \tag{5.1}$$

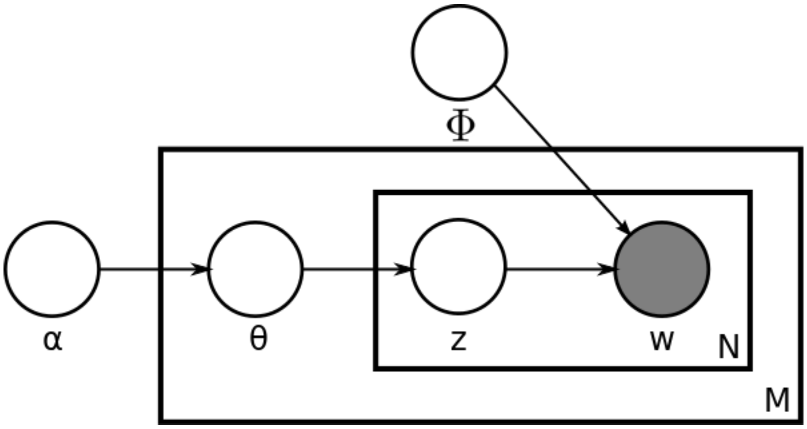


Figure 5.1: Plate notation of LDA without Dirichlet distribution on the topic-word matrix Φ Blei et al. (2003)

Computing the posterior is intractable due to the coupling between θ and z. Therefore, an approximation method is used to find a lower bound on the log likelihood. In Srivastava and Sutton (2017), they introduce two variational parameters λ and η to break the coupling between θ and z such that the approximate posterior would be $q(\theta, z|\lambda, \eta) = q_\lambda(\theta)\prod_n q_\eta(z_n)$. Then, in the VAE case, the optimization problem is to maximize 5.2 which is actually the evidence lower bound (ELBO) [10]. (Equation 5.2 is the RHS of Equation A.4. More detailed explanation of the loss function in VAE is provided in Appendix A.4)

$$L(\lambda, \eta|\alpha) = -D_{KL}[q(\theta, z|\lambda, \eta)||p(\theta, z|\mathbf{w}, \alpha)] + E_{q(\theta, z|\lambda, \eta)}[\log p(\mathbf{w}|z, \theta, \alpha)] \tag{5.2}$$

The first term in 5.2 tries to match the approximate posterior distribution to the true posterior and the second term is the *reconstruction term*.

5.2.3 Reparameterization Trick in AVITM

In order to find the parameters in the model, Reparameterization Trick (RT) is required to allow backpropagation in the VAE. Although the prior in LDA is Dirichlet, it is hard to develop an effective reparameterization function for the RT (Srivastava and Sutton, 2017). Consequently, Srivastava and Sutton (2017) resolved this issue by constructing a Laplace Approximation to the Dirichlet prior. The details of this approximation is beyond the scope of this thesis and interested reader can refer to MacKay (1998). This approximation involves changing the simplex basis to

[10] Variational Autoencoder is explained in Appendix A.4

the softmax basis and the covariance matrix becomes diagonal for large K . Therefore, $p_{\alpha}(\theta)$ will be approximated in the softmax basis by a logistic normal with mean μ_1 and covariance Σ_1 [11]. In order to learn these parameters, two inference networks f_{Σ} and f_{μ} are defined for inference where their outputs are in $\mathbb{R}^K$. Note that since the Covariance is diagonal, f_{Σ} has K elements. Then, for a document w, $q(\theta)$ is a logistic normal with mean $\mu_0 = f_{\mu}(w)$ and $\Sigma_0 = diag(f_{\Sigma}(w))$. Finally, samples are generated from $q(\theta)$ by sampling $\varepsilon \sim N(0, I)$ and $\theta = \sigma(\mu_0 + \Sigma_0^{1/2}\varepsilon)$, where σ is a softmax function. Following the results in Srivastava and Sutton (2017), ELBO can therefore be written as below

$$L(\Theta) = \Sigma_{m=1}^{M}[-(\frac{1}{2}\{tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) - K + \log\frac{|\Sigma_1|}{|\Sigma_0|}\}) + \\ E_{\varepsilon\sim N(0,\ I)}[w_m^T \log(\sigma(\Phi)\sigma(\mu_0 + \Sigma_0^{1/2}\varepsilon))]] \tag{5.3}$$

The first line of the Equation 5.3 is computing the KL divergence between q and $\hat{p}$ and the second one is the reconstruction error.
In Appendix A.5, we explain the relationship between a VAE and the AVITM.

5.2.4 ProdLDA

ProdLDA, or Latent Dirichlet Allocation with Product of Experts, is another model that we tried in this thesis that is a little different from LDA and was introduced by Srivastava and Sutton (2017). In LDA, the probability of words given document-topic and topic-word matrices is a mixture of multinomials [12]. A problem with this assumption that is common for all mixture models is that it can never make predictions sharper than the components that are being mixed. Hinton and Salakhutdinov (2009) illustrates this problem by providing an example. The combination of the topics “government”, “mafia” and “playboy” gives very high probability to a word “Berlusconi” while the probability of this word may not be high in any of these individual topics. We will elaborate the problem in the next paragraph.
The probability of n-th word in document m given θ and Φ (i.e. $p(w_{m,n}|\theta, \Phi)$) is the dot product of the m-th row of θ and the n-th column of $\sigma(\Phi)$. Because the rows of $\sigma(\Phi)$ rely in a simplex (or a softmax), the result of this dot product cannot be higher than the individual elements of this dot product. Srivastava and Sutton (2017) overcome this problem by substituting

[11] Although the idea is similar to the correlated topic model (Blei and Lafferty (2007)) – because of approximating Dirichlet priors with the logistic normal distribution, no correlation is assumed between the topics and the covariance matrix is diagonal.

[12] A mixture model is a collection of probability distributions for representing the presence of subpopulations within an overall population.

word-level mixture with a weighted product of experts which by definition is capable of making sharper predictions than any of the component experts (Hinton, 2002). As a result, in ProdLDA, there is no simplex constraint on Φ, i.e. $w_{m,n}|\theta,\Phi \sim Multinomial(1,\sigma(\Phi\theta))$, whereas in LDA, $w_{m,n}|\theta,\Phi \sim Multinomial(1,\sigma(\Phi)\theta)$. The connection of this modification to the product of experts is explained in Srivastava and Sutton (2017).

5.3 Inference for a New Document

After training the network with the train data, it can be used to make inferences about a previously unseen document i.e. finding the distribution of the topics, or θ, for that document. The inference is very simple in AVITM. After preprocessing the previously unseen document, and constructing the bag of words, which consist of both words and CUIs, the bag of words are used as an input to the trained network. After that, the mean μ_0 and the covariance matrix Σ_0 are derived (see Figure A.8). The mean and the covariance specify the distribution that the latent vector θ comes from, and they are used to sample the latent vector θ. Because of the randomness in this procedure, it can be repeated several times, and the average of the θ's can be used as the document-topic vector of the new document.

5.4 The Advantages of the Proposed Pipeline

As discussed in Section 1.4, the LDA/ProdLDA model presented by Srivastava and Sutton (2017), does not treat synonyms equally. The reason is that the optimization cost in the model is the sum of Reconstruction Loss (RL) and the KL divergence. In the original implementation, the synonyms e.g. “canine” and “dog”, have different one-hot encodings. As a result, when decoding, if the model gives a high probability to “canine” and a low probability to “dog”, the reconstruction loss will be non-zero if the true word was “dog”. However, UMLS gives the same CUI, C1280551, to both of them. In a nutshell, the model is punished when producing a synonym. Hence, with our preprocessing framework, both of these words will be replaced with the same CUI, and MedLinker ensures both of these terms convey the same meaning. In other words, it disambiguates their word senses using their contexts. Moreover, if the trained model sees the word “canine” in a new document, it will be able to use it for making an inference, i.e. assigning a topic distribution to the (new) document. This is because “canine” was replaced by the CUI “C1280551” during preprocessing and “C1280551” was already in the *vocabulary* set of the model.

The second advantage of replacing the entities with their CUI is to capture n-grams in our model without increasing the *vocabulary* size. Consider the entity “Myeloid derived suppressor cells”. If we want this entity to be in the dictionary of our model, we need to add bigrams, 3-grams, and 4-grams to our model, which highly increases the dictionary size. In some cases, the entity strings are more complex. They contain digits, symbols, or stopwords–like “Disease Grade 2”. These are all the things that are usually removed during preprocessing because they cause topic models to produce meaningless topic words. Furthermore, some of the entities are abbreviated, like “MERS”. It is natural to expect “MERS” and “Middle East Respiratory Syndrome” to have the same probability in the topic-word distribution. With our framework, all of these complicated entities are simply replaced by a single CUI.

In the next chapter, we show the results from applying the proposed pipeline to the CORD-19 dataset. The topics and the calculated scores are provided. The scores include topic coherence (C_v, UMASS, UCI , and NPMI)– and topic diversity (RBO and TD) measures.

Chapter 6

Experimental Results

We applied the preprocessing framework– described in Section 5.1– on $\sim 85,000$ English abstract documents from the CORD-19 dataset. In the next section, we talk about the CORD-19 dataset and the subset of this data that we used in Section 6.1. The AVITM– explained in Section 5.2– was trained for 500-1000 epochs using $\sim 36k$ of the preprocessed documents. We call the set of the preprocessed documents that we used for training the *internal* corpus whereas we call the remaining preprocessed documents ($\sim 47k$) the *external* corpus. The reason for this naming is that we used the train data (or the *internal* corpus) for calculating the *internal* coherence value and the test data (or the *external* corpus) for calculating the *external* coherence value.

Below are the different settings we experimented with for training:

- The size of the *vocabulary*– containing both words and CUIs– was more than 113k. However, we truncated the *vocabulary* to different sizes for experimenting. For instance, the top 10k or 2k most frequent words (and CUIs) were selected to use as the (truncated) *vocabulary*. We also tried training the model without truncating the *vocabulary*.

- We used different numbers of topics for training, such as 10, 15, 20, 25, 30, 35, 40, 45, and 50.

- We tried LDA with some of the above settings, but since it had far worse results than the ProdLDA in most cases we skipped trying the other settings.

Even though the goal in the AVITM is minimizing the sum of the reconstruction loss and the KL divergence, as we will see in this chapter, minimizing this loss does not necessarily guarantee coherent and diverse topics. This is why we saved the best model according to the coherence value, $\mathbf{C_v}$. [1]. To do so, at every 20 epochs, we evaluated the trained model by calculating the (internal) coherence value (C_v) [2] using the train data. The model was saved at the best epoch to avoid overfitting. The best model for each setting was then evaluated based on the external corpus, i.e. the held-out preprocessed set of documents.

In Chapter 4, we explained the common scores that are used for the evaluation in topic modeling. In all of these scores, only the topic-word matrix is evaluated and it is evaluated from the two aspects of *Topic Diversity* and *Topic Coherence*. For the former category, we chose the scores *Topic Diversity (TD)* and *Rank-Biased Overlap (RBO)*. For the latter one, we chose Coherence Value (C_v), *UMASS*, *UCI*, and *NPMI* because these are the scores that Röder et al. (2015) showed to have the highest correlation with the human judgement. In the next section, we explain the CORD-19 dataset and the subset of the abstract documents from this data that we used in our framework. Then we show the scores and topics obtained from applying the proposed framework to this subset under the different settings that we explained above.

6.1 COVID-19 Open Research Dataset (CORD-19)

CORD-19 (COVID-19 Open Research Dataset) Wang et al. (2020a) is a publicly available dataset with more than 400k scholarly papers related to COVID-19 published by the Allen Institute for AI. CORD-19 is provided to the global researchers to apply the state-of-the-art algorithms in natural language processing and other AI techniques.
CORD-19 is released daily and is growing very fast. The first version of CORD-19 was released on March 16th, 2020, and in January 2021, it has more than 400,000 documents. Each version of this corpus is tagged with a datestamp (e.g. 2020-05-26). Each release includes 'changelog', 'cord_19_embeddings.tar.gz', 'document_parses.tar.gz', and 'metadata.csv' files. In this work, we only used the 'metadata.csv' file. This file contains metadata for all the extracted CORD-19 papers. The papers are collected from multiple sources according to the keywords such as "COVID-19", "Coronavirus", "Corona virus", "2019-nCoV", "SARS-CoV", "MERS-CoV", "Severe Acute Respiratory Syndrome", "Middle East Respiratory Syndrome", and so on.

[1] Röder et al. (2015) showed that this measure has the highest correlation with human judgment. It is explained with more details in Appendix 4.3

[2] Coherence value involves estimating the co-occurrence probability of a topic's top words. This can be done either based on an internal corpus, i.e. the corpus that was used to train the data or an external corpus. Usually, a big and general corpus such as Wikipedia is used as an external corpus. However, since our *vocabulary* and consequently the bag of words in the train data contain both words and CUIs, we could not use a ready-to-use corpus. For this reason, we used the held-out data instead of the external corpus, although it is not very big.

The *metadata* file has the following columns: [3]

- cord_uid: The unique identifier assigned to the CORD-19 paper.
- sha: The SHA-1 of all PDFs associated with the CORD-19 paper.
- source_x: The source of the paper which is either 'ArXiv', 'Elsevier', 'PMC', or 'WHO'.
- title: The title of the paper
- doi: The *doi* of the paper
- pmcid: The paper's ID on PubMed Central.
- pubmed_id: The paper's ID on PubMed.
- license: The most permissive license associated with the paper.
- abstract: The paper's abstract
- publish_time: The published date of the paper
- authors: The authors of the paper
- journal: The paper's journal
- who_covidence_id: ID assigned by the WHO for this paper
- arxiv_id: The arXiv ID of the paper
- pdf_json_files: Path to the papers in JSON format (parsed from the PDF of the paper).
- pmc_json_files: Path to the papers in JSON format (parsed from the XML of the paper).
- url: All URLs associated with this paper.
- s2_id: The Semantic Scholar ID for this paper.

From the time that the data were published, many NLP tasks have been applied to this data, including, but not limited to, recommendation, information extraction, knowledge graphs, question answering, and summarization. Information extraction tasks involve finding (UMLS) entity mentions from the texts. In most cases, the extraction was done using Scispacy package in Python Neumann et al. (2019). However, Medlinker– which showed better results in (UMLS) entity extraction– has not been used on the CORD-19 data to the best of our knowledge.
It is hard to tell the exact number of papers in each version of the CORD-19 dataset because some entries of this dataset do not contain any 'abstracts'. For instance, the entry with 'cord_id=i5fcedbo'

[3] The provided information about the CORD-19 dataset is extracted from here.

has the ‘url = https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1.../’ and ‘title = Scientific Abstracts’. However, no abstract for this paper is provided. In fact, the page associated with this url is not found. We call these papers ‘empty’ documents.

6.1.1 Extracting Subsets of CORD-19 dataset for Train and Test

We used the ‘2021-01-11’ release of the CORD-19 dataset. That is, we used the dataset that was released on January 11th, 2021. This was the latest version when we downloaded this dataset. This version of the data contains 414,020 entries that 115,815 of the abstract documents are empty. Because of the time constraints, we were not able to apply our framework on all of the documents. Rather, we used the first 36,832 non-empty and English abstracts as our train data and the second 47,241 non-empty and English abstracts as our test data. To detect the language of the abstracts, we used the ‘langdetect’ package in python.

6.2 Scores

In this section, we show the *topic diversity* and *topic coherence* scores that we calculated when applying the proposed approach different settings to the subset of the abstract documents from the CORD-19. As discussed in the beginning of this chapter, different settings include 1) LDA vs ProdLDA 2) Different vocabulary sizes: 2k, 10k, and full vocabulary 3) The hyperparameter *Number of Topics* set to different values: 10, 15, 20, 25, 30, 35, 40, 45, and 50

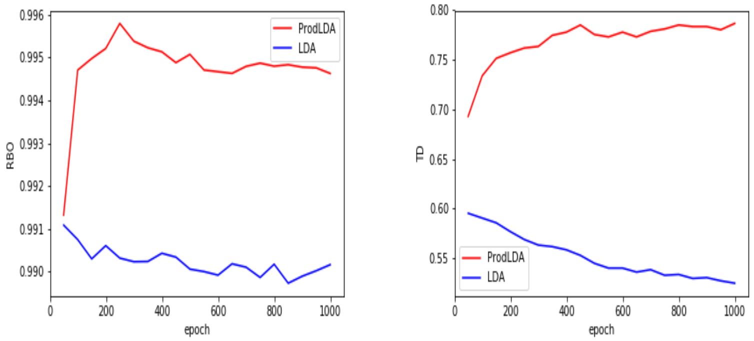
6.2.1 LDA vs ProdLDA

We compared LDA with ProdLDA using different settings. In all the settings, LDA had the worse results according to *topic diversity* and *topic coherence* measures– despite having the lower *train loss* (see Figure 6.2a). The settings shown in this section are *vocabulary* size equal to 2k and the model was trained with 50 topics for 1000 epochs. For calculating the scores, we used the top 10 words from each of the topics.

ProdLDA has much higher RBO (Figure 6.1a) and TD (Figure 6.1b) [4]. ProdLDA has higher coherence value (Figure 6.3a), higher NPMI (Figure 6.3b), and higher UMASS in magnitude (Figure 6.3d) [5]. Only, the UCI score is worse in ProdLDA (Figure 6.3c).

[4] In both of these score, the higher is the better.

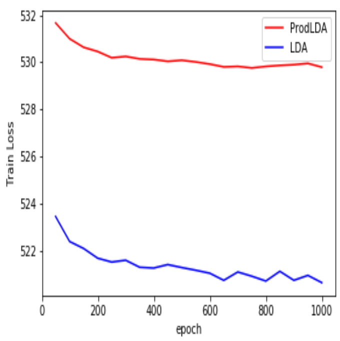
[5] The magnitude of UMASS should be considered and the higher means the more coherent topics.



(a) RBO score

(b) TD score

Figure 6.1: LDA vs ProdLDA by Topic Diversity



(a) Train Loss

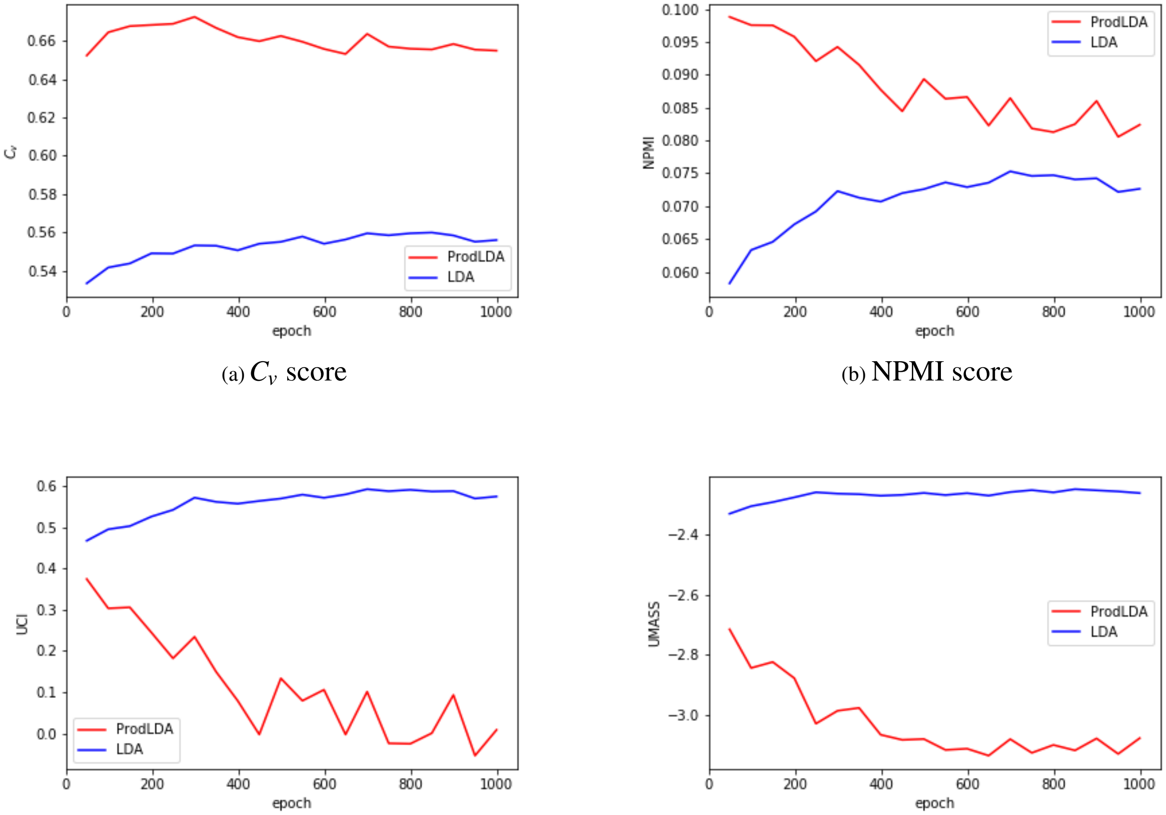
Figure 6.2: LDA vs ProdLDA

6.2.2 Comparing Vocabulary sizes

In this section, we will show the results of different *vocabulary* sizes with different numbers of topics. We experimented with truncating *vocabulary* to **2k** and **10k**, and no truncation (**full**) i.e. *vocabulary* size equal to $\sim 113k$. For each of these *vocabulary* sizes, we experimented three number of topics: **10**, **25**, and **50**. We trained the model that has a full *vocabulary* size for the smaller number of epochs, 250 instead of 500 since each epoch of training for this network takes so long. Also, the best epoch regarding C_v often occurs in the first 200 epochs. Hence, 250 epochs are usually sufficient. For each setting, we calculated coherence scores, which means C_v, NPMI, UMASS, and UCI using the internal corpus (the corpus used to train the model) We also calculated the topic diversity scores, i.e. TD, and RBO.

As depicted in Figure 6.4, the C_v (a) of the setting with 10k *vocabulary* and 10 topics (10k-10) is higher (blue-solid line). However, it drops significantly for the larger number of epochs when the number of topics is higher. This gets worse for the settings with the **full** *vocabulary* (green lines) and the higher number of topics. It seems that the model overfits sooner in cases with large *vocabulary* sizes and a large number of topics. We can see a similar phenomenon with other scores as well. The NPMI (b) for a large number of documents and large *vocabulary* size drops significantly as the number of epochs grows. Nevertheless, the NPMI (b) (as well as UCI (d)) for the 10k-10 setting (solid-blue line), is not very different from the settings with *vocabulary* size = 2k (red lines). The only score which is better for **full** *vocabulary* size is the UMASS (d) [6];

[6] For the UMASS score, the larger the absolute value is, the more coherent the topics are.



(a) C_v score

(b) NPMI score

(c) UCI score

(d) UMASS score

Figure 6.3: Comparing LDA with ProdLDA according to the Topic Coherence

however, since Röder et al. (2015) showed that C_v, and NPMI have the highest correlation with human judgment, we should not allow the results from UCI and UMASS affect our evaluation that the settings with the *vocabulary* size = 10k lead to the most coherent results. Moreover, in the next section, we will use C_v to determine the best coherent topics.
Figure 6.5 shows the results from the *Topic Diversity* point of view. Again. the setting with 10k *vocabulary* size and 10 topics (10k-10, blue-solid line) leads to both the higher RBO (a) and the higher TD (b). As opposed to the *Topic Coherence* scores, the *Topic Diversity* ones increase by the number of epochs. This observation necessitates a better loss function for the network that can consider the topic coherence, while not threatening the topic diversity.

6.2.3 Choosing the Best Number of Topics

Following the results from the previous section, that 10k-vocabulary settings lead to more coherent topics, we also calculated the *external* coherence scores for this setting with the different number of topics. We used $47,241$ preprocessed documents from the CORD-19 dataset that were not used in the training to calculate the *external* coherence value C_v. The result is shown in Figure 6.6. Although the *external* coherence value is lower than the *internal* one, they have a similar trend, and they both suggest the 10k-10 setting provides the most coherent topics.
Note that the range of the topic coherence (external or internal) for the different number of topics is from 0.6-0.72 which is a promising range of scores in the topic modeling domain.

6.2.4 Number of Documents by Dominant Topic

We obtained the topic distribution for the documents in the test set (external corpus) using the trained model with the 10k-20 setting. In this setting, the vocabulary size is truncated to 10k and the hyperparameter denoting the number of topics is set to 20. We used this number of topics to have more specific topics. The topic distribution of a document contains the probability of each topic in that document. As a result, each document may be related to several topics. Albeit, one can assign the topic with the highest probability to the document, which is what we did in this experiment. For each document in the test set, we assigned the topic with the highest probability in the corresponding topic-document vector to it. We call this topic the dominant topic. Then, for each of the 20 topics, we calculated the frequency (in percentage) of the documents that have this topic as their dominant topic. Figure 6.7 shows this frequency plot. The most frequent dominant topics are Topic 16, 5, 8, 18, and 15. In Table 6.1, the top 26 (which is an arbitrary number) words for each of these topics are shown.

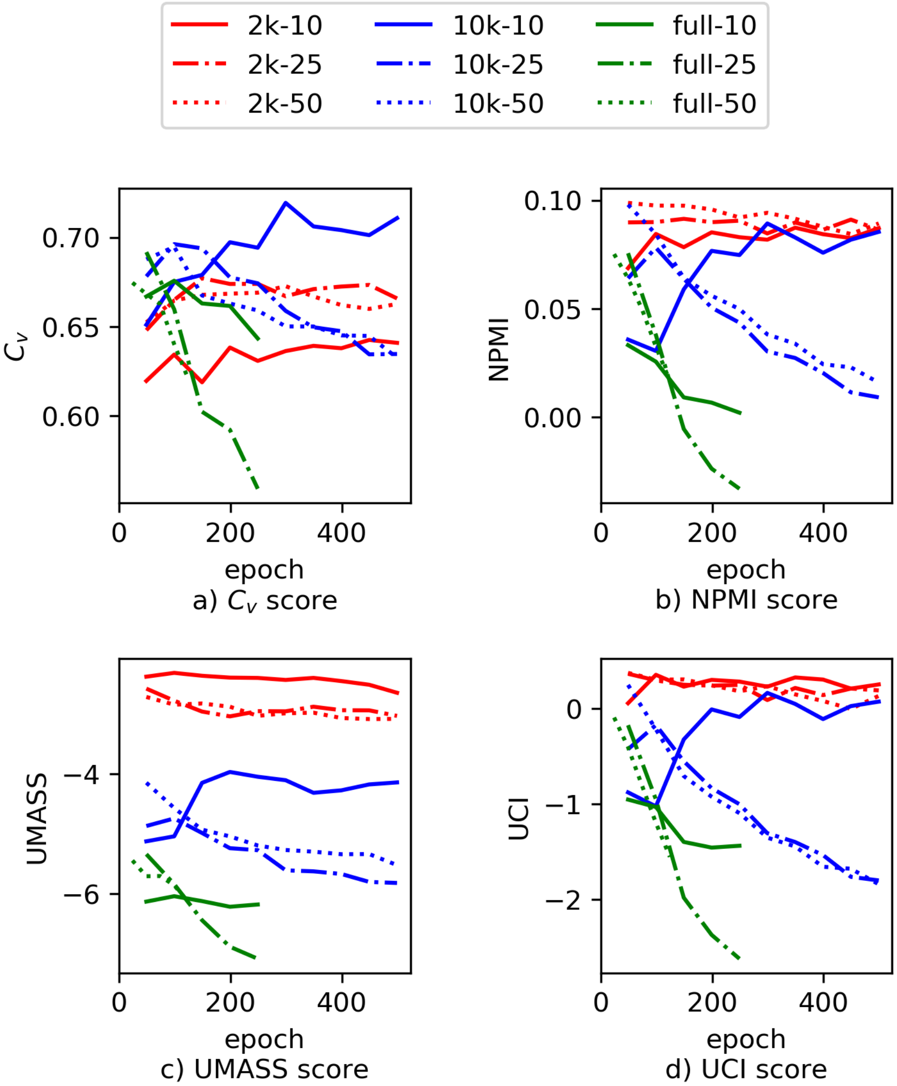


Figure 6.4: Topic Coherence for different (vocab size- ntopics) settings and 250-500 epochs

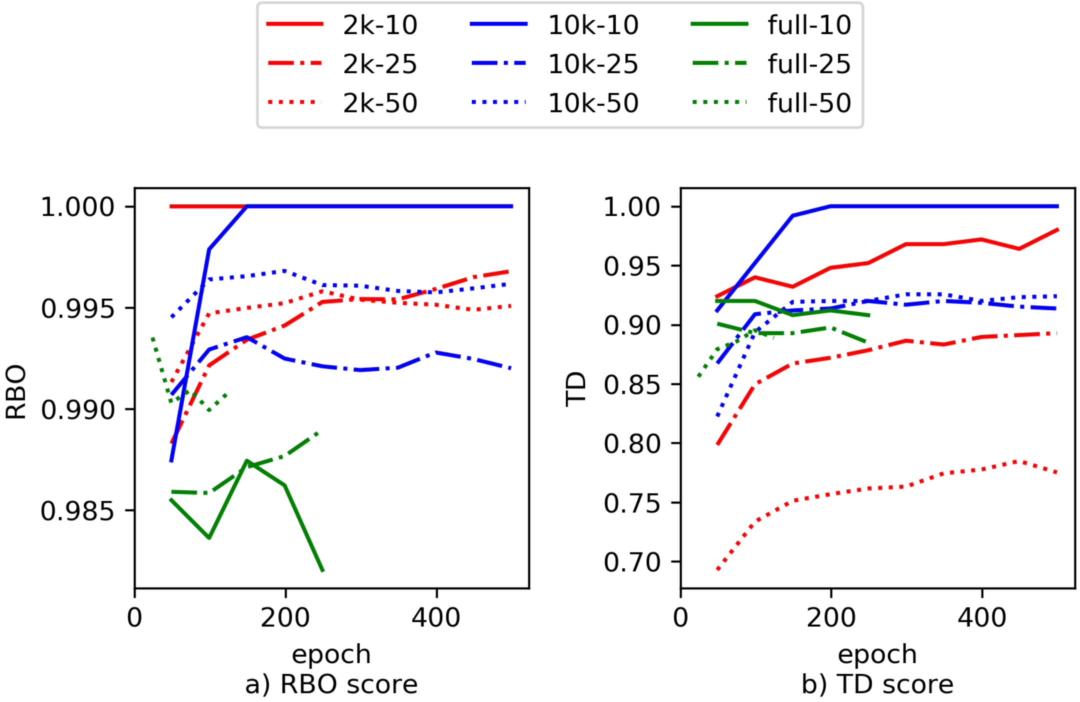


Figure 6.5: Topic Diversity for different (vocab size- ntopics) settings and 250-500 epochs

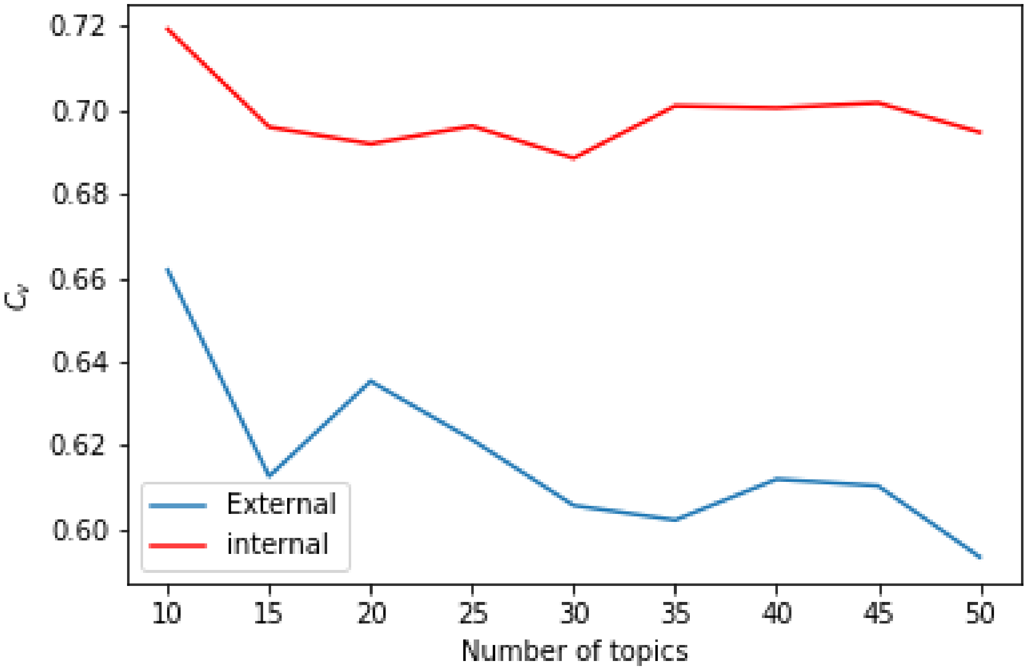


Figure 6.6: C_v vs number of topics for the 10k-vocabulary setting

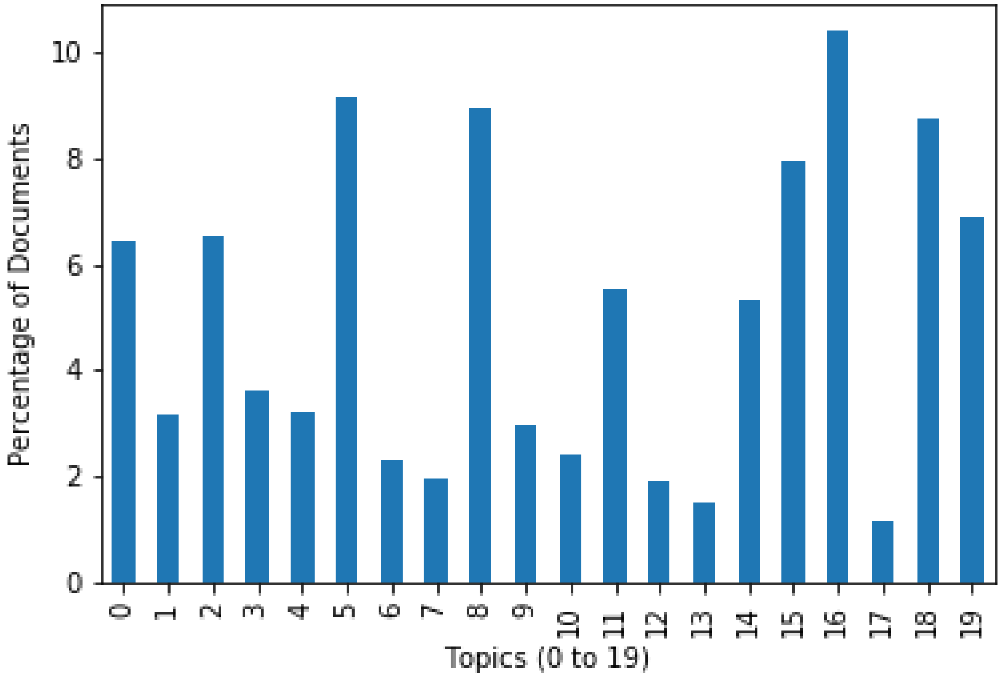


Figure 6.7: Document Frequencies (percentage) by Dominant Topics when the Number of Topics is set to 20

Topic 16	health, C0018696:Health Care Systems, challenge, C1708333:Health Care Organization, public, C0009450:Communicable Diseases, need, emergency, provide, key, development, threat, plan, C0699943:Public health service, global, C0086388:Health Care, recommendation, develop, preparedness, pandemic, resource, outbreak, C0454664:Country, management, C0035168:research, response
Topic 5	child, cause, C0011900:Diagnosis, case, C1457887:Symptoms, C3714514:Infection, clinical, common, severe, C0012634:Disease, C0010076:Coronaviridae, C0015967:Fever, present, manifestation, C0546788:Parainfluenza Virus 5, respiratory, C0039082:Syndrome, C0027442:Nasopharynx, C0035235:Respiratory Syncytial Virus Infections, presentation, C1446409:Positive, C0032285:Pneumonia, C0221423:Illness (finding), C0597404:Respiratory viruses, infection, C0038410:Streptococcus pneumoniae
Topic 8	C1257890:Population Group, pandemic, C0027361:Persons, covid-19, outbreak, number, epidemic, spread, C0017446:Geographic Locations, impact, measure, lockdown, period, C0021400:Influenza, C0546788:Parainfluenza Virus 5, social, estimate, case, age, distance, report, rate, C1306577:Death (finding), C0454664:Country, high, household
Topic 18	C0038492:student, support, train, care, virtual, C0679646:Participant, family, online, aim, C0086388:Health Care, need, practice, education, conduct, skill, program, provide, C1704312:Health Professional, C0282574:Intellectual Property, C0596545:Experience, C0376554:Knowledge, teach, work, C2603343:Study, C0025353:mental health, C0038495:Students, Medical
Topic 15	economic, crisis, argue, political, draw, right, market, policy, way, financial, economy, look, world, power, legal, cooperation, international, industry, conflict, bank, come, global, bring, chapter, contemporary, C0231224:Crisis

Table 6.1: Top Words for the most frequent Dominant Topics

6.3 Topics

Topics are represented with their most common words (words & CUIs). We chose the model with the *vocabulary* size equal to 10k and the number of topics equal to 25. Then, we selected the top 26 words from each topic to represent it. Although the most coherent topic is shown to be with 10 topics (see Figure 6.6), we used the 25-topics setting to have more specific topics. In Table 6.2, some of the topics with their top 26 words are shown. The words are sorted, meaning that one can consider the top 10 words for each topic simply by looking at the first 10 words. As evident in this table, each topic is represented by words and CUIs. Since CUIs are not understandable for humans, we also included the *most preferred* text of that UMLS entity. For instance, for the CUI C0021708, we also included its *most preferred text*, "intensive care unit", after the colon. As you can see, the *preferred texts* of most of the CUIs are n-grams, like "X-Ray Computed Tomography". Without using CUIs instead of these long bi-grams, we would have never managed to capture these important terms.
The topics in Table 6.2 are self-representative. The first one could be about "Image diagnosis of COVID-19", the second could be about "Mental Health Problems during COVID19", the third one is all about the "Mortality of Coronavirus", The fourth one is all about the "Global Economic Crisis". The other topics are also clear and a researcher may simply assign a theme to these topics.

Codes

The codes to reproduce the results in this chapter are available at:
https://github.com/DonyaHamzeian/BiomedicalTopicModelling

1	C0243095:Finding, C0430022:diagnostic procedure, value, C0041618:Ultrasonography, score, C0040405:X-Ray Computed Tomography, C0376519:Spectroscopy, Near-Infrared, correlation, predict, measurement, C0011923:Diagnostic Imaging, C0024485:Magnetic Resonance Imaging, measure, C0034108:Oximetry, Pulse, C0220825:Evaluation, accuracy, C0369768:Molecular oxygen saturation, C0032740:Positive End-Expiratory Pressure, C0032743:Positron-Emission Tomography, auc, volume, receiver, roc, coefficient, calculate, C0918012:Index
2	age, C0003467:Anxiety, factor, C0043210:Woman, C1257890:Population Group, female, year, behavior, adult, young, association, C0679646:Participant, C0027361:Persons, adolescent, C0025353:mental health, old, C0011570:Mental Depression, C0023974:Loneliness, C0034394:Questionnaires, social, associate, C0030971:Perception, anxiety, sociodemographic, C0010362:Cross-Sectional Studies, C0086132:Depressive Symptoms
3	mortality, patient, C0021708:intensive care unit, severe, associate, severity, C0809949:Admission activity, C0019993:Hospitalization, C0012634:Disease, high, covid-19, age, C0035648:risk factors, C1306577:Death (finding), clinical, median, comorbidities, C0006560:C-reactive protein, characteristic, admission, ci, C0060323:Fibrin fragment D, C0032285:Pneumonia, ratio, year, C0005516:Biological Markers
4	global, economic, threat, public, policy, health, supply, sector, C0242456:Policy, international, sustainable, governance, economy, climate, crisis, market, food, security, investment, disaster, national, industry, urban, political, draw, C0015176:Europe
5	C0042210:Vaccines, C0003320:Antigens, C0020971:Immunization, C0318793:Zika Virus, C1510800:Adenovirus Vector, C0003316:Epitopes, C0039194:T-Lymphocyte, protection, elicit, C0475463:Antibodies, Neutralizing, C0003250:Monoclonal Antibodies, C0004561:B-Lymphocytes, neutralize, C0301872:Immune response, C3714514:Infection, virus, protect, C0003241:Antibodies, vaccine, C0439663:Infected, infection, candidate, induce, C0042769:Virus Diseases, C0086418:Homo sapiens, C0042196:Vaccination
6	train, care, support, C0038492:student, education, program, C0086388:Health Care, need, virtual, skill, practice, online, team, communication, C0038495:Students, Medical, nurse, teach, C1704312:Health Professional, work, interview, C1522486:Professional Organization or Group, family, C0596545:Experience, access, pandemic, C0085537:Caregiver
7	C1171362:protein expression, activation, C0024432:macrophage, role, C0017262:Gene Expression, C0021368:Inflammation, C0007613:Cell physiology, C0007634:Cells, C0162638:Apoptosis, C0079189:cytokine, C0079904:NF-kappa B, C0021753:interleukin-1, beta, cell, C0037080:Signal Pathways, activate, C0007994:chemically induced, suppress, ali, C0225336:Endothelial Cells, C0596290:Cell Proliferation, lp, C0014597:Epithelial Cells, C0040690:Transforming Growth Factor beta, expression, ameliorate, C0013081:Down-Regulation
8	spread, epidemic, outbreak, C0242781:disease transmission, number, C0009450:Communicable Diseases, C0017446:Geographic Locations, estimate, contact, dynamic, reproduction, C0454664:Country, transmission, travel, C0012634:Disease, C0876936:Mathematical Model, forecast, C0237401:Individual, C0021400:Influenza, control, population, case, infectious, spatial, C0008115:China, C0679083:Simulations

Table 6.2: Top Words for Some Sample Topics

Chapter 7

Conclusion and Future Work

We introduced a pipeline for topic modelling on biomedical texts based on ProdLDA. With our preprocessing step, which involves using MedLinker, we demonstrated a solution that handles medical synonyms better than in ProdLDA and many other TM models. We used MedLinker for replacing the entities with their corresponding entity CUI from UMLS. MedLinker identifies the entities and links them to the UMLS by using the BERT model– which attends to the context of the entities– to also handle the WSD. Additionally, with MedLinker, we were able to include n-grams in our topic words without enlarging the *vocabulary* size. By applying our proposed approach to the CORD-19 dataset, we managed to achieve the internal topic coherence value of more than 0.7 and the external coherence value of more than 0.6. Likewise, the topic diversity scores were also notable.

Ongoing Work

Although we achieved good results in topic coherence and topic diversity, the topics should be further evaluated according to human judgement. On this basis, currently, we are *manually* evaluating the results obtained from applying the proposed TM approach to the CORD-19 dataset. The framework for human evaluation that we are using is what was proposed in Chang et al. (2009). This procedure includes two human evaluation tasks, *word intrusion* and *topic intrusion*.The former task helps to measure the coherency of topics. The latter task helps to quantify to the extent that the topic distributions of documents agrees with human association of topics with documents.

Future Work

Below, we will talk about some improvements that could be made to our model in future works:

- One problem with the existing topic models, that was also acknowledged by Gui et al. (2019), is the lack of consideration of the topic coherence measures during training. To make matters worse, as we saw in the results chapter, topic coherence and the network's loss (Reconstruction Loss + KL divergence) are sometimes negatively correlated, especially for a larger *vocabulary* size. Although for choosing the best model we determined according to the topic coherence (c_v) rather than the network's loss, this can be further improved by incorporating coherence value of the loss function or using reward signals like in Gui et al. (2019).

- The improvement in MedLinker enhances the preprocessing step and the topic models consequently. In Loureiro and Jorge (2020), they used several extensions of BERT for training MedLinker, like NCBI-BERT, SciBERT, and BioBERT. According to their report, SciBERT gave the best results. However, the results may still be improved by using UMLSBERT (Michalopoulos et al., 2020). UMLSBERT integrated the domain knowledge during the pretraining process by using the semantic group knowledge in UMLS for the embedding process and connecting the words that have the same CUI in the UMLS.

- It is natural to believe the true topic distributions of the CORD-19 documents evolve over time. An example for a time-dependant topic-word distribution may be the 'Pfizer'. For the topic 'vaccine', the word 'Pfizer' might have had a few probability within the first few months of COVID-19 emergence, whereas, at the time of writing this thesis, it might have a high probability. This calls for an algorithm that considers the changes in the topic models using time series analysis similar to Dynamic Topic Models (Blei and Lafferty, 2006).

References

Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. pages 13–22, 03 2013. URL https://www.aclweb.org/anthology/W13-0102.pdf.

Claus Boye Asmussen and Charles Møller. Smart literature review: a practical topic modelling approach to exploratory literature review. *Journal of Big Data*, 6(1):93, Oct 2019. ISSN 2196-1115. doi: 10.1186/s40537-019-0255-7.

Evgeniy Bart, Max Welling, and Pietro Perona. Unsupervised organization of image collections: Taxonomies and beyond. *IEEE transactions on pattern analysis and machine intelligence*, 33: 2302–15, 04 2011. doi: 10.1109/TPAMI.2011.79.

Michael Biggs, Ali Ghodsi, and Stephen A. Vavasis. Nonnegative matrix factorization via rank-one downdate. *CoRR*, abs/0805.0120, 2008. URL http://arxiv.org/abs/0805.0120.

David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL https://doi.org/10.1145/2133806.2133826.

David M. Blei and Michael I. Jordan. Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, page 127–134, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136463. doi: 10.1145/860435.860460.

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 113–120, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143859.

David M. Blei and John D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, Jun 2007. ISSN 1932-6157. doi: 10.1214/07-aoas114.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. ISSN 1532-4435. doi: http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993.

Olivier Bodenreider. The unified medical language system (umls): Integrating biomedical terminology, 2004. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC308795/pdf/gkh061.pdf.

Jonathan Chang and David Blei. Relational topic models for document networks. *Journal of Machine Learning Research - Proceedings Track*, 5:81–88, 01 2009. URL http://proceedings.mlr.press/v5/chang09a/chang09a.pdf.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-graber, and David Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 288–296. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/f92586a25bb3145facd64ab20fd554ff-Paper.pdf.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. doi: https://doi.org/10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/pdf/1810.04805.pdf.

Ekin Ekinci and Sevinç İlhan Omurca. Concept-lda: Incorporating babelfy into lda for aspect extraction. *Journal of Information Science*, 46(3):406–418, 2020. doi: 10.1177/0165551519845854.

L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531 vol. 2, 2005. doi: 10.1109/CVPR.2005.16.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998. ISBN 978-0-262-06197-1.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/pdf/1406.2661.pdf.

Jordan M. I. Tenenbaum J. B. Blei D. M. Griffiths, T. L. Hierarchical topic models and the nested chinese restaurant process. *In Advances in neural information processing systems*, page 17–24, 2004. URL https://papers.nips.cc/paper/2003/file/7b41bfa5085806dfa24b8c9de0ce567f-Paper.pdf.

Lin Gui, Jia Leng, Gabriele Pergola, Yu Zhou, Ruifeng Xu, and Yulan He. Neural topic model with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3478–3483, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1350.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018.

Geoffrey E Hinton and Russ R Salakhutdinov. Replicated softmax: an undirected topic model. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 1607–1614. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/31839b036f63806cba3f47b93af8ccb5-Paper.pdf.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014. URL https://arxiv.org/pdf/1312.6114.pdf.

D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788 – 791, 1999.

L. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei. Building and using a semantivisual image hierarchy. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3336–3343, 2010. doi: 10.1109/CVPR.2010.5540027.

Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 577–584, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143917.

Daniel Loureiro and Alípio Mário Jorge. Medlinker: Medical entity linking with neural representations and dictionary matching. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo

Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, *Advances in Information Retrieval*, pages 230–237, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45442-5.

David J. C. MacKay. Choice of basis for laplace approximation. *Mach. Learn.*, 33(1):77–86, October 1998. ISSN 0885-6125. doi: 10.1023/A:1007558615313.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing, 2016. URL https://arxiv.org/pdf/1511.06038.pdf.

George Michalopoulos, Yuanxin Wang, Hussam Kaka, Helen Chen, and Alex Wong. Umlsbert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus, 2020. URL https://arxiv.org/pdf/2010.10391.pdf.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/pdf/1301.3781.pdf.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew Mccallum. Optimizing semantic coherence in topic models. pages 262–272, 01 2011.

Sunil Mohan and Donghui Li. Medmentions: A large biomedical corpus annotated with umls concepts, 2019.

Christopher E Moody. Mixing dirichlet topic models and word embeddings to make lda2vec, 2016.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. *Proceedings of the 18th BioNLP Workshop and Shared Task*, 2019. doi: 10.18653/v1/w19-5034.

David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 680–686, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150487.

David Newman, Jey Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. pages 100–108, 01 2010. URL https://dl.acm.org/doi/pdf/10.5555/1857999.1858011.

Yulia Otmakhova, K. Verspoor, Timothy Baldwin, Simon Suster, and Jey Han Lau. Improved topic representations of medical documents to assist covid-19 literature exploration.

In *NLP4COVID@EMNLP*, 2020. URL https://www.aclweb.org/anthology/2020.nlpcovid19-2.12.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents, 2012. URL https://arxiv.org/pdf/1207.4169.pdf.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pages 399–408, 02 2015. doi: 10.1145/2684822.2685324.

G. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, S. Sohn, K. Schuler, and C. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association : JAMIA*, 17 5:507–13, 2010.

Josef Sivic, Bryan Russell, Andrew Zisserman, William Freeman, and Alexei Efros. Unsupervised discovery of visual object class hierarchies. pages 1–8, 06 2008. ISBN 978-1-4244-2242-5. doi: 10.1109/CVPR.2008.4587622.

Luca Soldaini. Quickumls: a fast, unsupervised approach for medical concept extraction. 2016. URL http://medir2016.imag.fr/data/MEDIR_2016_paper_16.pdf.

Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models, 2017. URL https://arxiv.org/pdf/1703.01488.pdf.

T.Hofmann. Probabilistic latent semantic analysis. *Proceedings of the Twenty-Second Annual International SIGIR Conference*, 1999. URL https://arxiv.org/pdf/1301.6705.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/pdf/1706.03762.pdf.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 977–984, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143967.

Chong Wang and David Blei. Collaborative topic modeling for recommending scientific articles. pages 448–456, 08 2011. doi: 10.1145/2020408.2020480.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, K. Funk, Rodney Michael Kinney, Ziyang Liu, W. Merrill, P. Mooney, D. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Brandon Stilson Stilson, Alex D Wade, Kuansan Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset. *ArXiv*, 2020a.

Rui Wang and Deyu Zhou. Open event extraction from online text using a generative adversarial network, 08 2019. URL https://arxiv.org/pdf/1908.09246.pdf.

Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. Neural topic modeling with bidirectional adversarial training, 2020b. URL https://arxiv.org/pdf/2004.12331.pdf.

William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28:20, 11 2010. doi: 10.1145/1852102.1852106.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018. URL https://www.aclweb.org/anthology/N18-1101.pdf.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

Finn Årup Nielsen, Daniela Balslev, and Lars Kai Hansen. Mining the posterior cingulate: Segregation between memory and pain components. *NeuroImage*, 27(3):520 – 532, 2005. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2005.04.034.

Appendix A

Detailed Technical Background

A.1 BERT: A Contextual Language Model

Bidirectional Encoder Representations from Transformers (Devlin et al., 2019) short for **BERT** is a bidirectional transformer-based language model that generated the state-of-the-art results in many NLP tasks, including Question Answering (SQuAD v1.1) (Rajpurkar et al., 2016), Natural Language Inference (MNLI) (Williams et al., 2018), and others. BERT uses the transformer architecture (Vaswani et al., 2017). However, as opposed to a vanilla transformer, which consists of an encoder and a decoder, BERT only uses the encoder. BERT is a bi-directional model that instead of reading the text inputs sequentially, either from left-to-right or from right-to-left, reads the entire sequence [1]at once. Bi-direction conditioning has a problem that the model sees each token indirectly. However, this problem was overcome by masking the tokens randomly and pre-training the network to predict the masked tokens. This task is called Masked Language Modelling (MLM). BERT is also pre-trained on the Next Sentence Prediction (NSP) task. In the NSP task, which is a binary classification task, the network is trained to predict whether sentence B follows sentence A. Figure A.1 shows how an input sequence is fed to the transformer encoder in BERT. After the sequence being tokenized using the WordPiece tokenizer (Wu et al., 2016)– with the *vocabulary* of size 30,000 tokens– each token is converted to a one-hot vector, or *Token Embeddings*. The tokens start with the [CLS] token indicating the beginning of the sequence and are separated with the [SEP] token used to separate the tokens of the sentence A and B. Another vector is also used to denote whether each token belongs to the sentence A or B–called *Segment Embeddings*. The position of each token in the sentence is also represented by a vector

[1] 512 is the maximum token size for BERT inputs.

called *Position Embedding*. Segment embeddings, position embeddings, and token embeddings are concatenated and fed to the network.

Figure A.2 shows the pre-training and the fine-Tuning phases. In the pre-training phase, the inputs as described above, are fed to the transformer, and then there is a softmax layer in the last layer over the entire *vocabulary* which is used to predict the masked token [2]. For the pre-training, they used the BooksCorpus (800M words) (**?**) and English Wikipedia (2,500M words). In the fine-tunng phase, additional layers are added on top of the BERT's architecture and the weights of the additional layer(s) are jointly trained with the pre-trained weights of the original layers according to the new tasks, like SQUAD, Named Entity Recognition (NER), MNLI, and so on.

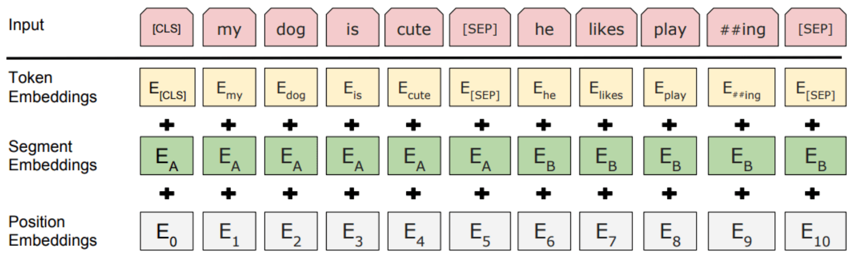


Figure A.1: Input representations in BERT.
Source: Devlin et al. (2019)

In Devlin et al. (2019), two variations of BERT were introduced. $BERT_{base}$ has 12 layers, each with 768 hidden nodes, and 12 self attention; whereas the $BERT_{large}$ has 24 layers each with 1024 hidden nodes, and 16 self-attention.

A.2 MedLinker

A very common task in biomedical texts is to extract entities. UMLS metathesaurus is the most common ontology for these tasks. Many methods have been introduced for extracting entities and linking to the UMLS, like QuickUMLS (Soldaini, 2016), Scispacy (Neumann et al., 2019), Ctakes (Savova et al., 2010), and so on. These methods give both the *span* of the text related to an entity (Entity Recognition) and (the most relevant) UMLS entity itself (Entity Linking).

Medlinker (Loureiro and Jorge, 2020) uses Natural Language Models (NLMs) for medical entity recognition and linking. Previous methods, like Scispacy, used the similarity between a query string and the strings of the UMLS entities along with their aliases. As an example, in

[2] 15 % of the tokens in each sequence are masked randomly

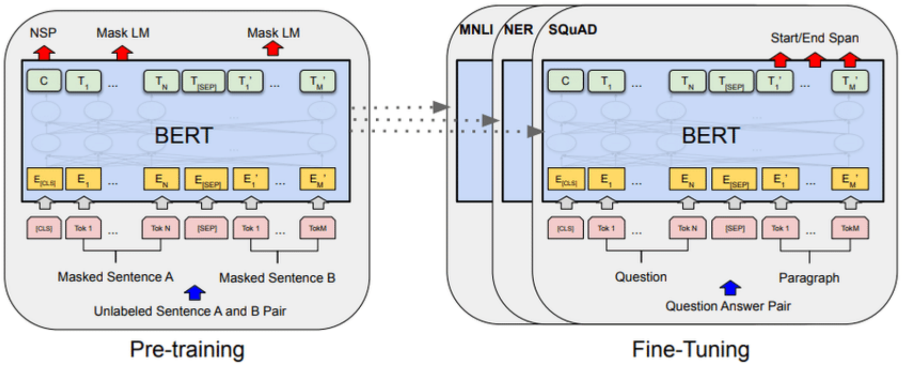


Figure A.2: Pretraining and Fine-Tuning BERT on different NLP tasks.
Source: Devlin et al. (2019)

the sentence "Inflammatory diseases of the respiratory tract are commonly associated with elevated production of nitric oxide (NO) and increased indices of NO -dependent oxidative stress", "Inflammatory diseases" is matched with the following UMLS entity based on the similarity between the query string "Inflammatory diseases" and the entity name " Inflammatory disorder" as well as all the aliases of the entity. Below, you can see one of the entities that Sispacy has extracted and linked from the previous example. It provides the CUI of the entity, the matching score (the higher is the better), TUI (ID of the UMLS semantic type or **STY**) of the entity, definition, and aliases.

CUI: C1290884
score: 1.0
Name: Inflammatory disorder
Definition: An infectious or non infectious disorder characterized by signs and symptoms derived from focal or extensive tissue infiltration by acute (e.g., polymorphonuclear) or chronic (e.g., lymphocytic-plasmacytic) inflammatory cells. Representative examples of infectious disorders include viral infections, bacterial infections, and parasitic infections. Representative examples of non-infectious inflammatory disorders include inflammatory bowel disease and inflammatory polyps.
TUI(s): T047
Aliases (abbreviated, total: 11): Inflammatory disorder, Inflammatory disorder, inflammatory disorder, Inflammatory Disorder, disorders inflammatory, Inflammatory disease, disease inflammatory, inflammatory disease, Inflammatory Disease, diseases inflammatory

Scispacy does not take the context into account and links the entities only based on the string level similarities. However, Medlinker considers both the string level similarity and the contextual similarity. Medlinker uses BERT, a contextual language model, to predict the spans (Entity Recognition) and the most relevant UMLS entities (Entity Linking). These two submodules have been trained using the MedMentions dataset (Mohan and Li (2019)), the largest mention level annotations targeting UMLS.

In Figure A.3, you can see the pipeline of the Medlinker model. On the left, the overview of the pipeline is outlined. The sentence goes through a submodule for **Mention Recognition** (NER) where the relevant spans are identified in the text. Then, the **Entity Linking** is done with two different approaches: **Dictionary Matching** (string level) [3] and **Contextual Matching** (with BERT). Finally, the **Post Processing** sub-module aggregates the scores derived from the previous sub-modules and predicts the final entities by applying logistic regression on the calculated scores. On the right side, the detailed view of the **Mention Recognition** and **Entity Linking (EL)** are depicted. Both of these sub-modules use the BERT embedded tokens. The former, NER, uses a BiLSTM with Conditional Random Field (CRF) to predict the spans and the latter, EL, uses a simple softmax layer to predict the entities.

Below is the result of applying MedLinker on an example sentence:

```
{’sentence’: ’Myeloid derived suppressor cells (MDSC) are immature myeloid cells with immunosuppressive
activity.’, ’tokens’: [’Myeloid’, ’derived’, ’suppressor’, ’cells’, ’(MDSC)’, ’are’, ’immature’,
’myeloid’, ’cells’, ’with’, ’immunosuppressive’, ’activity.’], ’spans’: [{’start’: 0, ’end’: 4, ’text’: ’Myeloid
derived suppressor cells’, ’st’: (’T017’, 1.0), ’cui’: (’C4277543’, 1.0)}, {’start’: 4, ’end’: 5, ’text’:
’(MDSC)’, ’st’: (’T017’, 0.54723495), ’cui’: (’C4277543’, 0.99998283)}, {’start’: 7, ’end’: 9, ’text’:
’myeloid cells’, ’st’: (’T017’, 1.0), ’cui’: (’C0887899’, 1.0)}]}
```

In the above result, the original sentence and the tokens are shown. The spans are identified using “start” and “end”. For example, the tokens 0 to 4, i.e. ‘Myeloid derived suppressor cells’ are predicted to be mostly related to the UMLS entity with cui = C4277543 and sty = T017. [4]

A.3 Autoencoder

Figure A.4 shows the architecture of an autoencoder and a sample from the MNIST dataset that is fed to the network. The encoder converts the original input into a latent representation, which usually has a lower dimension than the original space. Then, the decoder reconstructs the data

[3] Dictionary Matching in MedLinker is similar to the string level similarity in Scispacy. They both represent each query string and the UMLS entity names (with their aliases) with char n-gram vectors. Then the query string is matched with the UMLS entity having the highest cosine similarity in the char n-gram representation.

[4] The numbers in front of ‘cui’ and ‘st’ are the prediction probabilities for CUI linking and STY linking, respectively.

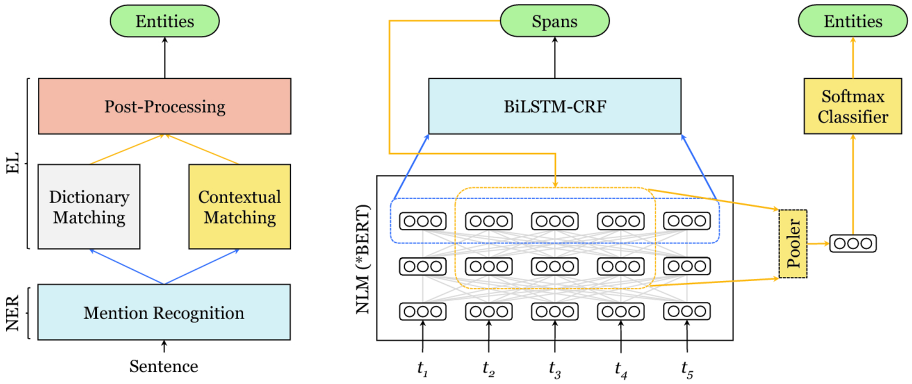


Figure A.3: Medlinker Pipeline

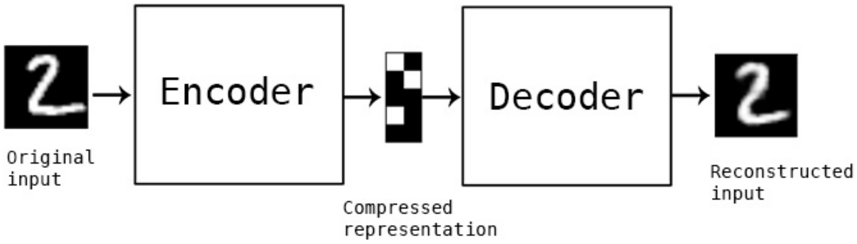


Figure A.4: Architecture of a Simple Autoencoder
Source: https://blog.keras.io/building-autoencoders-in-keras.html

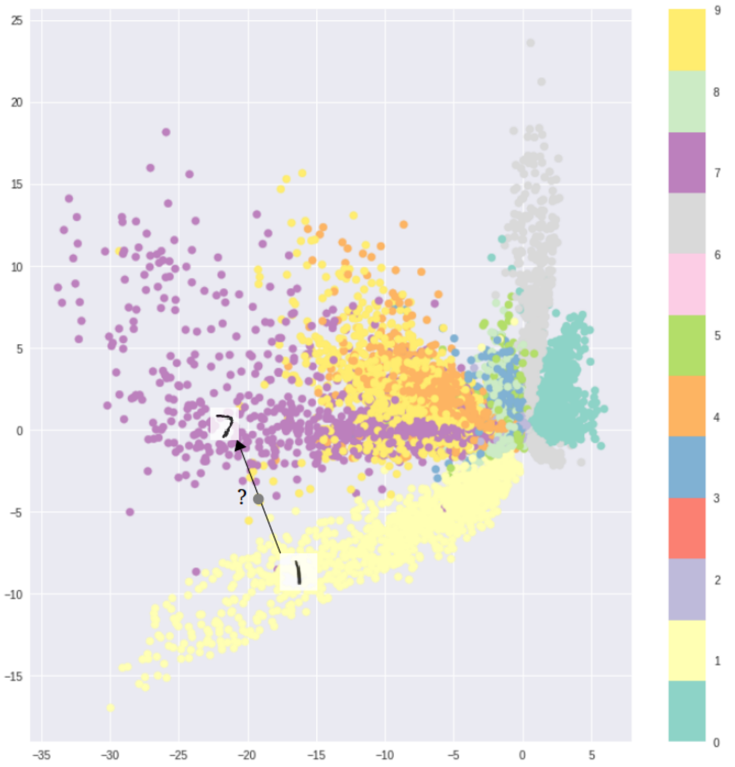


Figure A.5: The discontinuity in the latent space of the autoencoder trained on the MNIST data. If the sampled point in the latent space is not close to the latent representations of the train data, like the question mark (?) point, the decoder will generate an unrealistic output that is not similar to any of the MNIST digits. Source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

from the latent representation. The autoencoder by itself cannot be used as a *generative model*. The figure shows the 2D latent space derived from training a simple autoencoder on the MNIST data. As evident in the figure, this latent space is not continuous. As a result, one cannot generate a new data point by simple sampling from this latent space since the decoder will generate an unrealistic output.

A.4 Variational Autoencoder

Variational Autoencoder (Kingma and Welling, 2014), short for **VAE**, overcomes the sampling problem by mapping the inputs into a *distribution* rather than a *fixed* vector. For example, in the case that the *Normal distribution* is assumed, each input is mapped to mean, μ, and covariance, σ. We will briefly discuss the idea of VAE in the rest of this section.

We have data points $\mathbf{x}$ and the goal is to find the parameter, θ, of the distribution that $\mathbf{x}$ was generated from, by maximizing the log-likelihood.

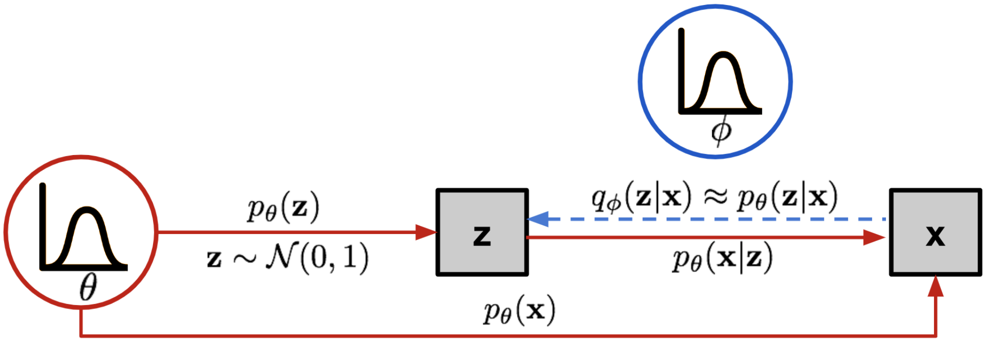

ϕ
θ
$p_\theta(\mathbf{z})$
$\mathbf{z} \sim \mathcal{N}(0,1)$
z
$q_\phi(\mathbf{z}|\mathbf{x}) \approx p_\theta(\mathbf{z}|\mathbf{x})$
$p_\theta(\mathbf{x}|\mathbf{z})$
x
$p_\theta(\mathbf{x})$

Figure A.6: The graphical model in the VAE. The solid lines show the generative processes and the dashed line shows the approximation of $p_\theta(\mathbf{z}|x)$ by $q_\Phi(z|\mathbf{x})$. Source: lilianweng.github.io

Hence, the goal is to optimize the following:

$$\sum_{x^{(i)} \in \mathbf{x}} \log p_\theta(x^{(i)}) \tag{A.1}$$

If we assume that the true parameter of this distribution is θ^*, a data point $x^{(i)}$ can be generated by following these steps:

1. First, sample a $z^{(i)}$ from a prior distribution $p_{\theta^*}(z)$.
2. Then, the data point $x^{(i)}$ is generated from a conditional distribution $p_{\theta^*}(x|z = z^{(i)})$.

Thus, we can write $p_\theta(x^{(i)})$ in Equation A.1, using the generative model, as below:

$$p_\theta(x^{(i)}) = \int p_\theta(x^{(i)}|z) p_\theta(z) dz \tag{A.2}$$

In many cases, $p_\theta(x^{(i)})$ is intractable. Therefore, an approximation function $q_\phi(z|x)$ is introduced. Trying to find $q_\phi(z|x)$ close to the posterior $p_\theta(z|x)$, and maximizing the log-likelihood of $\mathbf{x}$ being generated leads to finding the true parameters θ^* and Φ^*. Figure A.6 shows this process. In fact, $p_\theta(z|x)$ is the *encoder* and $p_\theta(x|z)$ is the *decoder*.

Finding the parameters θ and Φ such that $q_\phi(z|x)$ approximates $p_\theta(z|x)$, is equivalent to minimizing the kullback leibler (KL) divergence between these two distributions[5]. It can be shown that the KL divergence between q_Φ and p_θ can be written as in Equation A.3

$$D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})) = \log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\log p_\theta(\mathbf{x}|\mathbf{z}) \tag{A.3}$$

By rearranging Equation A.3, we have:

$$\log p_\theta(\mathbf{x}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) \tag{A.4}$$

The LHS is exactly what we want to maximize, i.e. maximizing the log-likelihood of $\mathbf{x}$ being generated and minimizing the distance between the true and the approximate posterior distributions. The first component of the RHS is usually called the *Reconstruction Loss*. The loss function ($L_{\mathrm{VAE}}(\theta,\phi)$) therefore is the negative of the RHS in Equation A.4:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} L_{\mathrm{VAE}} \tag{A.5}$$

In Variational Bayesian methods, this loss function is known as the variational lower bound, or **Evidence Lower Bound (ELBO)** [6].

Reparameterization Trick

The expectation term in the loss function L_{VAE} requires generating samples from $z \sim q_\Phi(z|x)$. In Figure A.7 you can see that training the VAE leads to taking gradient with respect to $\mathbf{z}$. However, we cannot take derivatives w.r.t. a non-deterministic variable. Reparameterization Trick (RT) is introduced to enable backpropagation. In the multivariate Gaussian with a diagonal covariance, depicted in Figure 2.1, the random variable $\mathbf{z}$ is generated using an auxiliary random

[5] $D_{KL}(P||Q) = E_{z\sim P(z)} \log \frac{P(z)}{Q(z)}$

[6] The "lower bound" in the name comes from the fact that the KL divergence is non-negative and $\log p_\theta(\mathbf{x})$ is constant w.r.t. q_Φ. Therefore, minimizing the loss function is equivalent to maximizing the lower bound of the probability of generating real data samples.

$$-L_{\mathrm{VAE}} = \log p_\theta(\mathbf{x}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})) \leq \log p_\theta(\mathbf{x})$$

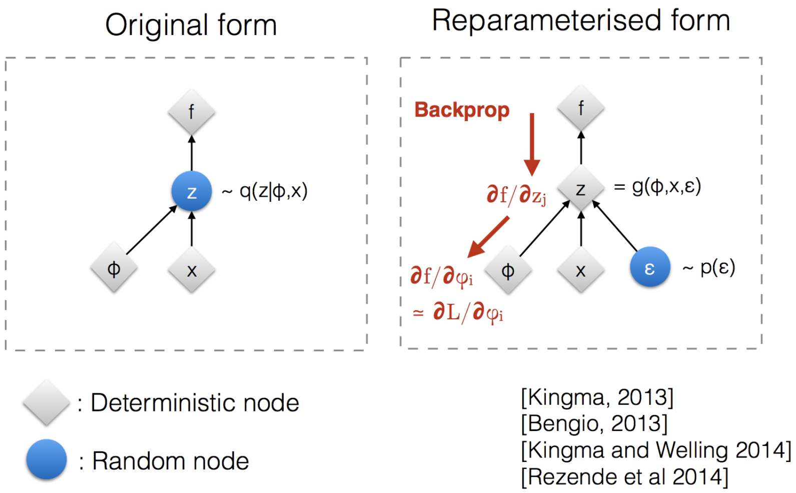


Figure A.7: Reparameterization Trick is required to enable the backpropagation.
Source: Kingma Welling, NIPS workshop 2015

variable sampled from the standard Gaussian distribution combined with the deterministic variables mean, μ, and covariance, σ as in A.6

$$
\begin{aligned}
\mathbf{z} &\sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \mathbb{N}(\mathbf{z};\mu^{(i)},\sigma^{2(i)}I) \\
\mathbf{z} &= \mu + \sigma \odot \varepsilon, \text{ where } \varepsilon \sim \mathbb{N}(0,I)
\end{aligned}
\tag{A.6}
$$

$\odot$ is the element-wise product of the two vectors.

A.5 The relationship between VAE of and AVITM

Recall that in Section 5.2.2, we explained that the prior $p_\alpha(\theta)$ is approximated by a logistic normal with mean μ_1 and covariance Σ_1.
As depicted in Figure A.5, the BoW vector(w) of document i is given to the network. The encoder encodes it to two latent vectors i.e. μ_0 and Σ_0. Then, a random vector, ε, is sampled from standard Gaussian. After that, the sampled latent vector which has the interpretation of the "topic distribution of document i" (the i-th row of doc-topic matrix) is derived using this formula: $\theta = \sigma(\mu_0 + \Sigma_0^{1/2}\varepsilon)$. θ and μ_0 both have the dimension k (the number of topics). The decoder has the job to reconstruct w. However, it outputs the "probability" of each word in document i. meaning that each element of w' is between 0 and 1, denoting the probability of that word

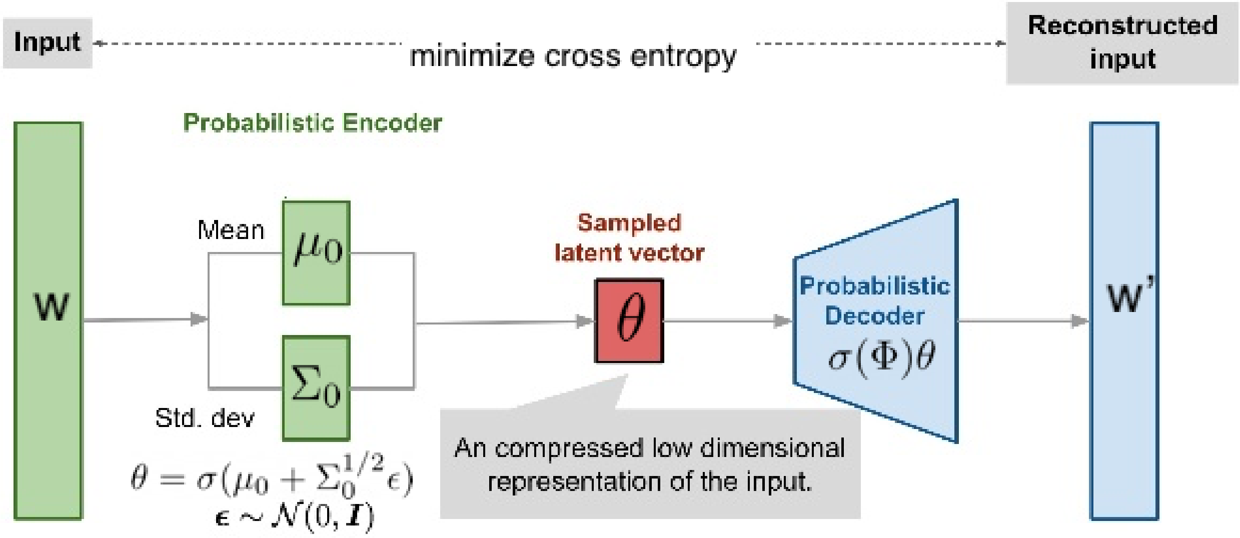


Figure A.8: Equivalency

in document i. According to the generative process of LDA, given the word-topic matrix, and the topic distribution of document i, θ, the probability of each word in document i will simply be the product of these two and this is because the probability of a word equals the probability of having topic z times the probability of that word in topic z and sum over all topics. In other words, for a document, the probability of word w_n is $p(w_n) = \sum_z p(w_n|z)p(z)$. Therefore, the decoder has the job to learn the word-topic matrix, Φ, so that the softmax of Φ, i.e. $\sigma(\Phi)$, times θ will have a very low cross-entropy loss with w that is minimizing the Reconstruction loss and the KL divergence. Note that since in this implementation of LDA only one Dirichlet process (on the doc-topics) is assumed, the softmax on Φ is used to ensure that the word-topic matrix is in the softmax basis. Recall that because Dirichlet was approximated with normal, the simplex basis is also changed to a softmax basis.

The KL- divergence loss as in Section A.6 is to have the approximate posterior $q_\Phi(z|x)$ close to the true posterior $p_\theta(z|x)$. With the notation of Section 5.2.2, the approximate posterior is $q_{\mu_0,\Sigma_0}(\theta|w)$ and the true posterior is $p_\alpha(\theta|w)$, but since we approximated the prior with logistic normal, the posterior will also be approximated, so the true posterior would be $p_{\mu_1,\Sigma_1}(\theta|w)$ Note that as in Equation A.3, the KL divergence leads to minimizing the KL divergence between $q_\Phi(z|x)$ and $p_\theta(z)$, so this leads to minimizing the divergence between $q_{\mu_0,\Sigma_0}(\theta|w)$ and $p_\alpha(\theta)$. The reconstruction loss that was $\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})$ means cross entropy loss between w and

w'[7]in Section 5.2.2.

Below, you can see the equivalency of parameters and notation in Section 5.2.2 and in Section A.4: The LHS is related to Section A.4 and the RHS is related to Section 5.2.2.

$$x \rightarrow w$$

$$\theta \rightarrow \mu_1, \Sigma_1$$

$$\phi \rightarrow \mu_0, \Sigma_0$$

$$z \rightarrow \theta$$

$$p_\theta(z) \rightarrow p_\alpha(\theta)$$

$$q_\Phi(z|x) \rightarrow q_{\mu_1,\Sigma_1}(\theta|w)$$

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) \rightarrow E_{\varepsilon\sim N(0,\, I)}[w_m^T \log(\sigma(\Phi)\sigma(\mu_0 + \Sigma_0^{1/2}\varepsilon]]$$

$$-D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) \rightarrow \Sigma_{m=1}^{M} \left[- \left(\frac{1}{2} \{ tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) - K + \log \frac{|\Sigma_1|}{|\Sigma_0|} \} \right) \right]$$

[7] The cross-entropy of the distribution q relative to a distribution p over set $\mathbb{X}$
$H(p,q) = -\sum_{x\in\mathbb{X}} p(x) \log q(x)$