Correlated Topic Model with Transformer Embeddings
トランスフォーマーの埋め込みによる相関トピックモデル

by

Chun Wa Leung
梁　俊華

A Master Thesis
修士論文

Submitted to
the Graduate School of the University of Tokyo
on January 19, 2022
in Partial Fulfillment of the Requirements
for the Degree of Master of Information Science and Technology
in Computer Science

Thesis Supervisor: Akihiko Takano　高野　明彦
Professor of Computer Science

ABSTRACT

Topic modeling is one of the most common information retrieval tasks in natural language processing. In particular, Correlated topic model(CTM) is a topic model which captures the correlation between topics associated. However, such a classic statistical approach was not able to capture positional information from sequential input. At that point, traditional topic models may perform poorly in generating words from a large number of topics. In this research, we introduce Correlated Topic Model with Transformer embeddings, a generative model that combine the advantage of using the positional information of words and topic correlation. Specifically, transformer embedding maps topic words into latent space and further assigns them to its assigned topic. We attempted to add a covariance prior to the topic model, LKJ correlation prior to logistic-normal distribution, which aims to fit the correlation information from the data. In addition, we extended our model to handle time-series data integrated with the Gaussian Process latent variable model(GPLVM), which also captures temporal information from the word occurrence of documents over time. The model was optimized using Stochastic Variational Inference (SVI), which allows handling massive data sets with mini-batching. As compared to empirical results from experiments, our approach performs a better fit of the data than the existing generative topic model and exhibits a better capability in obtaining high-quality topics.

Acknowledgements

I would like to express my thanks of gratitude to my guiding professor Akihiko Takano as well as professor Takeshi Abekawa me along the master thesis. It is my pleasure to receive many valuable suggestions on thesis writing which helped me a lot in doing research.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Today, processing myriads of terabyte data are one of the crucial challenges for the scientific researcher. Information Retrieval become increasingly important for building useful information from massive data sets. Specifically, topic modeling is one of the most popular techniques for extracting key point ideas and exploring documents. Specifically, correlated topic models (CTM) make use of the correlation between topics and deliver a better result. In the research, we would like to explore the application of one of the topic modeling techniques and try to improve their performance. In section 1.1, we provide a brief introduction to the existing algorithm and cover its background. Then, section 1.2 will discuss the current application and state-of-art improvement on topic modeling advances. Following that, section 1.3 will elaborate and conduct the research and the general direction. And section 1.4 will explain the way the proposed model is to be evaluated and compared to the existing model.

1.1 Motivation

Topic modeling is one of the most exciting domains in Information Retrieval (IR). It can be extended to accomplish a versatile range of IR and data mining tasks. For instance, one of the topic models: Latent Dirichlet Allocation (LDA), was proposed and examined its capability on extracting latent topics and output keywords suggestion for each topic. As result, LDA has been implemented into various areas of applications. However, there are several problems that LDA could not handle well.

Computational complexity Generally, LDA acquires to computer the posterior distribution for inference, which is relatively expensive to obtain an exact solution. In the same way, its variant, correlated topic model (CTM) requires the calculation of the covariance matrix specifically, which makes it not feasible to come into practical application.

Statistical Laws In particular, LDA does not take empirical statistical laws observed in text into account. For example, LDA's prior does not dependent on Zipf's Law or Heap's Law, which may not collaborate well with natural document text data. Similarly, Moody[32] proposed lda2vec, which exploits the meta-information of each document and evaluates their correlation between documents.

Correlation information Correlation information can be useful to identify topics. For instance, hockey and soccer are correlated but uncorrelated to other topics like space and religion. Such intuition could help the topic model to exploit that information.

Bag-of-word assumption Typical topic models like Nonnegative Matrix Factorization (NMF) [26] and Latent Dirichlet Allocation (LDA) [9] do not consider positional information from the document set. This leads to the drawbacks of those models that may not make a good prediction on the topic words due to the limitations. For most of the NLP tasks, it is very common to let the model learning context by

Transfer Learning Due to the prevalence of deep neural networks in recent years, Transform Learning has become a hot topic in research. Transfer learning aims to make find a way to reduce computational cost and improve the re-usability of machine learning models. In the ascendant of powerful accelerators such as GPU and more memory, we can build more complex architecture and boost up computation time. Specifically, Transformer has been one of the most used NLP architecture. Several variants have been built due to its success, such as BERT[12], ROBERTA[28], and ELMO[35], etc.

1.2 Applications

Topic models are one of the crucial tasks in discovering hidden topics from document collections. The success of LDA make able it does not limit to topic modeling tasks. Graber[10] gave a verbose survey on-topic model applications. Many tasks have been applied with the model, for examples below,

Feature Extraction For number of n topics, LDA can accomplish the task cluster them and extract a set of the corpus with k terms which can represent each topic most and uniquely. Eren [15] uses LDA to analyze all literature related to COVID-19 and subdivided them into minor topics. As result, each subtopic was extracted with a set of keywords.

Text Classification Topic model can also be treated to deal with classification tasks to identify unseen data. Kim [21] adapted the semi-supervised method with multi-co-training method to improve the overall classification performance. Moreover, the paper extended Word2Vec to Doc2Vec which maintains the semantic relationship between two paragraphs. Doc2Vec transforms a paragraph into a d-dimensional vector, which puts documents with similar paragraphs into near vector space.

Recommender Systems LDA often can be applied to recommendations. Xu[46] employed UIS-LDA (A User Recommendation based on Social Connections and Interests of Users in Uni-Directional Social Networks), which utilizes Generative Polya Urn (GPU) model and perform prediction for nearest user for the recommendations. Wang [44] implemented an LDA version that utilizes the Twitter datasets and recommends a serial of tourist location to user.

Moreover, the growth of word embedding [30] enables an effective way to capture semantic meaning in language in a continuous vector space. Vocabularies that have similar meanings are close together by Euclidean distance.

1.3 Literature Review

These topic models take bag-of-words assumption and model each document as an admixture of latent topics, which are multinomial distributions over words. Nonnegative Matrix Factorization (NMF) [26] uses singular value decomposition to construct latent topic and topic-word distribution from the document, which consists of document-topic distribution matrix and topic word distribution matrix. Probabilistic Latent Semantic Analysis (PLSA) [20] is a probabilistic model that assigns every document in a single topic, and then assigns a word for every word position given the topic assignment. Similar to PLSA, Latent Dirichlet Allocation (LDA) [9] advanced PLSA from the topic assumption documents, which every document consists of admixture of topic distribution.

Some improvements exploit the correlation information between topics, which models the topic assignment with multivariate distribution to parameterize the relation between topics with mean and covariance.

Moreover, due to the success of LDA. there have been several topic models proposed on top of the LDA model. Dynamic Latent Dirichlet Allocation[5] was developed for continuous-time data. Relational Latent Dirichlet Allocation [11] exploits the tuple information from the dataset and uses it to infer the document set. Supervised Latent Dirichlet Allocation [29] includes labeled data which is supposed to be helpful in several particular application areas such as movie review and sparse data prediction. Later LDA was extended to a nonparametric version, hierarchical Latent Dirichlet Allocation (hLDA)[39], which follows a stochastic process called n-Chinese Restaurant Process (nCRP)[38]. hLDA maintains a hierarchical structure of topic instead of a flat structure in LDA.

Amortized inference[22] is common in implementing to topic models, specifically, a neural network architecture with encoder-decoder is used into the topic model structure for model inference. Srivastava[37] applied amortized variational inference to approximate the variational distribution of the model. Specifically, the products of experts were used to collapse out the document-topic assignment parameter and simplify the inference process.

Some other attempts use graph techniques to model topic distributions. Yang[48] introduced a new topic model with Graph neural network techniques. The paper introduced Graph Attention TOpic Network (GATON) which hybridized the graph attention network (GAT) and amortized inference into the application of topic modeling which is supposed to reduce the required computation complexity. In past research, some considered n-gram to model the word pattern under a sentence structure which results in a better prediction. Wallach [43] proposed a topic model using bi-gram information from the data set to yield better performance in topic interpretability. Wang [45] extends the topic model to the n-gram assumption with a similar approach.

1.4 Objective and Outline

In the thesis, we would like to construct a topic model that makes use of the positional information we obtain from the document set. Moreover, we would exploit the usage for imposing a prior to model the covariance on the document-topic proportion. Particularly, we develop the Transformer Embedding Correlated Topic Model(TECTM), a model that combines word embedding and topic model together to make a better fit of the dataset. Moreover, we integrate the Transformer into embedding, such that we can also take the assumption of word position and convert it into meaningful contextual embeddings. In its generative process, the model uses the topic embedding to form a per-topic distribution over the vocabulary. Specifically, the TECTM uses a log-linear model that takes the inner product of the word embedding matrix and the topic embedding. With this form, the TECTM assigns a high probability to a word v in topic k by measuring the agreement between the word ' s embedding and the topic ' s embedding. To evaluate our model, we applied the proposed model on the *20Newsgroups* and *Reuter-21578* dataset. The experiment results demonstrate that our model is capable to obtain high-quality topics than the state-of-the-art model. In chapter 4, we also extended the model to handle time-series information, the model extends the architecture based on chapter 3. Specifically, we built Dynamic Transformer Embedding Correlated Topic Model(DTECTM), a model on top of the one from chapter 3, that makes use of the Gaussian Process Latent Variable Model(GPLVM) to capture time-series information. We put our model into a set of experiments with other instances to examine the effectiveness. The models are compared with *the NIPS* dataset and *UN debates* dataset, which each of them consists of a time label that represents the year a specific document belongs to. Additionally, we also visualize the time-to-topic proportion that the model obtained to explore the topics that evolve over time. The result shows that To give an outline of this thesis, chapter 2 will give a brief background to the problem description and the related knowledge, including the existing topic models, the related methodology in , and the evaluation metric in NLP domain such as perplexity, and topic coherence and topic diversity, which is specifically for topic model evaluations. Then chapter 3 will specify the methodology and explain the detail of our model, we compare the proposed model with LDA and the latest model. In 4, we explore the DTECTM model to train with time-series data. We will specify the model and the Lastly, chapter 5 will sum up the merit and limitation overall the research, and prospect the future works.

Chapter 2

Background

In this section, we give a short description of topic model techniques and several key components and terminologies related to the LDA model. In section 2.1, we formulate the problem description of topic modeling. In 2.5.1, we give a description of word embedding. In section 2.4.1, we explain the algorithm of the Correlated topic model(CTM). Then, section 2.3 will give the definition for LKJ correlation distribution. Section 2.4.2 will cover the Embedded Topic Model(ETM). Finally, section 2.5.2 will give introduction to transformer embedding that we are used in the model. Several topic modeling techniques include non-negative matrix factorization (NMF) [26], Latent Semantic Analysis [23], probabilistic Latent Semantic Analysis (pLSA) [20] and Latent Dirichlet Allocation (LDA)[9].

2.1 Problem definition

Massive data sets in internet has made the task of understanding data by accessing them one-by-one became not humanly possible. The raise of topic modeling gives possibility to summarize a given set of document collections. To describe topic modeling intuitively, for a document collection $d = \{1 \cdots D\}$ and define a number of topic K, the model outputs topic-word distribution $\beta \in \mathbb{R}^{K \times V}$, where K is number of topics and V is number of vocabularies in document set. For $\{\beta_{k,1}, \beta_{k,2}, \cdots, \beta_{k,V}\}_{k=1}^{K}$, each $\{\beta_{k,v}\}_{k=1}^{K}$ resulting the expression power vocabulary v could represent in topic k. The higher value a vocabulary obtained in tuple $\{\beta_{k,v}\}_{v=1}^{V}$, the high representation power that the word is related to a latent topic. Practically, we capture top-m words from topic model for each topic k. For a set of top-m words obtained from topic k in descending order, defined as $\{\beta_{k,v_m}\}_{m=1}^{M}$ where $\beta_{k,v_1} \succeq \cdots \succeq \beta_{k,v_m} \succeq \cdots \succeq \beta_{k,v_M}$.

2.2 Bag-of-word assumption

Suppose we have a collection of documents $\in \mathbb{R}$. The document and vocabularies Specifically, Topic model is a generative model that the probability based on Bag-of-words(BOWs) assumptions. Bag-of-words(BOWs) is a assumption that all words in the document are considered are independently distributed. To represent BOW, let a document collection $W = (w_1, w_2, \cdots, w_D), d \in \{1 \dots D\}$ documents, where w_d is a single document d contain words $w_d = (w_{d1}, w_{d2}, \cdots, w_{dN_d}), n \in \{1 \dots N_d\}$ word position, where w_{dn} is a single word in document d at position

n. In equation 2.1, we take unigram model as an example[49],

$$p(W|\phi) = \prod_{d=1}^{D} p(w_d|\phi) = \prod_{d=1}^{D}\prod_{n=1}^{N_d} p(w_{dn}|\phi) = \prod_{d=1}^{D}\prod_{n=1}^{N_d} \phi_{w_{dn}} = \prod_{v=1}^{V} \phi_v^{N_v} \tag{2.1}$$

For sake of convenience, documents contain different sizes of words. It is not feasible to construct a matrix representation for modeling. For this reason, we can exchange the representation for BOW from iterating the word occurrence for each word position to iterating the word occurrence in a vocabulary set. In this way, we can define a matrix for BOW: $W \in \mathbb{R}^{D\times V}$, with D rows of document and V rows of vocabulary count. Formally, for every document $W = (w_{d1}, w_{d2}, \cdots, w_{dV}), v \in \{1, \cdots, V\}$ vocabularies, w_{dv} is the occurrence of a vocabulary v in document d.

2.3 LKJ Correlation Distribution

LKJ distribution [27] is a distribution for modeling the correlation matrix. The distribution is described as equation 2.2[1]

$$f(C|\eta) = 2^{\sum_{k=1}^{K-1}(2(\eta-1)+K-k)(K-k)} \times \tag{2.2}$$

$$\prod_{k=1}^{K-1} (B(\eta + (K-k-1)/2, \eta + (K-k-1)/2)^{K-k})(\det(C)^{\eta-1}) \tag{2.3}$$

$B(\cdot,\cdot)$ is beta distribution, and K is the number of variables in the correlation matrix. η is the concentration parameter for LKJ distribution. When $\eta = 1$ it is a uniform distribution allocated over the correlation matrix. If $\eta > 0$ and become larger, the density of the matrix concentrates around the center. To apply it into normal distribution as covariance, we could apply transformation equation 2.4 and turn it into covariance matrix[2]. diag(σ) is the diagonal elements of the variance vector σ.

$$\Sigma = \mathrm{diag}(\sigma) \cdot C \cdot \mathrm{diag}(\sigma) \tag{2.4}$$

Directly drawing correlation matrix from LKJ distribution is not efficient in reality case. It is common to draw a correlation matrix from factorized Cholesky LKJ distribution instead, where the probability density function is described as equation 2.5

$$\mathrm{LKJChol}(L|\eta) \propto |J| \det(LL^\top)^{(\eta-1)} \tag{2.5}$$

$$= \prod_{k=2}^{K} L_{kk}^{K-k+2\eta-2} \tag{2.6}$$

The lower triangular matrix L is a Cholesky factorization for the correlation matrix iff $L_{k,k} > 0$

$$\Sigma = \mathrm{diag}(\sigma) \cdot LL^\top \cdot \mathrm{diag}(\sigma) \tag{2.7}$$

similarly, the transformation from LKJ Cholesky matrix to covariance matrix as an equation 2.4.

[1] https://distribution-explorer.github.io/multivariate_continuous/lkj.html

2.4 Topic Models

2.4.1 Correlated Topic Model

Correlated Topic Model (CTM)[8] is an extension of LDA[9] that utilizes the correlation of latent topics and relates the similar documents together. Instead of Dirichlet distribution, CTM applies multivariate logistic-normal distribution to model the word distribution.

The model contain K topics distribution as $\beta_{1:K}$, $z_{n,d}$ is the topic assigned to the n-th topic and d-th document.θ_d is the corresponding proportion a topic is distributed to d-th document.μ and Σ are the corresponding mean and $K \times K$ covariance matrix of the distribution between documents.

Algorithm 1: Generative Process for CTM

Initialize μ, Σ
for *document d in D* **do**
  Sample a topic distribution $\eta_d \sim \mathcal{N}(\mu, \Sigma)$
  for *word position n in* N_d **do**
    Sample a topic assignment $z_{dn} \sim \mathrm{Mult}(f(\eta_d))$
    Sample a word $w_{dn} \sim \mathrm{Mult}(\beta_{z_d,n})$
  end
end

From Algorithm 8, the parametrization μ, Σ are initialized. For each document d in document collection D, a topic distribution η_d is drawn from normal distribution parametrized μ, Σ. Then for each word position n in N words in doucment d, a topic assignment z_{dn} is drawn from multinomial distribution parametrized $f(\eta_d)$, where the transformation $f(\eta)$ represents the softmax function maps the sample draw from normal distribution to topic proportion θ, a topic distribution which points on the simplex that all elements in the vector sums to 1. Finally, a word is sampled from multinomial distribution $Mult(\beta_{z_d,n})$. From the figure 2.1 we can see, word and topic-word assignments are in the word

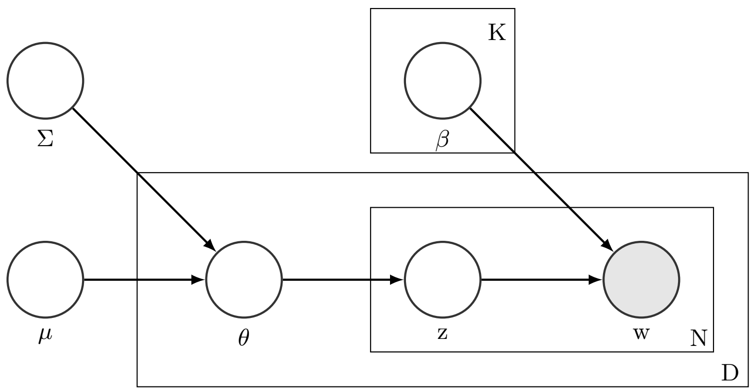

Figure 2.1: Graphical representation for CTM

and document plate $N \times D$. The document topic proportion η is on document plate D. Specifically, the topic word proportion β is on the topic plate K, which is specified as word distribution selected by topic assignment z.

Mathematical Formulation The joint distribution for CTM is described as follows,

$$p(\eta, z, w|\beta, \mu, \Sigma) = \prod_{d=1}^{D} p(\eta_d|\mu, \Sigma) \prod_{n=1}^{N_d} p(z_{dn}|\eta_d) p(w_{dn}|z_{dn}, \beta_{1:K})$$

and the ELBO is defined as,

$$\mathcal{L} \geq \sum_{d=1}^{D} \mathbb{E}_{q_d} [\log p(\eta_d, z_d, w_d|\mu, \Sigma, \beta_{1:K})] - \sum_{i=1}^{K} \log \mathrm{KL}(q(\eta_i|\lambda_i, \nu_i^2)|p(\eta_d|\mu, \Sigma))$$
$$- \sum_{n=1}^{N} \log \mathrm{KL}(q(z_n|\phi_n)||p(z_n|\eta_d))$$

2.4.2 Embedded Topic Model

The embedded Topic Model [14] is one of the state-of-art approaches for the topic model task. It takes word distribution β as a topic embedding for words. Similar to Word2Vec[30], the word distribution is a softmax function of the inner product of context matrix ρ and context embedding α. Specifically, the algorithm equation 2.8

$$\beta \sim \sigma(\rho^\top \alpha) \tag{2.8}$$

The word is drawn from the generative process shown in algorithm 2, for each document, sample a topic distribution θ from logistic-normal distribution parameterized with zero mean and identity covariance. Then for each word position n, the model sample topic assignment z_{dn} from categorical distribution θ_d. Finally, a word is drawn from softmax($\rho^\top \alpha$) on z_{dn} the row.

Algorithm 2: Generative Process for ETM

```
foreach document d ∈ 1 . . . D do
    Draw document topic distribution θ_d ∼ LN(0, I)
    foreach word position n ∈ 1 . . . N_d do
        Draw topic assignment z_{d,n} ∼ Cat(θ_d)
        Draw word w_{d,n} ∼ σ(ρ^⊤ α)_{z_dn}
    end
end
```

2.5 Representation learning

2.5.1 Word Embedding

Word embedding[3] is a kind of representation for words from document collections using a vector formulation. The nature of word embedding is that the words that have similar meanings have a close distance(in most cases euclidean distance), and vice versa. For instance, continuous bag-of-words(CBOW) [30] is a kind of word embeddings converting bag-of-words into a vector of n-dimension continuous space, which contains the following formulation,

$$w \sim \mathrm{softmax}(\rho^\top \alpha)$$

where $\rho \in \mathbb{R}^{L \times V}$ is the embedding matrix which a function $f : \mathbb{R}^V \mapsto \mathbb{R}^L$ maps V vocabularies into L dimension of continuous vector space. And α is the context embedding, which conveniently convert the latent dimension L to a custom dimension of continuous embedding space $\mathbb{R}^N$ as $\tilde{f} : \mathbb{R}^L \mapsto \mathbb{R}^N$.

2.5.2 Transformer

The transformer[42] is a popular neural network architecture in natural language processing. To briefly explain transformer, it is a stacked encoder-decoder architecture. The component that makes the transformer stand out from other architectures is the multi-head self-attention mechanism.

In this section we only cover the main components of transformer. The details for transformer can be reviewed in the author's blog post[2].

To define Transformer, equation 2.9 denotes the scaled dot product that transformer use to calculate attention score. $Q \in \mathbb{R}^{T \times d_k}$, $K \in \mathbb{R}^{T \times d_k}$ and $V \in \mathbb{R}^{T \times d_v}$ are the query, key, value term vector used to calculate the context vector. S represents the sequence length and d_k and d_v the dimension for the key and value respectively. $\frac{1}{\sqrt{d_k}}$ is used to scale down attention matrix in which to maintain a proper variance for the attention scores.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.9}$$

Weight matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$ and $W_i^V \in \mathbb{R}^{d_{model} \times d_k}$ for query, key and value are parameters to be learned. The scaled dot-product attention computes a sequence of vector outputs. $W^O \in \mathbb{R}^{hd_v \times d_{model}}$, where d_{model} is the dimension of key-value pair multiplies number of head.

$$\text{Multihead(Q,K,V)} = \text{concat}(h_1, \cdots, h_h)W^O \tag{2.10}$$

$$h_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.11}$$

The sub-layer multi-head self-attention is connected to the fully-connected feed-forward network. The outputs of each sub-layer are added to the original output with residual connection applying Layer Normalization is eq. 2.12.

$$x = \text{LayerNorm}(x + \text{Sublayers}(x)) \tag{2.12}$$

Positional Encoding

One drawback for Multi-Head Attention block is that it does not consider information about word positioning. The positional encoding function maps the sentence sequences into i-dimension of hidden space fo each permutation position *pos*. And so the model can identify the from the additional positional features space of the input. [3]

$$PE_{(pos,i)} = \begin{cases} \sin\left(\frac{pos}{10000^{i/d_{model}}}\right) & \text{if } i \mod 2 = 0 \\ \cos\left(\frac{pos}{10000^{(i-1)/d_{model}}}\right) & \text{otherwise} \end{cases}$$

the positional encoding function $PE_{pos,i}$, where the *pos* represents the permutation and the odd/even dimension are treated on sine/cosine function respectively.

2.6 Time Series model

2.6.1 Gaussian Process (GP)

Gaussian Process[36] is a versatile algorithm that is commonly used for both supervised and unsupervised learning problems including regression, classification,

[2] `https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html`

[3] Description from `https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial6/Transformers_and_MHAttention.html`.

and clustering. It has been demonstrated a strong pursuit on times-series and reinforcement learning problems. A Gaussian Process $\mathcal{GP}(\cdot,\cdot)$ is denoted by the following, where m is the mean and k stands for a kernel for covariance function,

$$f \sim \mathcal{GP}(m, k)$$

For a set of input $x_1, \ldots, x_N$, the joint probability density function $p(f(x_1), \ldots, f(x_N))$ is a normal distribution condition on mean vector $m(X)$ and covariance matrix $k(x, x')$ which is a positive semidefinite matrix. In most case the mean vector are specified with zero mean (m(X)=0).

$$p(f(X)) = \mathcal{N}(m(X), k(X, X))$$

In particular, in Gaussian Process regression task, a output variable y_i denoted $y_i \sim f(x_i) + \epsilon$ where ϵ is a Gaussian noise. Therefore, given a set of training input and target,

$$p(y|f) = \mathcal{N}(y; f, \sigma_\epsilon^2 I)$$

Radial Basis Function kernel (namely RBF kernel) is a common function that used as a kernel in GP model. For a kernel $k(x, x')$,

$$k(x, x') = \sigma^2 \exp\left(-\frac{||x - x'||^2}{2l^2}\right)$$

where l is the kernel bandwidth. The ELBO for GP is derived from [18], as

$$\log p(y) \geq \mathbb{E}_{q(f)}[\log p(y|f)] - KL[q(u)||p(u)]$$

where the variational distribution is $q(f) \approx \int p(f|u)q(u)du$

2.6.2 Gaussian Process Latent Variable Model (GPLVM)

Gaussian Process Latent Variable Model [24, 40] is a dimension reduction method that turns high dimensional data into low-dimension space. Formally, given a latent variable $X \in \mathbb{R}^{N\times Q}$, and high dimensional real-valued observations $Y \in \mathbb{R}^{N\times D}$, the model induce high dimensional from a low-dimensional mapping $\mathcal{X} \rightarrow \mathcal{Y}$ such that $Q << D$. The prior latent variable X is defined at a standard normal distribution in dimension Q.

$$p(X) = \prod_{n=1}^{N} \mathcal{N}(x_n; 0, I_Q)$$

The latent function f take the kernel K_d that determine the covariance matrix at D-dimension

$$p(f|X, \theta) = \prod_{d=1}^{D} \mathcal{GP}(f_d; 0, K_d)$$

and then it can recover back the observed data through the latent function f conditioned on the $\mathcal{GP}$ prior with $p(Y|f, X)$

$$p(Y|f, X) = \prod_{i=1}^{N} \prod_{i=1}^{D} \mathcal{N}(y_{n,d}; f_d(x_n), \sigma_y^2)$$

Here we discuss the variational inference method for the inference process. The variational distribution for GPLVM is given as eq. 2.13 according to GP sparse approximation in [25]. In the formulation, x is the latent variable, f_d is the latent function for the covariance where variational distribution for $q(X) = \sum_{n=1}^{N} \mathcal{N}(x_n; m_n, S_n)$ for each n^{th} row on x, and $q(u_d) = \mathcal{N}(u_d|0, K_{MM})$

$$q(\{f_d, u_d\}_{d=1}^{D}, X) = \prod_{i=1}^{N} q(x_n) \prod_{d=1}^{D} p(f_d|u_d, X)q(u_d) \tag{2.13}$$

The latent variable ELBO is given by,

$$\mathcal{L} = \sum_{n,d} \mathbb{E}_{q_\phi(x_n)} \mathbb{E}_{p(f_d|u_d,x_n)q_\lambda(u_d)} \left[\log \mathcal{N}(y_{n,d}; f_d(x_n), \sigma_y^2)\right] \tag{2.14}$$

$$- \sum_{n} \mathrm{KL}(q_\phi(x_n)||p(x_n)) - \sum_{d} \mathrm{KL}(q_\lambda(u_d)||p(u_d|Z)) \tag{2.15}$$

the ϕ is the local variational parameters, λ is the global variational parameters, θ is the kernel hyperparameters and σ^y is the likelihood noise. to avoid heaving computation of X inside the conditional probability $p(f_d|X)$, inducing input has been introduced [25] to replace the X with inducing variables $u_d \in \mathbb{R}^M$, which conditioning on inducing input locations $z \in \mathbb{R}^{M \times Q}$,

2.6.3 Dynamic Topic Model

Dynamic Topic Model [7] is a model that enables to capture of topic information from time-series data set. The model utilizes documents from a different time and generates the corresponding topic-word representation.

The generative process of the d^{th} document is the following:

Algorithm 3: Generative Process for DTM

Sample topics $\beta^{(t)} \sim \mathcal{N}(\beta^{(t-1)}, \sigma^2 I)$
Sample topic proportion mean $\eta_t \sim \mathcal{N}(\eta_{t-1}, \delta^2 I)$
for *document d in D* **do**
    Sample $\theta_d \sim \mathcal{LN}(\eta_{t_d}, \alpha^2 I)$
    for *word position n in* N_d **do**
        Sample topic $z_{d,n} \sim \mathrm{Mult}(\theta_d)$
        Sample word $w_{d,n} \sim \mathrm{Cat}(\beta_{z_{dn}}^{t_d})$
    end
end

The model assumes the topic-word proportion and they are in a state-space model that moves along the time $1 \ldots T$ with their corresponding variance σ and δ. η_t is the latent variable model the prior mean to the topic proportion along the timeline. α_k^{t-1} is the mean of current time t which take the value previous time step $t-1$. The original DTM model approximates the prior by deriving the variational lower bound with the Kalman Filter method. γ^2 and ξ^2 are the variances for the prior.

2.7 Posterior Inference

Since the exact inference of the posterior is intractable in a real application, we employed an approximation scheme for the posterior inference. The popular approaches are Markov Chain Monte Carlo Method (MCMC) and Variational

Inference(VI)[6, 19]. Gibbs sampling is one of the MCMC methods and it is fast to compute the approximation and easy to implement. Then, a Variational EM algorithm is to be carried out for maximizing the likelihood of the overall word in the corpus in the document. An alternative way to perform estimation is the Monte Carlo method.

2.7.1 Variational Inference

Given that posterior approximation is not always practical in a real-world application. Approximation methods are necessary to be applied. There are two main approaches for the posterior approximation: Markov Chain Monte Carlo(MCMC) Variational Inference. Variational Inference is a method approximating the posterior in an optimization fashion. To give a better intuition, let probability $p(x)$ depending on a latent variable z such that $p(x|z) = \int p(x|z)p(z)dz$. We can turn the following posterior inference problem into an optimization problem. Here we derive the bound and perform optimization, by doing some calculation, we can obtain the following lower bound for the likelihood $\mathcal{L}_i(p, q_i)$

$$\log p(x_i) \geq \mathbb{E}_{z\sim q_i(z))}[\log p(x_i|z) + \log p(z)] + \mathbb{E}_{z\sim q_i(z))}[\log q_i(z)] = \mathcal{L}_i(p, q_i) \tag{2.16}$$

then by deriving the KL-divergence, we can obtain the log probability $\log p(x_i)$ is the likelihood-term minus the KL-divergence of $q_i(z)$ and $p(z|x_i)$.

$$KL(q_i(z)||p(z|x_i)) = -\mathcal{L}_i(p, q_i) + \log p(x_i) \tag{2.17}$$

$$\mathcal{L}_i(p, q_i) = \log p(x_i) - KL(q_i(z)||p(z|x_i)) \tag{2.18}$$

with this derivation, we turn a posterior approximation problem into an optimization problem by maximizing the evidence lower bound(ELBO).

2.7.2 Stochastic Variational Inference

Stochastic Variational Inference (SVI)[19] is a scalable variant of variational inference, which enables mini-batching to split dataset and train for each epoch, then become a standard of optimization for probabilistic models. Two main improvements are made by the SVI: stochastic optimization and noisy gradient.

2.7.3 Collapsing Parameters

In the original LDA model, the parameter z is responsible for sampling topic assignment for each word position in every single document. Collapsing parameters[37] were introduced to reduce the latent variable z in the generative process in hence speeding up computation.

$$w_d \sim \prod_{n=1}^{N_d} \mathrm{Cat}(\sigma(\theta_d^\top \beta)) \tag{2.19}$$

The trick in equation refeq:cp rewrite the original LDA word drawing process, and hence define a piece of new evidence lower bound for the topic model.

2.7.4 Autoencoding Variational Bayes (AEVB)

Generally, when we optimize a variational parameter, it is necessary to derive an ELBO and then derive the optimization step for gradient descent. While amortized inference latent variable z is parameterized by two inference network $\mu_{\phi(x)}, \sigma_\phi(x)$.

$$z = \mathcal{N}(\mu_\phi(x_i), \sigma_\phi(x_i)) \tag{2.20}$$

After deriving the ELBO, we obtain the following likelihood term.

$$\mathcal{L} = \mathbb{E}_{z\sim\mathcal{N}(\mu_\phi(x_i),\sigma_\phi(x_i))} [\log p_\theta(x_i|z)] - KL\left(q_\phi(z|x_i)||p(z)\right) \tag{2.21}$$

2.7.5 Reparameterization trick

The drawback of amortized inference is that, sampling from normal distribution parameterizing $\mu_{\phi(x),\sigma_\phi(x)}$ could lead to high variance outcome and hamper the inference performance. For the reason, taking reparamterization trick[22] to transform as equation 2.22,

$$z = \mu_\phi(x_i) + \epsilon\sigma_\phi(x_i),\, \epsilon \sim \mathcal{N}(0, 1) \tag{2.22}$$

where ϵ is a sample from normal distribution $\mathcal{N}(0, 1)$. and so the modified ELBO becomes equation 2.23,

$$\mathcal{L} = \mathbb{E}_{\epsilon\sim N(0,1)} [\log p_\theta(x_i|\mu_\phi(x_i) + \epsilon\sigma_\phi(x_i))] - KL\left(q_\phi(z|x_i)||p(z)\right) \tag{2.23}$$

2.8 Evaluation metrics

2.8.1 Perplexity

The proposed model will be evaluated with a perplexity metric. The metric will examine how well the model can tackle unseen data. It is equivalent algebraically to the inverse of the geometric mean per-word likelihood. Lower perplexity scores mean better.

$$\text{Perplexity}(D_{test}) = \exp -\frac{\sum_{d=1}^{M}\sum_{m=1}^{N_d}\log p(w_{dm})}{\sum_{d=1}^{M} N_d} \tag{2.24}$$

2.8.2 Topic Coherence

Topic Coherence[31] measures the quality of the topic

$$TC = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{45}\sum_{i=1}^{10}\sum_{j=i+1}^{10} f(w_i^{(k)}, w_j^{(k)}) \tag{2.25}$$

where $\{w_1^{(k)}, \cdots, w_{10}^{(k)}\}$ denotes top-10 most likely words in topic k. And function $f(\cdot,\cdot)$ is te normalized point-wise mutual information.

$$f(w_i, w_j) = \frac{\log\frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)} \tag{2.26}$$

2.8.3 Topic Diversity

To compare how the words of each topic are differentiated from the others. We applied the Topic Diversity metric [14]. Topic Diversity (TD) is the percentage of unique words in the top 25 words of all topics. Diversity close to 0 indicates redundant topics; diversity close to 1 indicates more varied topics. We define the overall metric for the quality of a model ’ s topics as the product of its topic diversity and topic coherence.

$$TD = \frac{|A \cap B|}{|A \cup B|} \tag{2.27}$$

where A and B are top-k words from two topics.

Chapter 3

Transformer embedding with Correlated Topic Model

In this chapter, we give a detailed explanation and procedure of Transformer embedding with Correlated Topic Model(TECTM) to be implemented.

3.1 Introduction

Latent Dirichlet allocation(LDA)[9] is one of the popular models in topic modeling. However, the model takes the bag-of-words assumption that all words are independently distributed. Also, the original model takes an optimization step on the entire set of documents, where it limits the size of the data set that can be trained within a limited memory size and hence scalability is a concern for LDA. The correlated Topic Model(CTM)[8] takes consideration between topics by implementing the correlation information over the document-topic proportion. However, the model does not make assumptions of prior information on the covariance matrix. Embedded Topic Model(ETM)[14] explores the possibilities the word2vec embedding to be working together with a topic model to improve the quality of topic-words generation. However, since word2vec is a simple embedding architecture, the model could be better to be working with embedding that captures positional information.

In this chapter, we propose Transformer embedding with Correlated Topic Model(TECTM), a topic model that takes prior assumption on covariance matrix over the document-topic distribution and integrates Transformer with the topic model to maintain a better quality of word representations in latent space. We propose LKJ correlation prior to our correlated topic model, where the correlation prior takes place to capture the correlation between topics by modeling the proportion over document-topic. To take advantage of positional information, we access the possibility of the use of the Transformer model to improve topic model performance. The transformer takes input sequence from documents and performs scaled dot-product to compute that each token in a sentence relates to each other and the importance over a hidden context. In the following chapter, we first go through the related works that have been proposed by other authors. Then, we define the proposed model TECTM and derive its inference process. After that, we explain the implementation detail and the algorithmic setting for the experiment. Finally, the results are compared with other state-of-the-art models and discuss the performance of our model to outperform the other instances.

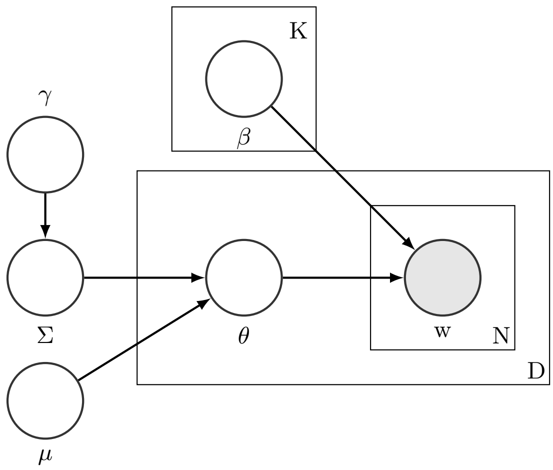

Figure 3.1: Graphical model for TECTM

3.2 Related works

Correlated Topic Model (CTM)[8] is the original work that proposed to alleviate the problem LDA, which did not utilize the topic information between correlated topics. The proposed model replaced Dirichlet distribution with a logistic-normal prior with covariance matrix to represent the relationship between topics.

There have been several works focusing on word embedding and topic models. Major of them combined statistical model and embedding approach to model topic distribution. In other words, these methods represent a word by mapping every single word into continuous space instead of using a probability distribution as was in the typical LDA model. The embedded Topic Model[14] uses word2vec embedding to capture the word representation in latent continuous space. The posterior of the model was approximated by amortized inference. Xun [47] employed word embedding into Correlated Topic Model, the new correlated topic model as Correlated Gaussian Topic Model (CGTM). In their paper, they make use of word embedding space and model the correlation between topics by calculation of similarity between words in the embedding space. Similarly, He[16] proposed Correlated Topic Modeling with Topic Embedding (CTMTE), which transformed the topic distribution previously obtained into lower dimension topic embedding space. The correlation between topics was directly computed through the similarity calculation in the vector space. The paper stated it reduces the running time as a scalable framework into large applications.

3.3 Model description

The TECTM utilizes the Transformer as embedding to the topic-word representations. To compare with the original topic model, the word-topic distribution β is the First, the topic embedding embeds the vocabulary into L-dimensional space, which is by Transformer embedding. Second, the context embedding maps the embedding into K-dimensional space. In the generative process, the TECTM uses the topic embedding to form a per-topic vector to represent the meaning over the vocabulary. The generative process of the d^{th} document is the following:

Algorithm 4: Generative Process for TECTM

1 Initialize hyperparameters γ, μ
2 Sample Cholesky factor $L \sim \text{LKJChol}(\gamma)$
3 **for** *document d in D* **do**
4   Sample topic distribution $\theta_d \sim \mathcal{LN}(0, LL^\top)$
5   **for** *word position n in* N_d **do**
6   Sample topic assignment $z_{dn} \sim \text{Cat}(\theta_d)$
7   Sample word $w_{d,n} \sim \text{Cat}(\sigma(\rho^\top \alpha)_{z_{dn}})$
8   **end**
9 **end**

From the algorithm 4, starting from step 1, the topic proportion θ_d is drawn from the logistic-normal distribution $\mathcal{LN}(\cdot)$ with zero mean and identical covariance. From Step 2-a, for each word position n in document d, a topic assignment to word w_{dn} is drawn from categorical distribution $Cat(\theta_d)$ parameterized by topic proportion θ_d Step 2-b, the model draws a word from the embedding of the vocabulary ρ and the assigned topic embedding $\alpha_{z_{dn}}$ to draw the observed word from the assigned topic, as given by z_{dn}. The embedding is applied softmax function to make them topic distribution. The TECTM likelihood uses a matrix of word embedding ρ, a representation of the vocabulary in a lower-dimensional space. In practice, it can either rely on previously fitted embeddings as part of the fitting procedure, it simultaneously find topics and an embedding space.

3.3.1 LKJ Correlation prior

The LKJ Correlation prior LKJChol draws a Cholesky factor of lower triangle matrix L, from a decomposed LKJ correlation distribution. The product of lower triangular matrix $LL^\top$ reconstruct the correlation matrix

$$\Sigma = LL^\top$$

A covariance matrix can be reconstructed in the following fashion. In our implementation, we don't scale the covariance matrix with variance σ, which is simply a correlation matrix.

3.3.2 Transformer Embeddings

Following the ETM architecture, we modify topic-word distribution as an embedding and put transformer embedding to work into it.

$$\beta_k = \text{softmax}(\rho^\top \alpha_k) \tag{3.1}$$

Equation 3.1, the topic-word distribution is composed of the dot product of transformer embedding $\rho \in \mathbb{R}^{L\times V}$ representing the word vector in L-dimension of continuous space and topic matrix $\alpha \in \mathbb{R}^{L\times K}$ mapping the L dimension vector into K-dimension of topic proportions.

3.3.3 Marginal Likelihood

To compute the parameters of the model, we first compute the log-marginal likelihood. In equation 3.2, the marginal likelihood is parameterized by transformer embedding ρ and topic embedding α, which is for constructing topic-word proportion β as referenced in equation 3.1,

$$\mathcal{L}(\rho, \alpha, \Sigma) = \sum_{d=1}^{D} \log p(w_d|\rho, \alpha, \Sigma) \tag{3.2}$$

the marginal probability for $p(w|\alpha)$ on d-th document,

$$p(w_d|, \Sigma, \rho, \alpha) = \int_\theta \prod_{n=1}^{N} \sum_{z_n=1}^{K} p(w_n|z_n, \rho, \alpha)p(z_n|\theta)p(\theta|\Sigma)d\theta \tag{3.3}$$

the conditional distribution $p(w_{dn}|\theta_d, \alpha)$ marginalize out the topic assignment z_{dn} by collapsing parameters transformation $w \sim \theta^\top \beta$,

$$p(w_{dn}|\theta_d, \rho, \alpha) = \text{Cat}\left(\sigma(\theta_d \beta)\right) \tag{3.4}$$

as always, computing integrals for posterior is intractable, approximate inference is necessary to estimate the true parameter from the integral.

3.3.4 Joint Distribution

We give a description of the joint distribution, W, Z, θ and Σ are variables and α, μ and γ are latent variables. W is the word likelihood from the document collections, Z represents the topic-word assignment, θ models the topic distribution for each document, and Σ is the covariance matrix which depends on the document-topic distribution θ.

$$\begin{aligned} p(W, Z, \theta, \Sigma|\alpha, \gamma) &= p(W|Z)p(Z|\theta)p(\theta|\Sigma) \\ &= p(\Sigma|\gamma) \prod_{d=1}^{D} p(\theta_d|\Sigma) \prod_{n=1}^{V} p(z_{d,n}|\theta_d)p(w_{d,n}|z_{d,n}, \alpha) \end{aligned}$$

by taking a log on the joint probability, we obtain an objective function for optimization

$$\begin{aligned} \log p(W, Z, \theta, \Sigma|\alpha, \mu, \gamma) = &\sum_{d=1}^{D} \left[\log p(\theta_d|\Sigma) + \sum_{n=1}^{V} [\log p(z_{d,n}|\theta_d) + \log p(w_{d,n}|z_{d,n}, \alpha)] \right] \\ &+ \log p(\Sigma) \end{aligned}$$

3.4 Inference and Estimation

To perform posterior inference, we apply variational inference to transform the log-likelihood function into a lower bounded optimization problem.

3.4.1 Variational distribution

The variational distribution for $q(\theta_d)$ is a multivariate normal distribution in $\mathbb{R}^D$ that parameterized by mean vector $\mu(w_d)$, a inference network take input from bag-of-words w_d and then outputs means from its weight parameters; Σ is a conditional variational parameter from $q(\Sigma)$.

$$q(\theta_d|\Sigma) = \mathcal{N}(\mu_\varphi(w_d), \Sigma)$$

3.4.2 Evidence Lower Bound(ELBO)

To perform Variational inference, it is essential to derive the Evidence Lower Bound (ELBO) first as the objective function for the optimization. By equation

3.5

$$
\begin{aligned}
\mathcal{L} \geq & \mathbb{E}_q[\log p(W, Z, \theta, \Sigma)] - \mathbb{E}_q[\log q(Z, \theta, \Sigma)] \\
= & \sum_{d=1}^{D}\sum_{n=1}^{V} \mathbb{E}_q[\log p(w_{d,n}|z_{d,n}, \alpha)] + \sum_{d=1}^{D}\sum_{n=1}^{V} \mathbb{E}_q[\log p(z_{d,n}|\theta_d)] \\
& + \sum_{d=1}^{D} \mathbb{E}[\log p(\theta_d|\mu, \Sigma)] + \mathbb{E}_q[\log p(\Sigma)] - \sum_{d=1}^{D}\sum_{n=1}^{V} \mathbb{E}_q[\log q(z_{d,n}|\alpha_{d,n})] \\
& - \sum_{d=1}^{D} \mathbb{E}_q[\log q(\theta_d|\lambda_d, \nu_d)] - \mathbb{E}_q[\log q(\Sigma|\gamma)]
\end{aligned} \tag{3.5}
$$

Collapsing Parameters we can speed up the computation by marginalize out the z. The probability for word w could be simplified as $w_{dn} \sim \mathrm{Cat}(\sigma(\theta_d^\top \beta))$.

$$
\begin{aligned}
\mathcal{L} &\geq \sum_{d=1}^{D} \int\int q(\theta_d)q(\Sigma) \log \frac{p(W_d|\theta_d, \alpha)p(\theta_d|\mu, \Sigma)p(\Sigma|\gamma)}{q(\theta_d)q(\Sigma)} d\theta_d d\Sigma \\
&= \sum_{d=1}^{D} \left(\mathbb{E}_{q(\theta_d)}\left[\log p(W_d|\theta_d, \alpha)\right] - KL(q(\theta_d)||p(\theta_d|\mu, \Sigma))\right) - KL(q(\Sigma|\gamma)||p(\Sigma))
\end{aligned} \tag{3.6}
$$

Here we define amortized inference [22], an optimization technique that performs inference by defined neural networks. $\mu_\theta(w)$ is an inference network that takes inputs from the bag-of-words vector w_d. Then a output is generated by the normal distribution parameterized by mean vector generated by $\mu_\theta(w)$ and covariance Σ.

$$
\begin{aligned}
= & \sum_{d=1}^{D} \left(\mathbb{E}_{\theta_d \sim \mathcal{N}(\mu_{\theta_d}(w_d), q(\Sigma))}\left[\log p(w_d|\theta_d, \beta)\right] - KL(q(\theta_d)||p(\theta_d|\mu, \Sigma))\right) \\
& - KL(q(\Sigma|\gamma)||p(\Sigma))
\end{aligned} \tag{3.7}
$$

To apply reparameterization trick, we take transformation from normal distribution to $\theta = \mu + \Sigma^{1/2}\epsilon$ where $\epsilon \sim N(0, I)$ drawn as $1 \times K$ vector, as equation 3.8

$$
\begin{aligned}
= & \sum_{d=1}^{D} \left(\mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)}\left[\log p(w_d|\sigma(\mu_{\theta_d}(w_d) + \Sigma^{1/2}\epsilon), \alpha)\right] - KL(q(\theta_d)||p(\theta_d|\mu, \Sigma))\right) \\
& - KL(q(\Sigma)||p(\Sigma|\gamma))
\end{aligned} \tag{3.8}
$$

The KL-divergence for the logistic-normal distribution is given as equation 3.9 closed-form expression, the KL-divergence between $q(\theta_d)$ and $p(\theta_d)$ becomes

$$
\mathrm{KL}(q(\theta_d|w_d, \mu_0, \Sigma_0)||p(\theta_d|\Sigma_1)) = \frac{1}{2}\left(\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + \mu_0^\top \Sigma_1^{-1}\mu_0 + \log\frac{|\Sigma_1|}{|\Sigma_0|} - K\right). \tag{3.9}
$$

so the ELBO then becomes 3.10

$$
\begin{aligned}
\tilde{\mathcal{L}} = & \sum_{d=1}^{D} \frac{1}{S}\left[\mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)}\left[w_d^\top \log \sigma(\mu_0(w_d) + \Sigma_0^{1/2}\epsilon)^\top \sigma(\rho^\top \alpha)\right]\right. \\
& \left. - \frac{1}{2}\left(\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + \mu_0^\top \Sigma_1^{-1}\mu_0 + \log\frac{|\Sigma_1|}{|\Sigma_0|} - K\right).\right] \\
& - KL(q(\Sigma)||p(\Sigma|\gamma))
\end{aligned} \tag{3.10}
$$

The expectation log likelihood term in 3.5 can be efficiently approximated by the Monte Carlo sampling method,

$$\begin{aligned}\mathcal{L} \approx \sum_{d=1}^{D} \Big[& \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \Big[w_d^\top \log \sigma(\mu_0^{(s)}(w_d) + \Sigma_0^{1/2}\epsilon^{(s)})^\top \sigma(\rho^\top \alpha) \Big] \\ & -\frac{1}{2}\left(\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + \mu_0^\top \Sigma_1^{-1}\mu_0 + \log\frac{|\Sigma_1|}{|\Sigma_0|} - K \right). \Big] \\ & - KL(q(\Sigma)||p(\Sigma|\gamma)) \end{aligned} \tag{3.11}$$

where the expectation of the reconstruction loss is taken from a set of sample S to compute the unbiased estimate of ELBO. We also apply the minibatch to make able the model perform by sub-sampling the document collection. By equation 3.12

$$\begin{aligned}\tilde{\mathcal{L}} \approx & \frac{\mathcal{D}}{|\mathcal{B}|} \sum_{d\in\mathcal{D}_\mathcal{B}} \Big[\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \Big[w_d^\top \log \sigma(\mu_0^{(s)}(w_d) + \Sigma_0^{1/2}\epsilon^{(s)})^\top \sigma(\rho^\top \alpha) \Big] \\ & -\frac{1}{2}\left(\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + \mu_0^\top \Sigma_1^{-1}\mu_0 + \log\frac{|\Sigma_1|}{|\Sigma_0|} - K \right). \Big] \\ & - KL(q(\Sigma)||p(\Sigma|\gamma)) \end{aligned} \tag{3.12}$$

The transformer loss is calculated over the sum of tokens of a selected sentence sequence SEQ from each document, the loss of each token is evaluated in Cross entropy loss.

$$L_{CrossEntropy} = \sum_{d=1}^{D} \sum_{n=1}^{|\mathrm{SEQ}|} \mathrm{CrossEntropy}(w_{dn}) \tag{3.13}$$

where

$$\mathrm{CrossEntropy}(w) = -\sum_{i=1}^{V} p(w^{(i)}) \log \hat{p}(w^{(i)}) \tag{3.14}$$

it is a metric for comparing the probability between the word probability $p(w)$ and the probability from prediction $\hat{p}(w)$, which is converted to a probability from one hot vector by mapping it to simplex by softmax function that sum to 1.

3.4.3 Optimization step

In algorithm 21, first, initialize the model and variational parameters. Then, for each epoch, we obtain the transformer embedding ρ from the transformer. After that, the topic embedding β is computed by taking softmax of dot-product of ρ and α. Then a minibatch $\mathcal{B}$ is selected from the document for optimization. The number of minibatch is the document collection divides minibatch size where #minibatch $= \frac{\mathcal{D}}{|\mathcal{B}|}$. For each minibatch, the model takes a document and sample lower Cholesky matrix from LKJ Cholesky distribution(description see section 2.3). A topic assignment for document d θ_d is sampled from logistic-normal distribution $\mathcal{LN}(\mu, \mathrm{LL}^\top)$, where μ is sampled from half-Cauchy distribution and covariance is a transformation from equation 2.4. For each word position n, a word is sampled from the softmax of dot-product of transformer embedding ρ and NN weight α. After the sampling process for the document collection, we estimate the ELBO loss L_{ELBO} for the topic model, and the cross-entropy loss

$L_{CrossEntropy}$. Remind that the topic model and transformer take input differently. The topic model part takes bag-of-words input, a document-vocabulary matrix $D \times V$ counting the occurrence of vocabulary v in document d. While transformer takes a sequence of the document as input. To calculate the loss of the model , we sum up the ELBO loss L_{ELBO} and cross-entropy loss for transformer $L_{CrossEntropy}$. Then a stochastic gradient is computed by backpropagation. a gradient step to . The process iterates until the maximum iteration is reached.

Algorithm 5: Training on TECTM

Initialize model and variational parameters
for *epoch* $i = 1, 2, \ldots N$ **do**
    Obtain trnasformer embedding ρ
    Compute $\beta = \text{softmax}(\rho^\top \alpha)$
    Choose a minibatch $\mathcal{B}$ of documents
    foreach *document* d *in* $\mathcal{B}$ **do**
        Compute $\mu_d = \mu_\phi(w_d)$
        Sample $L \sim \text{LKJChol}(\gamma)$
        Sample $\theta_d \sim \mathcal{LN}(\mu_d, \Sigma)$ where $\Sigma = LL^\top$
        foreach *word position* n *in docuemnt* N_d **do**
            Sample word $w_{dn} \sim \text{Cat}(\sigma(\theta_d^\top \beta))$
        end
    end
    Estimate ELBO loss $\mathrm{L_{ELBO}}$ from Eq. 3.12
    Compute Transformer loss $\mathrm{L_{CrossEntropy}}$ from Eq. 3.13
    Compute the total loss $\mathrm{L} = \mathrm{L_{ELBO}} + \mathrm{L_{CrossEntropy}}$
    Compute the stochastic gradient via backpropagation
    Take a stochastic gradient step
    Update model parameters
    Update variational parameters
end

3.5 Results & Evaluations

In this chapter, we evaluate our model and the other algorithms.

3.5.1 Experiment Testing

The experiment will be conducted with a number of existing proposed topic models as mentioned related work section above. We conduct an experiment with those baseline algorithms and evaluate them in terms of accuracy and running time. Some of the source code of competitive were provided by their authors in Github[1]. The outcome result will be extensively studied and conclude the insight behind the algorithms and methodologies. Detail to be stated in section 3.5.3. Our probabilistic part of the model is implemented using Pyro[4], while the model optimization and transformer implementations are based on PyTorch[34].

3.5.2 Dataset

To evaluates the performance of the model, we selected *20Newsgroups* and *Reuters-21578* data sets in our evaluation stage. 20Newsgroups consist of 18,846 news-

[1]For instance, Correlated Topic Model(CTM), https://github.com/blei-lab/ctm-c

group documents [2] and the Reuters-21578 includes 10,788 documents in total. Both of the datasets will be preprocessed to remove stop-words and stemming before the evaluation. Both data set were separated into training/testing set for the training and evaluation process. *20Newsgroups* data set contains around 20,000 newsgroups documents, which are divided into 20 different groups. In the preprocessing stage, we remove the document with only one word. We filter the stop-words, remove the word with special characters. The frequency of words is limited to between 2%-50%. After the preprocessing, the data set was split into 11314, 7532 documents with 5651 vocabularies. *Reuters-21578* data set is a collection of documents from Reuters newswire in 1987. After performing the preprocessing, the processed data set consisted of 7769, 3019 documents for train/test documents with 1622 vocabularies. On the data preprocessing stage, we perform tokenization, stopword removal, lemmatization on the documents.

Transformer learning task For transformer embedding training, we shape the dataset into a sequence set S of equal distance of pre-defined sentence length SEQLEN. Token <PAD> are padded to the sequence when the sentence in i-index $S[i]$<SEQLEN>. The out-of-vocab tokens are replaced with <OOV> token. The sequence dataset is to train the transformer embedding as input in the training process. To organize the training task, we define a set of tokens and target to Transformer learn it explicitly. We configure 15% of tokens within a sentence are to be masked for the training task. First, we replace a <MASK> token on the selected word position of the token input. Other tokens will be filled a <IGNORE> token on the corresponding target position to exclude those token positions to be calculated in the prediction score.

3.5.3 Models

We compare the model performance with a number of rivals. We take Latent Dirichlet Allocation (LDA)[9] as the baseline model. Other models include Transformer[42][3], ProdLDA[37] and Embedded Topic Model(ETM)[14]. LDA fits the model with with mean-field variational inference [4] ProdLDA, is a topic model learning using amortized inference to approximate the variational distribution on document-topic proportion. ETM model, a topic model built on top of the ProdLDA model, uses dot-product of word embedding and topic embedding to represent the topic-word distribution β.

3.5.4 Algorithmic Settings

To perform posterior inference, we employed Stochastic Variational Inference (SVI) [19] for the optimization problem. We set the minibatch size to 512 documents. For LDA, we applied the model provided from sklearn package (version 0.24.0) [5]. For ETM, we run the experiment with the parameter suggested [14]. For ProdLDA, we perform optimization with inference network architecture as described in the paper [37]. To train the models, every model runs on 300 epochs to give the best performance. To perform optimization, we use Adam for the gradient ascent algorithm, and we set the learning rate to 2e-3. we use $l2$-regularization to the 1e-6, We applied the settings [37] to perform amortized

[2] `http://qwone.com/~jason/20Newsgroups/`

[3] Not a topic model, but we think it is worth making a comparison still.

[4] Adopted from scikit-learn library

[5] Sklearn website `https://scikit-learn.org/stable/index.html`

inference. The inference architecture included 300 dimensions of hidden layers. The dimension for embedding ρ are set to 256. For the Transformer model settings, we define the sequence length to 20, a number of heads to 8, 4 layer stacks of transformer encoder, and 256 hidden dimensions.

3.5.5 Quantitative Result

Perplexity has been known for evaluating the fitness of NLP models, however, it may not be the best metric to examine the wellness of a topic model[33] Secondly, to compute the perplexity for the topic model using AEVB, we follow the previous work, using the variational lower bound to compute the perplexity. For the same reason, the perplexity may not perfectly reflects the true quality of a topic model. Hence, we take the topic coherence [31] as the main measure to evaluate the models. Also, sometimes even a topic model could obtain a high coherence score, the topics could be repeated many times that yields poor results. We also implemented topic diversity [14] as one of the measures of the topic model, the measure that evaluates how well each topic contains a distinct word from other topics. In this section, we evaluate the model with the following metric: Perplexity, Topic Coherence (TC), Topic Diversity (TD).

#Topic		k=20			k=50	
Metrics	PPL	TC	TD	PPL	TC	TD
LDA	478.8	**0.248**	0.702	507.1	0.193	0.554
ETM	279.0	0.213	0.500	336.8	0.200	0.211
ProdLDA	796.6	0.188	**0.792**	521.8	0.180	**0.609**
TECTM	**259.6**	0.237	0.586	**263.0**	**0.226**	0.367

Table 3.1: Result for Reuters-21578 dataset

#Topic		k=20			k=50	
Metrics	PPL	TC	TD	PPL	TC	TD
LDA	2442.7	0.168	**0.774**	2538.2	0.154	**0.713**
ETM	**1640.5**	0.191	0.592	**1715.5**	0.159	0.337
ProdLDA	6018.5	0.072	0.734	9057.9	0.014	0.703
TECTM	1954.4	**0.200**	0.630	1944.6	**0.194**	0.509

Table 3.2: Result for 20Newsgroups dataset

Reuters-21578 On the data set Reuters-21578, our model performs the best in perplexity, with 259.6 and 263.0 when k=20 and k=50 respectively. Our model also performs the best on topic coherence when $k = 50$, where it is 0.226, with a competitive topic coherence score when $k = 20$ as well. To focus on the TD score, our model did not beat the best model, but also maintain a good enough TD score to generate a variety of topics, with 0.586 and 0.367 when $k = 20$ and $k = 50$ respectively. As result apparently, our model performs well on several metrics.

20Newsgroups On the result from table 3.2, displays that our model has outperformed the other model by TC score. On perplexity score, ETM obtain the best score when $k = 20$ and $k = 50$. On the other hand, LDA has the best TD score on both $k = 20$ and $k = 50$. It can When $k = 20$, our model has 1954.4 in perplexity score, 0.200 TC score and 0.630 in TD score. When $k = 50$, our model has 1944.6 in perplexity score, 0.194 TC score and 0.509 in TD score. ProdLDA, performs the worse in both perplexity and topic coherence scores.

It can be seen that, when the number of the topic increases, the topic model performance decrease proportionally. For instance, ProdLDA has a quite well metric score on small data set such as Reuters-21578. However, its performance drags down drastically on a bigger data set such as 20Newsgroups. Also, we observe that our model is capable to maintain a relatively good topic coherence score when the topic increases from the result above. To compare with LDA, it could obtain a high enough TC and TD scores in our experiment settings, however, our model does better in TC scores in both topic numbers we conducted investigations.

3.5.6 Training

From figure 3.2, 3.3, display the training process of the training loss and log probability by 200 epochs. We observe that the negative log probability is decreasing over epochs. For the metrics, the topic coherence and the topic diversity improve slowly along with the training proceeds, the perplexity also decreases along.

3.5.7 Qualitative Result

The proposed model will be evaluated with a number of specifically selected topics and examined with their performance separately. The result will be exhaustively compared with other existing models.
From table 3.3, we have selected some topic words each model generated from 20Newsgroups when $k = 20$. The topics represent space, operating system, religion, encryption, and guns respectively. Our model has shown capability on capturing keywords from each topic, such as on topic "space": *nasa*, *space*, *jpl*, *moon*, *earth*, *station*, *flight* are the outputs.

3.5.8 Visualization

To clearly demonstrate the representation for the word embedding space inside the parameters obtained, we ran a t-SNE algorithm to map the topic-word representation into a 2-dimension continuous space. We selected and zoomed in a specific topic with its neighboring topics with a red box and displayed over the figures. In figure 3.4 and 3.6, demonstrate the t-SNE visualization of the topic-word distribution for 20Newsgroups for k=20 and k=50.

We also compare the model with the ETM as displayed in the figure 3.5. In figure 3.4, the topic "space"(which consists of words like space, earth, orbit, etc) laying on the 2-dimensional space. The most neighboring topic includes "hardware" and "operating system", which appear the words such as "cable", "cd", "power" for "hardware" topic, and "images", "graphics" and "files" for "operating system" topic. Generally speaking, ETM is not able to distinguish the words of topics into visible clusters, as shown in the figure, the upper part of word dots with different topic labels mix. And inside the topic cluster, there is also some word from other topic mixed into it. This may imply ETM does not generate a good enough topic-word representation to distinguish the difference between topics.

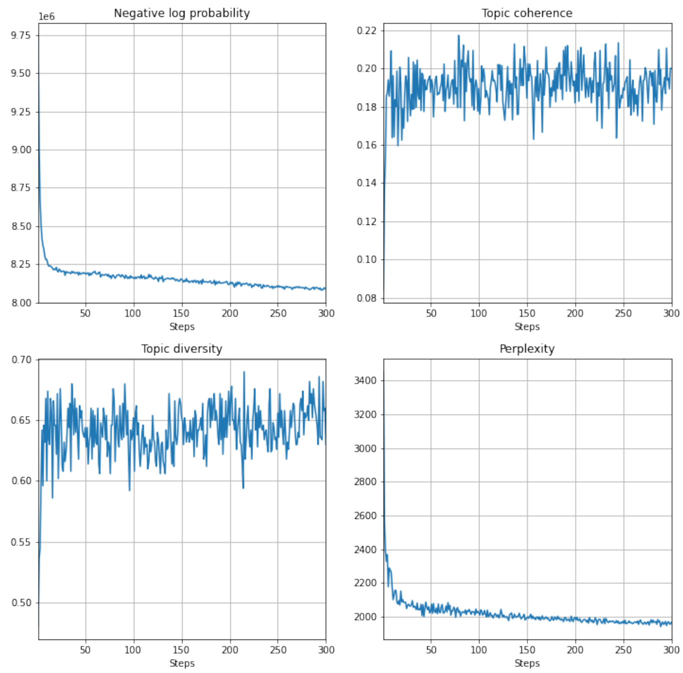

Figure 3.2: Training performance for 20Newsgroups in 20 topics, in terms of negative log probability, topic coherence, topic diversity and perplexity

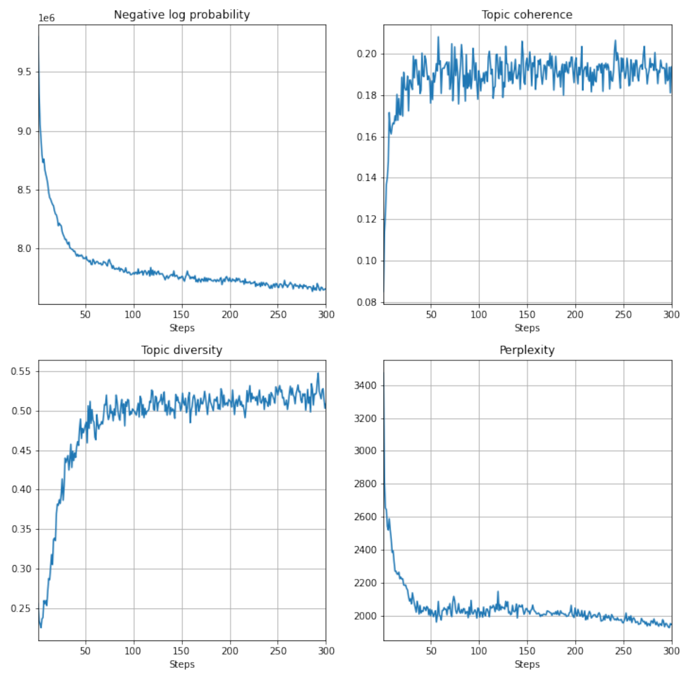

Figure 3.3: Training performance for 20Newsgroups in 50 topics, in terms of negative log probability, topic coherence, topic diversity and perplexity

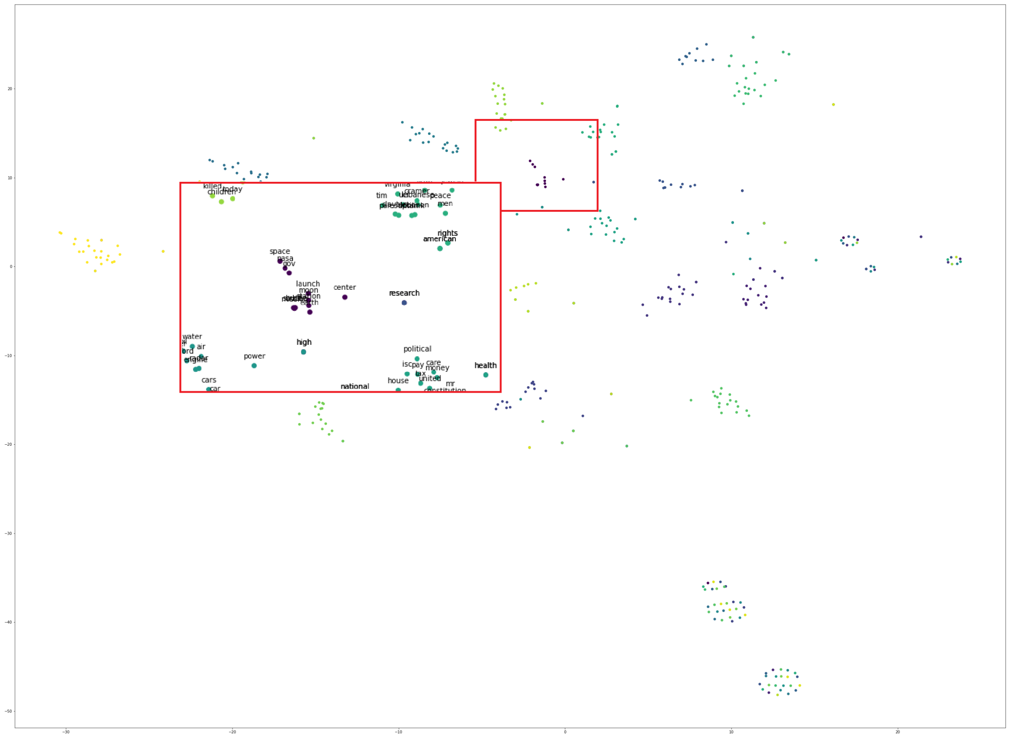

Figure 3.4: t-SNE Visualization for our model Topic=20

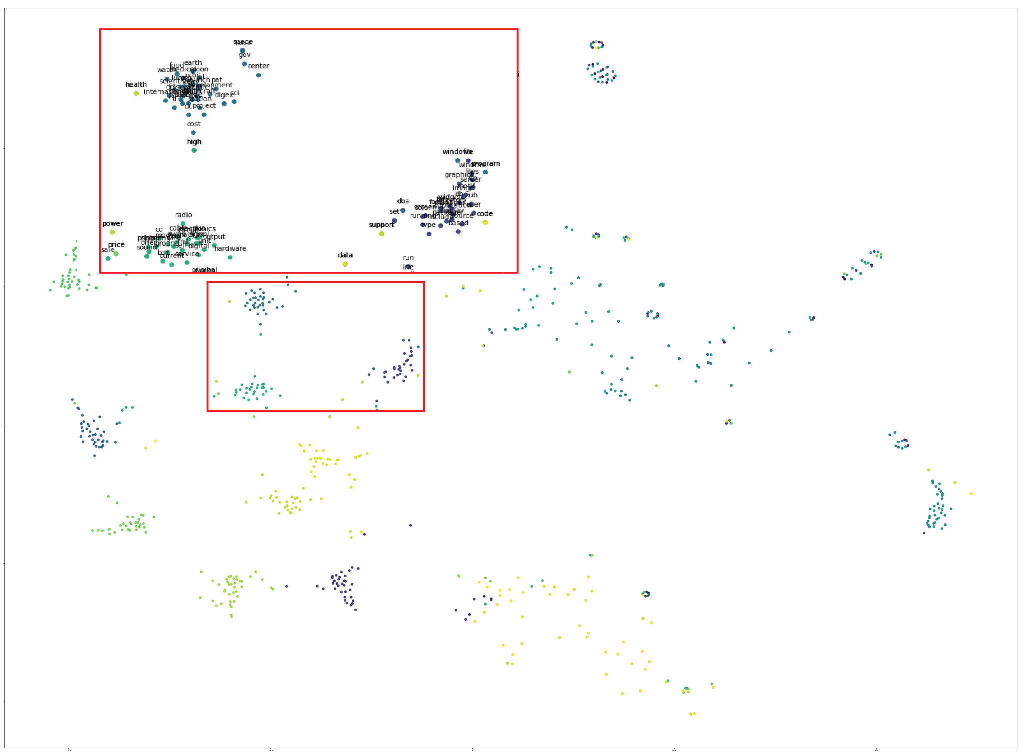

Figure 3.5: t-SNE visualization for ETM Topic=20

Our Model
nasa gov **space jpl moon earth station flight** research digex
windows window problem **dos** running **file mouse** mit de ms
god jesus christian people faith bible time church good things
key chip **encryption** clipper **security privacy** government **keys** public escrow
gun people control government **guns weapons** american make clinton state
ProdLDA
nasa space gov people **station time orbit** dc program shuttle
scsi drive controller max **drives ide** senior **tape** time people
god jesus atheists christian bible religion atheism christians word truth
key crypto session nt chips chip **serial** dos **keys encrypted**
gun people god religion writes life morality ohio argument question
LDA
space, nasa, gov, access, **launch, earth**, digex, **moon, orbit**
file, window, program, ftp, **files**, **server**, **image, graphics, windows**
god, people, **jesus**, **christian**, **bible**, writes, life, **christians**, time
key, **encryption**, chip, clipper, **keys**, **security**, **government**, **privacy**
gun, guns, law, police, people, **weapons, crime, fbi**, control
ETM
space, nasa, gov, mr, president, health, research, year, center
windows, file, window, program, files, server, **version**, dos, **image**
god, people, **jesus**, **christian**, **israel**, **bible**, **jews**, **christians**, **israeli**
key, encryption, chip, clipper, **keys**, **privacy**, **security**, technology, government
gun, people, government, **law**, state, **guns**, article, **weapons**, control

Table 3.3: Top-9 words for each topic from 5 topics selected

3.5.9 Discussion

We have compared our model with a number of state-of-the-art models on multiple data sets. As we mentioned in the previous section, perplexity may not be a good measure for comparing the topic models. Also, normally the perplexity for the topic model is calculated using the posterior distribution, while the topic model using Autoencoding Variational Bayes(AEVB) is calculated using the variational lower bound[37]. The results between these models may not able to conclude easily. We have carefully compared the models on topic-model-specific metrics like topic coherence and topic diversity. The result exhibits our model has an outstanding performance on obtaining high-quality topic words over various predefined topic numbers, especially in topic coherence score. In two of the metric we compared, topic coherence and topic diversity, our model shows its strength in generating high-quality topics and diversified words. The comparison on the quantitative result also demonstrates that our model is robust to generate high-quality topics upon the topic number defined increases. In the t-SNE visualization, our model also made a clearer representation of 2-dimensional space which implies a better intra-cluster divergence the model distinguishes the difference between topics.

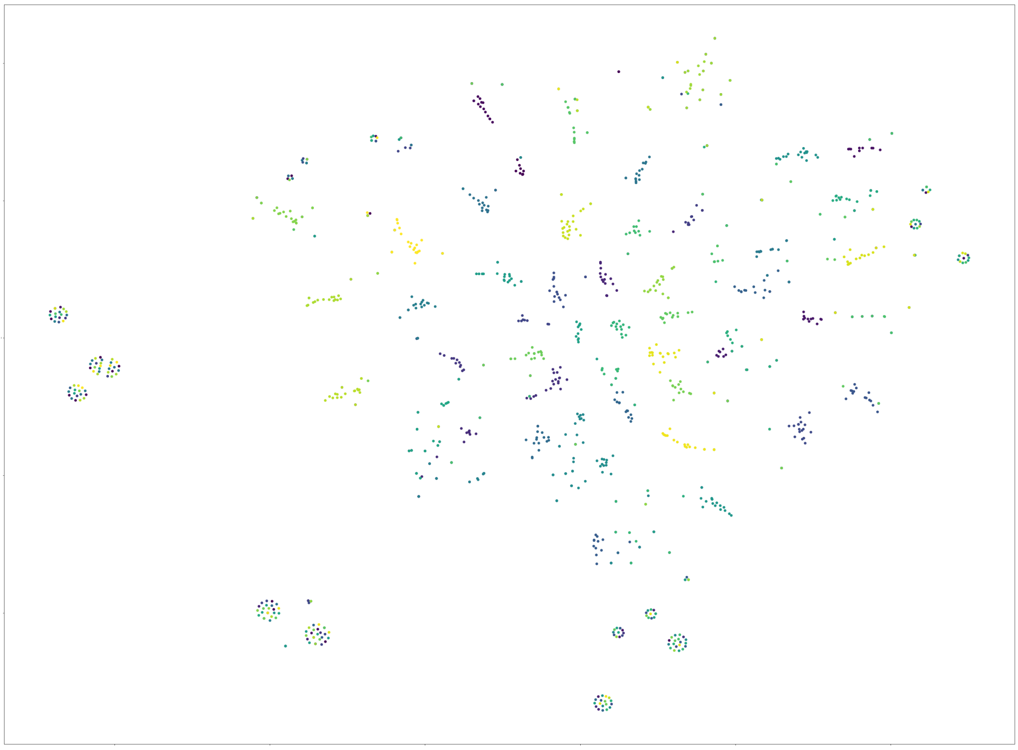

Figure 3.6: Visualization #Topics:50

Chapter 4

Times Series Topic Retrieval with TECTM

In most real-life cases, the context (or formally topic information), that to be mentioned in the media such as news and documents are changing over time. Also, the word meaning and slang at the time may not valid in another span of time. In the previous chapter 3, we have implemented a topic model that could capture high-quality topic words. However, the model cannot distinguish the difference of topic representation and the topic-word relation along the time of the model. In this chapter, we expand the embedded topic model to deal with the times-series task, namely Dynamic Transformer embedded topic model. The model utilizes the time information.

4.1 Introduction

Most of the existing topic models are designed to handle unified document set, and such no time specified information are assumed forehand. Dynamic Topic Model(DTM)[7] is an extended version from LDA, in which a state-space model introduce a latent variable to control the model parameters and proportion over time stamp $1 \dots T$. However, the model does not handle correlation information for topics obtained. Dynamic Correlated Topic Model(DCTM)[41] have considered the correlation information between topics though out the time. The author proposed . However, the model estimate the mean of the prior distribution of document-topic proportion with bag-of-words per-document, which may miss out to explore the time-topic relation from exploiting the input of bag-of-word per-time period. Our model demonstrate a better strength in obtaining topic coherence scores. Dynamic Embedded Topic Model(DETM)[13] constructed a times-series Embedded Topic Model by constructing the state-space model to guide the latent variables that control the mean for topic-proportion prior and the topic-word proportion. Likewise, the model does not consider correlation information. Besides, our model refines the word embedding quality on the latent space and hence improves overall topic-word predictive performance.

For this reason, we propose Dynamic Transformer Embedding Correlated Topic Model(DTECTM), a model that we build on top of the model from chapter 3. That is, a model that make use of the transformer embedding to compute the topic words from the document set. Additionally, we manage to capture the time-series information. By using Gaussian Process Latent Variable Model(GPLVM), we infer the topic distribution in different time from the latent variables it obtains. Our model takes the information from GPLVM as residual input and bag-of-words to infer the document-topic proportion. The results are compared with data sets against other state-of-the-art models.

4.2 Related works

Moreover, time-series model is one of the most practical for real application in topic modeling, which it is essential to extract keywords along the time line. In chapter 4 we will explore and implement . As an introductory, here we discuss the related works in advance. Blei [7] extended a time-series topic model on top of LDA, namely dynamic topic model(DTM). The model assumes the prior for topic-word proportion is an Markov state-space model along time t. The posterior is approximated with variational bayes and Kalman filter inference. Hening[17] proposed Kernel topic model, which the model is equipped the gaussian process with kernel covariance as the hyperparameter of Dirichlet prior for document-topic proportion. Tomasi[41] implemented a time-series correlated topic model, using Gaussian process to model for modeling the hyperparameter of topic-word proportion and the mean for document-topic proportion, along with using Wishart process for parametrizing the covariance matrix. Dieng[14] improve a model which on top of the ProdLDA topic model, implemented Word2Vec semantics to further improve the performance on topic coherence and predictive distribution. And respectively, the same author[13] extended the dynamic embedded topic model from previous embedded topic model. The model perform inference on posterior by deriving variational lower bound and amortized inference. The model take assume of topic-word proportion and document topic proportion as a Markov state-space model.

4.3 Model description

The DTECTM utilizes the Transformer as embedding to the topic-word representations. To compare with the original topic model, the word-topic distribution β is the dot-product of the Transformer embedding ρ and the topic embedding α. First, the topic embedding α embeds the vocabulary into L-dimensional space, which is by Transformer embedding ρ. Second, the context embedding maps the embedding into K-dimensional space. In the generative process, the LKJTM uses the topic embedding to form a per-topic vector to represent the meaning over the vocabulary.

To express the way word embedding to be applied in our model, $\rho \in \mathbb{R}^{V \times L}$ is the Transformer embedding in the model. ρ_v is a vector represents the embedding of vocabulary on n-th index. To explain the way our model capture time-related information from document set, we here discuss variables change over time. The topic embedding $\{\alpha_k^{(t)}\}_{t=1}^{T} \in \mathbb{R}^L$ is a vector distributed at specific k topic. Topic proportion θ_d is same as typical topic model, which a simply a vector represent proportion for each topic on document d. The latent variable η decide the topic proportion holds on each timestamp ranged between $1, \ldots, T$.

4.3.1 Generative Model

The generative process is as following:

Algorithm 6: Generative Process for DTECTM

Initialize hyperparameters
Obtain transformer embedding ρ
for *time t in T* **do**
  Draw topic embedding $\alpha^{(t)} \sim \mathcal{N}(\alpha^{(t-1)}, \xi^2 I)$
  Draw topic proportion mean $\eta_t \sim \mathcal{N}(0, I)$
  Sample correlation $L_t \sim \text{LKJChol}(\gamma_t)$
end
for *document d in D* **do**
  Sample topic proportion $\theta_d \sim \mathcal{LN}(\eta_{t_d}, \Sigma_{t_d})$
  for *word position n in N_d* **do**
    Sample word $w_{d,n} \sim \text{Cat}(\sigma(\theta_d^\top(\rho^\top \alpha^{(t_d)})))$
  end
end

From algorithm 6, first the model draws a topic embedding $\alpha^{1:T}$ from normal distribution at time $1, \ldots, T$. At time step 0, the topic embedding initialized at $\mathcal{N}(0, I)$. Then a topic mean $\eta_t \in \mathbb{R}^K$ over timestamps is generated from the Gaussian Process Latent Variable Model(GPLVM), which performs inference a dimensions of topic K from number of vocabularies V dimension. Specifically, taking bag-of-word by time w_t, which is collected by categorizing the document by time and group them into word count matrix by timestamp. And then a normalization is performed to make sure the words in different timestamp are in same proportion. For each document, draw a topic proportion θ_d from logistic-normal distribution $\mathcal{LN}(\cdot, \cdot)$ condition on topic mean η_{t_d} at the timestamp t of document d, and the variance $\xi^2 I$. After that, for each word position n in N_d, a word is drawn from the dot-product of word embedding ρ and topic embedding $\alpha_d^{(t_d)}$ at timestamp t_d.

4.3.2 Joint Distribution

To describe the joint distribution for the model, equation 4.1

$$\begin{aligned} p(W, \theta, \Sigma, \eta, \alpha | \rho, \gamma) = \prod_{d=1}^{D} &\left[p(\theta_d | \eta_{t_d}, \Sigma_{t_d}) \prod_{n=1}^{N_d} p(w_{d,n} | \theta_d, \rho, \alpha^{(t_d)}) \right] \backslash \\ &\prod_{t=1}^{T} \left[p(\eta_t) p(\Sigma_t | \gamma_t) \prod_{k=1}^{K} p(\alpha_k^{(t)} | \alpha_k^{(t-1)}) \right] \end{aligned} \tag{4.1}$$

For the bag-of-word input, we have V vocabularies over D documents. A number of K topics are defined to be introduced in the model. Each document is labeled a timestamp t over a time span between $1, \ldots, T$. $\theta_d \in \mathbb{R}^K$ is the topic proportion for document d. $\eta_t \in \mathbb{R}^K$ is the topic proportion corresponds to time t of document d. $w_d \in \mathbb{R}^V$ is the bag-of-word of distribution at document d. $w_t \in \mathbb{R}^V$ is the bag-of-word of distribution at time t, which we arranged the documents in different timestamp and group them into bag-of-word representation in dimension $\mathbb{R}^{V \times T}$. In particular, w_t is normalized to attain the same ratio of tokens on each single time stamp. $\alpha_k^{(t)} \in \mathbb{R}^V$ is the topic embedding for topic k at time t. The topic embedding demonstrates the vocabulary representations on the specific time stamp for the document. $\rho \in \mathbb{R}^{V \times L}$ is the transformer embedding that maps the words into L dimension of continuous latent space.

$$p(w_{d,n}|\theta_d, \rho, \alpha^{(t_d)}) = \text{Cat}\left(\theta_d \sigma(\rho^\top \alpha^{(t_d)})\right) \tag{4.2}$$

The $p(\Sigma_{t_d}|\gamma_{t_d})$ draws the covariance prior to the document-topic proportion θ_d. The LKJ correlation prior generates a positive definite matrix of $\mathbb{R}^{K \times K}$

$$p(\Sigma_t|\gamma_t) = \text{LKJCorr}(\gamma_t) \tag{4.3}$$

where Σ_{t_d} can be decomposed in form of the product of two lower triangular matrices. And such the correlation prior distribution can be factorized as a LKJCorrChol parameterized by the concentration variable γ_t, where it produces lower triangular matrices L_t that can be converted into covariance matrix by transformation mentioned in 2.4.

$$\text{LKJCorr}(\gamma_t) = \Sigma_t = L_t L_t^\top \tag{4.4}$$

$$L_t \sim \text{LKJCorrChol}(\gamma_t) \tag{4.5}$$

The document-topic proportion θ_d is a logistic-normal distribution that conditioned on the mean prior η_t and covariance matrix Σ_t. The distribution ensure the proportion vector $\theta_d \in \mathbb{R}^K$ in every document d are constrained into a simplex that values are summed to 1.

$$p(\theta_d|\eta_{t_d}, \Sigma_{t_d}) = \mathcal{LN}(\eta_{t_d}, \Sigma_{t_d}) \tag{4.6}$$

The time-to-topic proportion is a Gaussian Process Latent Variable Model(GPLVM), which takes an input of latent variable from lower dimension space to recover the observed data. We take the latent variable to induce the relation between time-topic representation. The $\eta_{1:T} \in \mathbb{R}^K$ is a latent variables learnt from GPLVM that contributes the time-topic proportion as a mean for variable θ. We can use the latent variable η to recover the observed data $w_{1:T}$. The latent function is then computed using the kernel $\mathcal{K}_\theta$ parameterized by the hyperparameter θ and the information from η. $\mathcal{K}_\theta \in \mathbb{R}^{V \times V}$ is the kernel function that determine the shape of covariance matrix, which controls the shape of outcome distribution. Then the observed bag-of-words w is recovered back by normal distribution taking mean of $f_n(\eta_t)$ and variance σ^2.

$$p(\eta_t) = \mathcal{N}(0, I) \tag{4.7}$$

$$p(f_n|\eta_t, \theta) = \prod_{n=1}^{V} \mathcal{N}(0, \mathcal{K}_\theta) \tag{4.8}$$

$$p(w_{t,n}|f_n, \eta_t) = \prod_{t=1}^{T} \prod_{n=1}^{V} \mathcal{N}(f_n(\eta_t), \sigma^2) \tag{4.9}$$

The topic-word proportion is contributed as $p(\alpha^{(t)}|\alpha^{(t-1)}) = \mathcal{N}(\alpha^{(t-1)}, \xi^2 I)$, which is a Markov chain conditioned on variance parameterization ξ^2. The topic embedding $p(\alpha_k^{(t)}|\alpha_k^{(t-1)})$ is a random walk go from time step $1 \ldots T$, each time step t take the prior mean from the sampled value from previous time step $t-1$ and Gaussian noise ξ^2 as variance.

$$p(\alpha_k^{(t)}|\alpha_k^{(t-1)}) = \mathcal{N}(\alpha_k^{(t-1)}, \xi^2 I) \tag{4.10}$$

4.4 Inference and Estimation

Since the posterior is intractable to compute, we apply variational inference to approximate the parameters for the log-marginal likelihood.

4.4.1 Variational Distribution

To begin with, we first set up variational distribution to approximate the parameter of the model.

$$q(\theta, \eta, \Sigma, \alpha) = \prod_{d=1}^{D} q(\theta_d | w_d, \eta_{t_d}, \Sigma_{t_d}) \prod_{t=1}^{T} q(\eta_t | w_t) q(\Sigma_t | \gamma_t) \prod_{k=1}^{K} q(\alpha_k^{(t)}) \tag{4.11}$$

The variational distribution for topic proportion $q(\theta_d|\eta_{t_d}, w_d)$ is logistic-normal distribution. We applied amortized inference to approximate the model, which the mean and covariance matrix is generated by two inference network μ_ϕ and σ_ϕ taking bag-of-word input w_d at each document d and residual input η_{t_d} end up $\mathbb{R}^{D\times(V+K)}$ dimension in the input space. The output for time-topic proportion is applied to the residual connection on the amortized inference of document-topic proportion θ. The inference network for variational parameter ϕ take the residual input from stacked input: both $\mathbb{R}^V$ bag-of-words vectors w_d, and the $\mathbb{R}^K$ time-topic proportion η_{t_d}. Then we transform the using reparameterization trick. To perform amortized inference for θ, we apply Layer Normalization[1] to normalize the input vectors. The LayerNorm performs normalization over features, which enables better training. Also, it is more stable than batch normalization for training while the batch size is pretty small. And thus it can maintain a lower variance than batch normalization does throughout the training loop.

$$q(\theta_d|\eta_{t_d}, w_d, \Sigma_{t_d}) = \mathcal{N}(\mu_\phi(x), \Sigma_{t_d}) \tag{4.12}$$
$$x = \text{LayerNorm}([w_d, \eta_{t_d}]) \tag{4.13}$$

The variational distribution for $q(\eta_t|w_t)$ is basically the normal distribution parametrized by two inference networks, which takes the input from bag-of-words w_d and the variable from [25], L is the dimension for latent input space, V is the token size.

$$q(\eta_t|w_t) = \int \prod_{i=1}^{V} q(w_i) \prod_{d=1}^{L} p(f_d|u_d, W) q(u_d) du_d \tag{4.14}$$

The variational distribution for $q(\alpha_k^{(t)})$ is the embedding is parameterized mean μ_φ and σ_φ conditioned on local parameter φ. Notice that φ is not a variational parameter as usual amortized inference.

$$q(\alpha^{(t)}) = \mathcal{N}(\mu_\varphi^{(t)}, \sigma_\varphi^{(t)}) \tag{4.15}$$

both variational distribution $q(\theta_d|\eta_{t_d}, w_d)$ and $q(a^{(t)})$ applied reparameterization trick [22] with transformation $\mathcal{N}(\mu, \sigma) \approx \mu + \epsilon\sigma^{1/2}, \epsilon \sim \mathcal{N}(0, 1)$, to avoid high variance on the variational variables.

4.4.2 Evidence lower bound (ELBO)

We take the log marginal likelihood from eq. 4.1. Noted the given equation could derive the ELBO by simply applying the Jensen inequality. For sake of simplicity, we decompose the marginal likelihood into two parts, the document-topic proportion part, and the time-topic proportion part. Then summing up the terms as eq. 4.16. In such way, we are able to derive a ELBO without calculating the KL-divergence for $q(\eta_{t_d})$ and $p(\eta_{t_d}|w_t)$. And so we could obtain the ELBO for $\log p(\eta_{t_d}|w_t)$ conveniently from [40].

$$
\begin{aligned}
\mathcal{L} \geq & \mathbb{E}_q[\log p(W, \theta, \eta, \Sigma, \alpha|\rho, \gamma)] - \mathbb{E}_q[\log q(\theta, \eta, \Sigma, \alpha)] \\
= & \sum_{d=1}^{D} \sum_{n=1}^{V} \mathbb{E}_q[\log p(w_{d,n}|\theta_d, \rho, \alpha^{(t_d)})] + \sum_{d=1}^{D} \mathbb{E}[\log p(\theta_d|\eta_{t_d}, \Sigma_{t_d})] \\
& + \sum_{t=1}^{T} \mathbb{E}_q[\log p(\eta_t)] + \sum_{t=1}^{T} \mathbb{E}_q[\log p(\Sigma_t)] + \sum_{t=1}^{T} \sum_{k=1}^{K} \mathbb{E}_q[\log p(\alpha_k^{(t)}|\alpha_k^{(t-1)})] \\
& - \sum_{d=1}^{D} \mathbb{E}_q[\log q(\theta_d|\mu_\phi(x_d), \Sigma_t)] - \sum_{t=1}^{T} \mathbb{E}_q[\log q(\eta_t|w_t)] - \sum_{t=1}^{T} \mathbb{E}_q[\log q(\Sigma_t|\gamma_t)] \\
& - \sum_{t=1}^{T} \sum_{k=1}^{K} \mathbb{E}_q[\log q(\alpha_k^{(t)}|\alpha_k^{(t-1)})]
\end{aligned} \tag{4.16}
$$

and by rearranging the terms, the ELBO can be represented as follows, where the first term is the reconstruction loss, and the remaining is the KL-divergence between the prior and variational distribution for its variational parameters.

$$
\begin{aligned}
= & \sum_{d=1}^{D} \sum_{n=1}^{V} \mathbb{E}_q[\log p(w_{d,n}|\theta_d, \rho, \alpha^{(t_d)})] - \sum_{d=1}^{D} \mathrm{KL}(q(\theta_d|w_d, \eta_{t_d}, \Sigma_{t_d})||p(\theta_d|\eta_{t_d}, \Sigma_{t_d})) \\
& - \sum_{t=1}^{T} \mathrm{KL}(q(\eta_t|w_t)||p(\eta_t)) - \mathrm{KL}(q(\Sigma_t|\gamma_t)||p(\Sigma_t)) \\
& - \sum_{k=1}^{K} \mathrm{KL}\left(q(\alpha_k^{(t)}|\alpha_k^{(t-1)})||p(\alpha_k^{(t)}|\alpha_k^{(t-1)})\right)
\end{aligned} \tag{4.17}
$$

For sake of simplicity, we separate the ELBO into three parts. We first derive the ELBO for the document-topic model part, denoted as $\mathcal{L}_1$, we can obtain $p(w_d, \theta_d, \alpha|\eta_{t_d}) = p(w_d|\theta_d, \alpha^{(t_d)})p(\theta_d|\eta_{t_d})p(\alpha)$ by factorization.

$$
\begin{aligned}
\mathcal{L}_1 = & \sum_{d=1}^{D} \sum_{n=1}^{V} \mathbb{E}_q[\log p(w_{d,n}|\theta_d, \rho, \alpha^{(t_d)})] - \sum_{d=1}^{D} \mathrm{KL}(q(\theta_d|w_d, \eta_{t_d}, \Sigma_{t_d})||p(\theta_d|w_d, \eta_{t_d}, \Sigma_{t_d})) \\
& - \sum_{t=1}^{T} \mathrm{KL}(q(\Sigma_t^\phi|\gamma_t)||p(\Sigma_t))
\end{aligned} \tag{4.18}
$$

To speed up computation, we apply mini-batching as the previous chapter

$$
\tilde{\mathcal{L}}_1 \approx \frac{|\mathcal{D}|}{|\mathcal{B}|} \sum_{d \in \mathcal{D}_\mathcal{B}} \left[\sum_{n=1}^{V} \mathbb{E}_q[\log p(w_{d,n}|\theta_d, \rho, \alpha^{(t_d)})] - \mathrm{KL}(q(\theta_d|w_d, \eta_{t_d}, \Sigma_{t_d})||p(\theta_d|w_d, \eta_{t_d}, \Sigma_{t_d})) \right] \tag{4.19}
$$

$$
- \sum_{t=1}^{T} \mathrm{KL}(q(\Sigma_t^\phi|\gamma_t)||p(\Sigma_t)) \tag{4.20}
$$

The first term is the expected likelihood term for reconstructing the word w_{dn} from the model, where $p(w_{d,n}|\theta_d, \rho, \alpha^{(t_d)})$ is the log likelihood probability parameterized by the variational parameters θ_d, α^{t_d} and transformer embedding ρ, which $\sigma(\cdot)$ is the softmax function. The topic-word proportion is a dot product of transformer embedding ρ and $\alpha^{(t_d)}$

$$\mathbb{E}_{q(\theta_d)q(\alpha)}[\log p(w_{d,n}|\theta_d,\rho,\alpha^{(t_d)})] = \mathbb{E}_{\theta_d\sim\mathcal{N}(\mu_\phi(x),\Sigma^\phi_{t_d})}[w_{d,n}\theta_d^\top\sigma(\rho^\top\alpha^{(t_d)})_{w_{dn}}] \tag{4.21}$$

and then apply the reparameterization trick to maintain a low-variance gradient estimate to the likelihood term, the transformation $\theta_d = \mu_\phi(x) + (\Sigma^\phi_{t_d})^{1/2}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is gaussian noise variance.

$$= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[w_{d,n}\sigma(\mu_\phi(x_d) + (\Sigma^\phi_{t_d})^{1/2}\epsilon)^\top\sigma(\rho^\top\alpha^{(t_d)})_{w_{dn}}\right] \tag{4.22}$$

The second term is the KL-divergence between $p(\theta_d)$ and $q(\theta_d)$, the distributions are logistic-normal distributions so can be represented in closed-form. So we simply derive the KL-divergence by substitution, the variable x is the parameter stacked with bag-of-words w_d and residual input η_{t_d} from 4.12

$$\begin{aligned}&\mathrm{KL}(q(\theta_d|\mu_\phi(x),\Sigma^\phi_{t_d})||p(\theta_d|\eta_{t_d},\Sigma_{t_d}))\\ =&\frac{1}{2}\left(\mathrm{tr}(\Sigma^{-1}_{t_d}\Sigma^\phi_{t_d}) + (\eta_{t_d}-\mu_\phi(x_d))^\top\Sigma^{-1}_{t_d}(\eta_{t_d}-\mu_\phi(x_d)) + \log\frac{|\Sigma_{t_d}|}{|\Sigma^\phi_{t_d}|} - K\right)\end{aligned} \tag{4.23}$$

For the second part, we derive the ELBO for GPLVM, which the detailed derivation for the ELBO has been discussed in the original paper [40] as the equation 4.24. $w = \{w_t\}_{t=1}^T \in \mathbb{R}^{T\times V}$ is the observed data, which bag-of-words with respect to the word count by that timestamps over the documents. $\eta = \{\eta_t\}_{t=1}^T \in \mathbb{R}^{T\times K}$, the latent variable distributes the topic proportion over time. Known that the dimension reduction is performed, as $K << V$, where the defined number of topics is supposed to be much smaller than the size of vocabularies. f_d is the latent function that takes M inducing point u_d, where M is the number of inducing points defined for the training process. And where u_d is conditioned at input locations $Z \in \mathbb{R}^{M\times K}$. ϕ is the local parameters. λ is the global variational parameters. σ_w is the gaussian noise. Specifically, we only extract the terms $\mathcal{L}_2$, the kl-divergence for eta and u, that to be contributed in the ELBO of the whole model.

$$\begin{aligned}\log p(w_t|\eta_{t_d}) \geq &\sum_{d=1}^V\sum_{t=1}^T \mathbb{E}_{q_\phi(\eta_t)}\mathbb{E}_{p(f_d|u_d,\eta_t)q_\lambda(u_d)}\left[\log\mathcal{N}(w_{d,t};f_d(\eta_t),\sigma_w^2)\right]\\ &\underbrace{-\sum_{t=1}^T\mathrm{KL}(q_\phi(\eta_t)||p(\eta_t)) - \sum_{d=1}^V\mathrm{KL}(q_\lambda(u_d)||p(u_d|Z))}_{\mathcal{L}_2}\end{aligned} \tag{4.24}$$

The KL-divergence for $\alpha^{(t)}$ in time t is a closed-form in normal distribution. And so the equation can be derived as 4.25. The variational distribution for $q(\alpha_k^{(k)})$ is parametrized by two inference network μ_φ and σ_φ, where φ is a local variational parameter. And the prior $p(\alpha_k^{(t)}|\alpha_k^{(t-1)})$ at time t takes the mean from previous step α_k^{t-1} with variance ξ^2 In initial step, the mean for $\alpha^{(1)}$ located at 0.

$$\mathcal{L}_3 = \mathrm{KL}(q(\alpha_k^{(t)})||p(\alpha_k^{(t)}|\alpha_k^{(t-1)})) = \frac{1}{2}\left(\log\frac{\xi^2}{\sigma_\varphi^2} + \frac{\sigma_\varphi^2 + (\mu_\varphi - \alpha_k^{(t-1)})^2}{\xi^2} - 1\right) \tag{4.25}$$

By assembling the ELBO terms from above, and substituting the KL terms $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ from eq. 4.25, 4.24,4.23 into 4.17, the ELBO equation becomes follows

$$\begin{aligned}
\log p(w|\theta, \alpha) \geq & \frac{|\mathcal{D}|}{|\mathcal{B}|} \sum_{d \in \mathcal{D}_{\mathcal{B}}} \sum_{n=1}^{V} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[w_{d,n} \sigma(\mu_\phi(x_d) + (\Sigma_{t_d}^{\phi})^{1/2} \epsilon)^\top \sigma(\rho^\top \alpha^{(t_d)})_{w_{dn}} \right] \\
& - \frac{1}{2} \frac{|\mathcal{D}|}{|\mathcal{B}|} \sum_{d \in \mathcal{D}_{\mathcal{B}}} \left(\mathrm{tr}(\Sigma_{t_d}^{-1} \Sigma_{t_d}^{\phi}) + (\eta_{t_d} - \mu_\phi(x_d))^\top \Sigma_{t_d}^{-1} (\eta_{t_d} - \mu_\phi(x_d)) + \log \frac{|\Sigma_{t_d}|}{|\Sigma_{t_d}^{\phi}|} - K \right) \\
& - \frac{1}{2} \sum_{t=1}^{T} \sum_{k=1}^{K} \left(\log \frac{\xi^2}{\sigma_\varphi^2} + \frac{\sigma_\varphi^2 + (\mu_\varphi - \alpha_k^{(t-1)})^2}{\xi^2} - 1 \right) \\
& - \sum_{t=1}^{T} \mathrm{KL}(q(\Sigma_t^{\phi}) | p(\Sigma_t | \gamma_t)) \\
& - \sum_{t=1}^{T} \mathrm{KL}(q_\phi(\eta_t) || p(\eta_t)) - \sum_{d=1}^{V} \mathrm{KL}(q_\lambda(u_d) || p(u_d | Z)) \\
= & \mathcal{L}
\end{aligned} \tag{4.26}$$

By the above derivation of ELBO, we can compute the unbiased gradient with Monte Carlo sampling.

Algorithm 7: Training on DTECTM

Initialize weights, hyperparameters
for *epoch* $1, \ldots, N$ **do**
    for *time* t *in* $1 \ldots T$ **do**
        Sample topic embedding $\alpha^{(t)}$ from eq. 4.15
        Sample time-topic proportion η_t from eq. 4.14
        Sample $L_t \sim \mathrm{LKJChol}(\gamma_t)$
    end
    Choose a minibatch $\mathcal{B}$ of documents
    for *document* d *in minibatch* **do**
        Compute the topic proportion θ_d from eq. 4.12
        for *word* n *in document* d **do**
            Sample the word $w_{d,n}$ from eq.4.22
        end
    end
    Estimate ELBO loss $\mathrm{L}_{\mathrm{ELBO}}$ from Eq. 4.26
    Compute Transformer loss $\mathrm{L}_{\mathrm{CrossEntropy}}$
    Compute the unbiased gradient estimate
    Compute the stochastic gradient via backpropagation
    Take a stochastic gradient step with Adam
    Update the model and variational parameters
end

Algorithm The procedure for the model training is described in algorithm 7. To begin with, the parameters are initialized. For each epochs $1, \ldots, N$, the topic embedding $\alpha^{(t)}$ in every single time stamp t are computed. To perform stochastic variational inference, we divide the data set into smaller data batch $\mathcal{B}$. For each document d in $\mathcal{B}$, we sample time information η_{t_d} from 4.14 to the specific time stamp t_d the document d belongs to. Then we compute the topic proportion θ_d. For each word position n in the document, a word is then to be drawn as $w_{d,n}$

The ELBO loss is being computed by the sum of the document-topic proportion part from the equation 4.18 and the time-topic proportion part from the equation 4.24. Following that, the transformer loss is computed by cross-entropy error, To optimize the model, we compute an unbiased gradient estimate from the model The procedure continues repeating until the maximum iteration is reached.

4.5 Experiment and results

In this chapter, Dynamic Correlated Topic Model(DCTM)[41], and Dynamic Embedded Topic Model(DETM)[13] to compare with our model.[1]

4.5.1 Experiment settings

Datasets We select the UN DEBATES as one of the testing corpora for the experiment. It is a collection of transcripts from the official of UN member countries expressing the government's perspective over the world issues at the time. After preprocessing, It contains 46 years time span of data, with 7507 documents and 6831 tokens in total. We selected 6005 for training, 1402 documents for testing and 100 documents for validation. Second dataset we selected is NeurIPS conference paper dataset. The dataset contains conference papers ranging from 1987-2019. After preprocessing, it contains 9677 papers in total and 9182 tokens. Within the dataset, we pick 7345 documents for training, 1737 for testing, and 100 for validation.

Data pre-processing To prepare the datasets for training, we pre-process the documents and turn them into a useful corpus. We remove the special characters and stop words, and perform tokenization to split document sentences into a list of tokens. To train the model, we have to leverage the datasets to feed-in different models. Specifically, the data are shaped into different forms. First, the bag-of-word $w_{1:D} \in \mathbb{R}^{D \times V}$ is a matrix consist of the word count for vocabularies that exist in every document. The document frequency for tokens is set to 100 documents minimum and 50% at max. We also created a time-vocabularies word count matrix for the training of η. The normalized bag-of-words for time-to-words distribution $\tilde{w}_{1:T} \in \mathbb{R}^{T \times V}$ holds the word count for the vocabulary set over the time span $t = 1, \dots, T$.

Transformer learning task See 3.5.2.

Models We maintain the default settings for DCTM [41], with 0.1 kernel bandwidth and 1 for kernel amplitude.[2] For D-ETM model, we follow the default settings instructed in [13]. We set the variance of on the prior to 0.005.[3]

Algorithm configurations Following the parameter settings from [7], the variance of prior in are set to $\xi^2 = 0.005$ on $\alpha \sim \mathcal{N}(\cdot, \cdot)$. We set the dimension for the transformer embedding to be 256, and so for the hidden dimension for both transformer encoder and decoder. Each transformer encoder and transformer decoder consist of two layers and 2 heads for scaled dot-product. The sequence length of the transformer is set to 20. For the gaussian process latent

[1]DTM was tested in our experiment, nonetheless, did not yield result after 10 hours of running time.

[2]`https://github.com/spotify-research/dctm/`

[3]`https://github.com/adjidieng/DETM/`

variable model, we selected zero mean with a squared exponential function (RBF kernel) as the covariance prior, which allows providing adaptive change on topic-proportion against the possible rapid topic changes from documents. The number of inducing points is set to 50. We follow the setting [41] and set the length scale of the kernel as 0.1.

4.5.2 Results

Training In the training stage, we perform black-box variational inference to estimate the unbiased gradient estimator with Monte Carlo sampling for intractable variational lower bound.

Quantitative results The result of the models is to compare in terms of perplexity, Topic Coherence (TC), Topic Diversity (TD). To calculate the topic coherence and topic diversity, we average down the scores over time span $1 \dots T$. We put the models into UN debates and NIPS dataset for comparison.

In addition to both of the results above, we also put the model we construct from chapter 3 to compare with the time-series version models in this chapter. Since the model does not consider time-series information, it only predicts a single set of topic words and thus for the TC and TD score.

	Perplexity	TC	TD
DCTM	4205.1	-0.017	**0.953**
D-ETM	**2842.1**	0.076	0.427
Our model(ch3)	2746.3	0.152	0.643
Our model	4103.8	**0.092**	0.429

Table 4.1: Result on UN debates dataset, k=30

	Perplexity	TC	TD
DCTM	4898.4	-0.092	**0.966**
D-ETM	**2282.3**	0.120	0.517
Our model(ch3)	1960.8	0.201	0.733
Our model	5461.0	**0.136**	0.502

Table 4.2: Result on NeurIPS dataset (1987-2019), k=30

From table 4.1, we have compared our model with the latest model. We only highlight the times-series model with the highest score. On the UN debate dataset, we conduct the experiment with the number of topics $k = 30$. In terms of perplexity, the D-ETM model has performed a better score. On the other hand, our model obtains a better performance in terms of topic coherence and topic diversity, which are 0.092 and 0.429 respectively. In particular, the UN debates dataset tends to contain more diversified topic mentions in different years and word usages by leaders from different countries. The DCTM holds the best topic diversity value, but the poor result in perplexity and topic coherence score make it meaningless. On the NIPS dataset, table 4.2 contains the scores that we obtained from each model we conducted an experiment. From the result, when $k = 30$, our model resulted from 0.136 in topic coherence score and 0.502 in topic diversity. To compare with other instances, our model performs the best in topic coherence score. D-ETM has the lowest perplexity score, which performs the best in the predictive task. On the other hand, our model has apparently beaten the other models in both TC and TD scores.

Qualitative results

UN debates We put the document-topic proportion obtained from the model and run the t-SNE algorithm to transform the document-topic proportion into 2-dimensional continuous space. As shown in the figure 4.1, different colors of dots represent a specific topic. As can be seen, apparently the documents can be classified into interpretable clusters according to their topic.

From the result we obtained, we select the top-6 topics have the highest proportion to the document set and display them in figure 4.2. And for those topics, we extract the top-5 words having the highest proportion. Then we track those words how they change their proportion to the corresponding topic. As demonstrated, the topic words in each topic selected don't diverse each other very much in proportion, resulting in a coherent trend that the word is more likely to adhere to the topic.

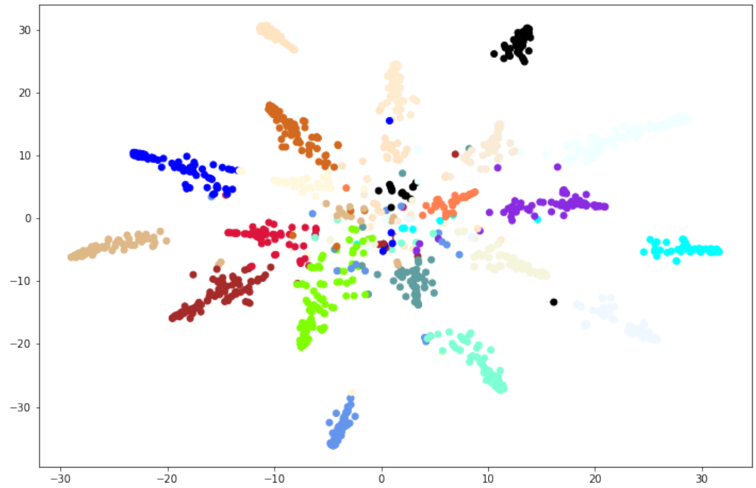

Figure 4.1: t-SNE visualization for documents labeled by topic(UN debates)

Accordingly, in the figure 4.3 we also track the change of those topics over time. Different color in the cumulative graph represents the topics that evolve along the period.

NeurIPS dataset To investigate the word trends change over time, table 4.3 visualize the words by 5 years interval. In particular, we selected a topic corresponding to reinforcement learning and pick the top-10 words for each timestamp. By observation, we see that the topic are coherent in several keywords like CONTROL, ACTION and STATE. On the other hand, some other keywords also highlight their importance upon specific time period. For instance, words like MARKOV, POLICY and DISCOUNT are more likely to appear before; word REINFORCEMENT appears since 2012. For sake of interpretability, we also highlight the first appearance in the year for those keywords related to reinforcement learning. On figure4.4, displays the word trend from top-6 topics over the time. We have selected 3 representative tokens from the top-10 words for each topic. And observe how the words trend through the years. In figure 4.5, we provide a stack plot

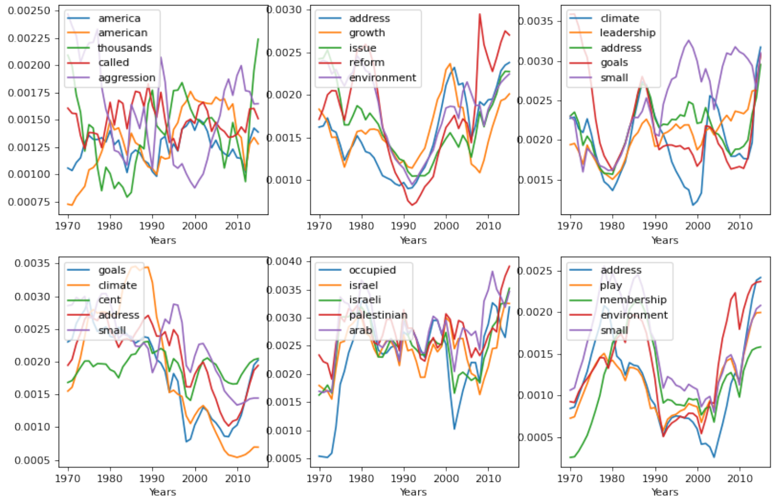

Figure 4.2: Word trend for top-6 topics (UN debates)

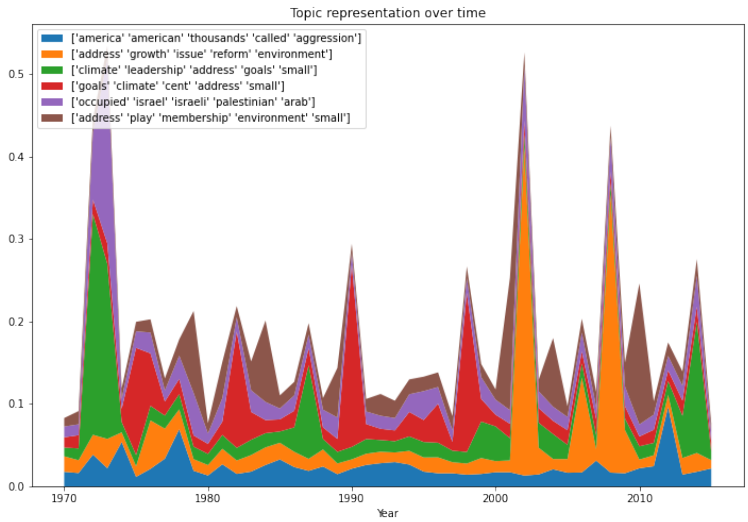

Figure 4.3: The topic trend for top-6 topics (UN debate)

for how those top-6 topics changed over time. Apparently, it demonstrates how the topics are inclining or declining. For example, the topic related to "reinforcement" is gaining more popularity over time. Besides, the topic of "cortex, cells" is being less important over the years.

4.5.3 Discussion

In the result, we observe that our model has been outperforming the D-ETM model in TC and TD scores. Apparently, the topic words in our model generated are both consistent and coherent over time. On the other hand, we notice that our model does not beat the state-of-the-art model in perplexity. We speculate that the model trade-off the loss from the training transformer brings down the

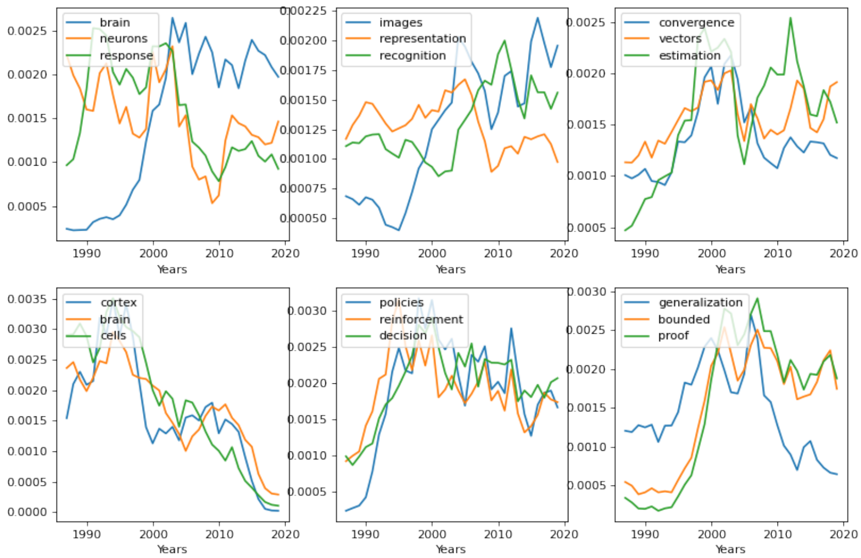

Figure 4.4: Word trend for top-6 topics (NeurIPS dataset)

Year	Topic: Reinforcement Learning
-	policy reward action goal control agent states actions reinforcement search
1987	control position world simulated initial search change environment modification action
1992	temporal watkins cambridge dynamic controller sutton states control actions action
1997	states actions markov decision control dynamic discount action discounted policy
2002	transition decision dynamic reward markov policies states actions policy action
2007	artificial intelligence rewards policies programming states action reward actions policy
2012	reinforcement decision markov states mdp action policies reward actions policy
2017	action markov policies reinforcement transition expected control states policy reward

Table 4.3: Word trend in topic reinforcement learning (5 years interval)

predictive performance of the model. From analyzing the trend of the word and word group from each topic, we investigate how well the words obtain from each topic evolve. We convince that our model can obtain coherent topics from the document set and retrieve diverse keywords during the time span.

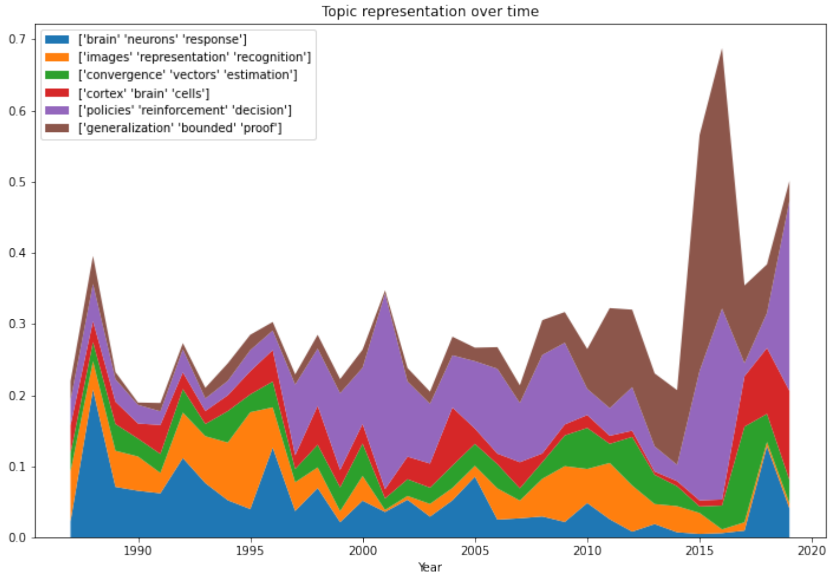

Figure 4.5: The topic trend for top-6 topics (NeurIPS dataset)

Chapter 5

Conclusions

In this thesis, we have proposed a state-of-the-art topic model, Transformer Embedded Correlated Topic Model(TECTM), and with its extension for time-series data, Dynamic Transformer Embedded Correlated Topic Model(DTECTM). The result has shown TECTM a better performance in returning high quality topics compared with other state-of-the-art models. Our model also demonstrates a capability in classifying substantial number of topics.

We also expanded the model to handle time-series data. The model was integrated with Gaussian process latent variable model, which make able the model to capture the time-series information from document set. We also expose our model has a competitive performance compared to other state of the art models.

In the studies though our thesis, we carried a series of experiments with varies of metrics to validate the models. Yet our model is capable to obtain a high quality topics in terms of topic coherence and topic diversity. However, we still notice that DTECTM model does not produce a good enough perplexity score compared with other models. So it is expected to improve the predictivity of the model as a future work.

Their are more improvement to the models. First, Nonparametric Bayesian method did not consider in the research due to the time limitation. Also, since graph model and document shares similarity on power law, we expect to explore the possibility to use graph machine learning knowledge to work with the model.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *arXiv:1607.06450 [cs, stat]*, 2016.

[2] John Barnard, Robert McCulloch, and Xiao-Li Meng. Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. pp. 1281–1311, 2000. Publisher: JSTOR.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. Vol. 3, pp. 1137–1155, 2003.

[4] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. Vol. 20, No. 28, pp. 1–6, 2019.

[5] David M. Blei. Probabilistic topic models. Vol. 55, No. 4, pp. 77–84, 2012.

[6] David M. Blei and Michael I. Jordan. Variational inference for dirichlet process mixtures. Vol. 1, No. 1, pp. 121–143, 2006. International Society for Bayesian Analysis.

[7] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pp. 113–120. Association for Computing Machinery, 2006.

[8] David M. Blei and John D. Lafferty. A correlated topic model of science. Vol. 1, No. 1, pp. 17–35, 2007. Institute of Mathematical Statistics.

[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. Vol. 3, pp. 993–1022, 2003.

[10] Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of topic models. Vol. 11, No. 2, pp. 143–296, 2017.

[11] Jonathan Chang and David Blei. Relational topic models for document networks. In *Artificial intelligence and statistics*, pp. 81–88. PMLR, 2009.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

[13] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. The dynamic embedded topic model. In *arXiv:1907.05545 [cs, stat]*, 2019.

[14] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. Topic modeling in embedding spaces. In *arXiv:1907.04907 [cs, stat]*, 2019.

[15] Nick. Nicholas Charles. Raff Edward Eren, E. Maksim. Solovyev. COVID-19 literature clustering. 2020. event-place: University of Maryland Baltimore County (UMBC), Baltimore, MD, USA.

[16] Junxian He, Zhiting Hu, Taylor Berg-Kirkpatrick, Ying Huang, and Eric P. Xing. Efficient correlated topic modeling with topic embedding. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 225–233. ACM, 2017.

[17] Philipp Hennig, David Stern, Ralf Herbrich, and Thore Graepel. Kernel topic models. In *Artificial Intelligence and Statistics*, pp. 511–519. PMLR, 2012. ISSN: 1938-7228.

[18] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. 2013.

[19] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. Vol. 14, No. 5, 2013.

[20] Thomas Hofmann. Probabilistic latent semantic analysis. In *arXiv:1301.6705 [cs, stat]*, 2013.

[21] Donghwa Kim, Deokseong Seo, Suhyoun Cho, and Pilsung Kang. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and doc2vec. Vol. 477, pp. 15–29, 2019.

[22] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *arXiv:1312.6114 [cs, stat]*, 2014.

[23] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Vol. 104, No. 2, p. 211, 1997. American Psychological Association.

[24] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, Vol. 2, p. 5. Citeseer, 2003.

[25] Neil D. Lawrence. Learning for larger datasets with the gaussian process latent variable model. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pp. 243–250. PMLR, 2007. ISSN: 1938-7228.

[26] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. Vol. 401, No. 6755, pp. 788–791, 1999. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 6755 Primary_atype: Research Publisher: Nature Publishing Group.

[27] Daniel Lewandowski, Dorota Kurowicka, and Harry Joe. Generating random correlation matrices based on vines and extended onion method. Vol. 100, No. 9, pp. 1989–2001, 2009.

[28] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. In *arXiv:1907.11692 [cs]*, 2019.

[29] Jon D. Mcauliffe and David M. Blei. Supervised topic models. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pp. 121–128. Curran Associates, Inc., 2008.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[31] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 262–272. Association for Computational Linguistics, 2011.

[32] Christopher E. Moody. Mixing dirichlet topic models and word embeddings to make lda2vec. In *arXiv:1605.02019 [cs]*, 2016.

[33] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pp. 100–108, 2010.

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

[35] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *arXiv:1802.05365 [cs]*, 2018.

[36] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. 2005.

[37] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. In *arXiv:1703.01488 [stat]*, 2017.

[38] Yee Whye Teh. A tutorial on dirichlet processes and hierarchical dirichlet processes. 2007.

[39] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. Vol. 101, No. 476, pp. 1566–1581, 2006.

[40] Michalis Titsias and Neil D Lawrence. Bayesian gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 844–851. JMLR Workshop and Conference Proceedings, 2010.

[41] Federico Tomasi, Praveen Chandar, Gal Levy-Fix, Mounia Lalmas-Roelleke, and Zhenwen Dai. Stochastic variational inference for dynamic correlated topic models. In *Conference on Uncertainty in Artificial Intelligence*, pp. 859–868. PMLR, 2020.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, {\textbackslash}Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[43] Hanna M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pp. 977–984. ACM Press, 2006.

[44] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. ST-SAGE: A spatial-temporal sparse additive generative model for spatial item recommendation. Vol. 8, No. 3, pp. 48:1–48:25, 2017.

[45] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 697–702, 2007. ISSN: 2374-8486.

[46] Ke Xu, Yi Cai, Huaqing Min, Xushen Zheng, Haoran Xie, and Tak-Lam Wong. UIS-LDA: a user recommendation based on social connections and interests of users in uni-directional social networks. In *Proceedings of the International Conference on Web Intelligence*, pp. 260–265. ACM, 2017.

[47] Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. A correlated topic model using word embeddings. In *IJCAI*, pp. 4207–4213, 2017.

[48] Liang Yang, Fan Wu, Junhua Gu, Chuan Wang, Xiaochun Cao, Di Jin, and Yuanfang Guo. Graph attention topic modeling network. In *Proceedings of The Web Conference 2020*, pp. 144–154. ACM, 2020.

[49] 岩田具治. トピックモデル. 講談社, 2015.