Latent Semantic Topics Distribution Discovery Over Web Content Corpus

Zhengyuan Ding Rui Jiang Ruijie Chen Danfeng Li

Center for Data Science, New York University
{zd415, rj1407, rc3959, dl3983}@nyu.edu

Abstract

Topic models are widely applied to provide automated analysis of unclassified online web content in various business settings. This paper makes use of web browsing text content provided by Bombora and presented the results of implementing and evaluating several topic models for comparison. We first employed a Latent Dirichlet Allocation (LDA), a popular and very effective generative probabilistic model, as our baseline. We then experimented with a Hierarchical Dirichlet Process (HDP) model which serves as an extension of LDA to further infer the optimal number and the structure of topics given the whole corpus. In addition, we implemented Correlated Topic Modeling (CTM) in order to further relax the assumption of topics independency to discover correlations between such topics. We also adapted a more recent Variational Auto-Encoding Bayes inference to combine advantages of both LDA and deep learning technique. With our new framework, while the computational cost remains stablem, the perplexity has decreased dramatically. In the end, we evaluated the performance of all above stated models on 20NewsGroups, a human labeled benchmark dataset, with evaluation metrics as perplexity, model's complexity, and topic coherence. The advantages and drawbacks of each model are discussed based on the evaluation.

[2019 Fall] NYU DS-GA 1006 Capstone Project
Code available at: `https://github.com/Rae-Jiang/Capstone-Topic-Modeling`.

1 Introduction

Large ever-growing web content corpus has been increasingly valuable for generating business insights in any field. The volume and noise in those web content make it difficult for human to manually browse and manage. Therefore, automated analysis of underlying structure and corpus topics becomes important.

Bombora, as the leading provider of the leading provider of organization intent data for business to business (B2B) marketers, ingests and processes a billion daily events that are tied to companies and business professionals across the globe. With thousands of topics labels and human labeled data of web contents, Bombora is able to capture the topic related trends in a daily basis.

In this project, rather than modeling the problem as a classification task, where each topic is a single class, we would like to discover the underlying latent semantics topics distribution with unsupervised methods using bombora's immense data. This approach allows us to find additional information from web content corpus without assuming certain topics labels. Considering the changing nature of web content, the unsupervised method should be a better fit for our purpose of knowledge discovery.

One of the advantage of latent variable topic models is that the result often provides interpretable and useful structures and even enables meaningful construction of topic graphs. In addition, the essential statistical structure found by topic modeling can be later used for other task such as classification, novelty detection, and summarization.

In this project, we aimed to provide a topic modeling tool that serves as an extension of Bombora's current modeling tools. For better practical application, our goal is to formulate a model with great representation power and scalibility. We experimented with both traditional statistical models and more advanced deep learning models. Models implemented in this project includes LDA, HDP, CTM, and VAE. In the end, we evaluate each model with perplexity (PPL), coherence score, and computational efficiency combined with human evaluation.

1.1 Literature Review

Topic models are widely used in exploratory analysis of large collection of unstructured text and other discrete data. It is a convenient way to conduct text mining and information retrieval with unclassified text. Web based libraries can use topic models to recommend books based on users' past readings. News providers can use topic modelling to understand articles efficiently or cluster similar articles and provide digest easily.

Topic models originated from the goal of dimensionality reduction of the document-term matrix for large corpora. In 1990, Deerwester et al introduces Latent semantic indexing (LSI), which utilized singular value decomposition (SVD) as a technique to reduce the dimension of matrix.[1] The resulting matrix are the latent factors that captures the semantic structures within the documents. Later, latent semantic analysis (LSA) is proposed to replace the raw word count with term-frequency inverse-document frequencies (TF-IDF) weightings. However, SVD is based on Gaussian noise assumption, and therefore the latent factor is not statistically well-defined. To solve this issue, Hofmann presented Probabilistic latent semantic indexing (pLSI) by adding a probabilistic component to LSA model. [2] It is essentially a generative model that assumes each word is generated from certain word distribution of a topic.

Based on pLSI, latent Dirichlet allocation (LDA) added a layer of document level probabilistic model. With this approach, each word is generated by the joint-probability of two mixture components. [3] Documents are a mixture of topics, while topics are a mixture of words. In the past ten years, there is an increasing demand of automated technique for documents analysis. Therefore, LDA becomes very popular as a completely unsupervised method for topic discovery. There are tens of LDA based models. In particular, Hierarchical Dirichlet Process (HDP) is introduced as an extension of LDA to find the optimal number of topics that best models the underlying structure of given corpus, since the topic number is usually unknown.

However, with the assumption of topic independence, LDA could be limited when applied to highly correlated documents. The idea of correlated topic modeling (CTM) was then proposed by Blei et al to overcome the limitation.[4] They made use of logistic normal distribution that captures correlations between documents. In particular, the covariance matrix found by CTM can be used to form a topic graph, which is a useful feature in practice.

In additional to the traditional models, researchers have been actively exploring deep neural network for topic modeling recently. This includes the use of Variational Autoencoders (VAE) and generative adversarial network (GAN). The generative nature of these models aligns well with LDA model. One drawback of deep learning method is the longer training time and instability during the training, which limited its application in industry. Therefore, computational efficiency is addressed during the evaluation.

Methods for evaluation and interpretation are essential to any machine learning tasks including topic modeling. For predictive topic modeling, researchers aims to build a model to predict future document, in which case perplexity measures are important. For the purpose of knowledge discovery, it is ultimately necessary to perform human evaluation for interpretability. In reality, it is often a trade-off between predictibility and interpretability.[3] Therefore, semantic coherence is proposed in 2011 to balance those two aspects.[5] In this project, we make use of perplexity, coherence and human evaluation to evaluate our models.

1.2 Problem Statement

In this project, we are given a corpus that consists of documents from various web source. Each document is represented by the vectors of word counts in a high dimensional space, which is also known as "bag of words". This method easily transforms documents into fixed length list of numbers, but neglected the order of word. This assumption of exchangeability for the words greatly improves computational efficiency and is widely used in the field of topic modeling.

1.2.1 Notation and terminology

We define basic terms that will be used throughout this paper.

- **Corpus:** A corpus C is a collection of D documents $C = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_D\}$.
- **Documents:** A document is represented by N words $\mathbf{w}_d = (w_1, w_2, \cdots, w_N)$.
- **Words:** A word w_n is the basic unit, which comes from a vocabulary indexed by $\{1, 2, \cdots, V\}$.
- **Topics:** Assume there are K topics in the corpus. An optimal k can be found during evaluation. In LDA based models, topics are denoted by β.
- **Latent Variables:** z denotes vector of latent variables. In LDA based models, $z_{d,n}$ represents the topic assignment of dth document and nth word.

1.3 Problem Formulation

Topic models are essentially generative. In statistical models, it specifies a probabilistic procedure to generate representations of the documents. Our goal is to improve the generative method such that the topic distribution of generated data is similar to that of the real documents.

A simple generative procedure would be: [2]

- Create a new document by choosing a distribution over topics.
- Each word in that document could choose a topic at random depends on the distribution.
- Draw a word from that topic.

Based on this simple structure, various statistical topic models are implemented by taking different assumption on the distribution, which is discussed in the next section.

2 Model Architecture

In this section, we will first explain the theory of most popular statistical topic models we experimented on and then and present the framework and illustrate the deep learning techniques we applied to improve these models, mainly autoencoding variational Bayes based inference for LDA and a generative adversarial approach to topic modeling.

2.1 Statistical Models

In this section, we will discuss three statistical topic models deployed in this paper. First is the most common one in topic modelling field, while the other two build on LDA and make some improvements.

2.1.1 LDA

Among all the statistical approach to unsupervised topic modeling, Latent Dirichlet Allocation proposed by Prof. David M. Blei(2003) is perhaps the most representative one and of most significance.[6] LDA is a generative probabilistic model that can display topics of each document as a probability distribution. It views documents as bags of words(BOW) and assumes that each document is a mixture of a set of topics and that each word's presence is attributable to one of the document's topics. It overcomes the shortcomings of another widely used topic model, pLSI(Probabilistic latent semantic indexing) (Hofmann, 1999) by adding two Dirichlet prior for topic distribution θ_i and word distribution $\varphi_{z_{d,n}}$ respectively.[7] While pLSI uses a large set of individual parameters which are explicitly linked to the training set, LDA treats the topic mixture weights as k-parameter *hidden random variable* so that LDA can generalize easily to new documents.

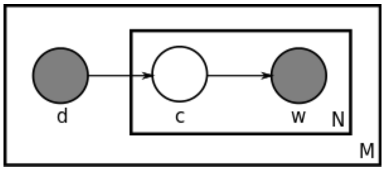


Figure 1: Comparison: pLSI/PLSA

Assumptions and generative process

The key assumptions that are made to simplify the model are: 1. the dimensionality k of the Dirichlet distribution (and thus the dimension of the topic variable z) is priorly assumed and fixed; 2. the word probabilities are parameterized by a $k \times V$ matrix β where $\beta = p(wj = 1|zi = 1)$; 3. N is independent of all the other data generating variables(θ and z). With N as number of words in a document, α and β as corpus-level parameters, $z_{d,n}$ as topic for the word w in document d and $w_{d,n}$ as the specific word w, LDA assumes the following mathematical form of generative process for each N-word document d in a corpus C:

1. Select a document $p(d_i)$according to the prior probability
2. Sample from the Dirichlet distribution $Dir(\alpha)$ to generate the topic distribution θ_{d_i}, $\theta_{d_i} \sim$ D $ir(\alpha)$ where $i \in \{1, \ldots, D\}$
3. Choose word distribution $\varphi_k \sim \mathrm{Dir}(\beta)$ for topic $z_{d,n}$, where $k \in \{1, \ldots, K\}$
4. For each word in document d:
   (a) Choose topic $z_{d,n} \sim \mathrm{Mult}(\theta_n)$.
   (b) Choose word $w_{d,n} \sim \mathrm{Mult}(\varphi_{z_{d,n}})$.

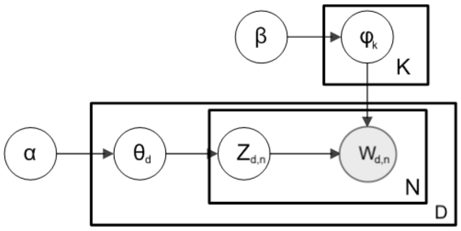


Figure 2: Graphical model representation of LDA.

Model representation

LDA model's structure is as shown in Figure 2: the boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document. Different from a classical simple Dirichlet-multinomial clustering model, LDA involves 3 levels: a Dirichlet is sampled once for a corpus, so that documents can be associated with multiple topics.

Inference

Though key inferential problem for LDA which is computing the posterior distribution of the hidden variables given a document is intractable:

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{1}$$

A wide variety of approximate inference algorithms could be applied to approximate it including Sampling(Gibbs Sampling), variational approximation and Markov chain Monte Carlo(MCMC). Here due to space limitations, we would like to briefly introduce variational inference. The main idea is actually the same as what we will present in Variational Autoencoder(VAE) part: instead of computing the intractable posterior $p(H|D)$, we approximate with a tractable distribution $q(H|D, V)$, which is from a family of simpler distributions defined by a set of free variational parameters V. Thus the goal is to find parameters V which minimize the KL divergence $KL(q(H|D, V)||p(H|D))$ to the true posterior.

$$D_{KL}(P||Q) = \sum P(x) log \frac{P(x)}{Q(x)} \tag{2}$$

Thus the optimization problem becomes as follows where γ, ϕ, λ are free variational parameters we approximate θ, z, β with, respectively:

$$\gamma^*, \phi^*, \lambda^* = argmin_{\gamma,\phi,\lambda} D_{KL}(q(\theta, z, w|\gamma, \phi, \lambda)||p(\theta, z, w|\alpha, \beta) \tag{3}$$

To find the best $\gamma^*, \phi^*, \lambda^*$, we only need to iteratively solve the optimization problem until the solution converges.

2.1.2 HDP

HDP, Hierarchical Dirichlet Process is an extension to LDA and their major difference is that HDP doesn't require the specification of number of topics.[8] Therefore, compared to LDA, HDP is useful in the case when the number of topics is unknown. It has the advantage that the maximum number of topics can be unbounded and learnt from the data rather than specified in advance.

In terms of modelling structure, HDP is similar to LDA, except that it also uses a Dirichlet process as a base distribution to simulate the uncertainty in the number of topics. The structure is shown in figure 3. First a base distribution G_0 that shared among all groups is generated from a Dirichlet process via a base parameter H and a concentration parameter γ. Then the kth group topic assignment G_k is independently sampled for each topic condition on G_0 and a concentration parameter α. G_k then produce the topic distribution $\theta_{d,n}$ for the dth document and nth word to generate each word $w_{n,d}$. Thus, the main extension of HDP on LDA is attains from G_0 to generate G_ks that represent the finite set of possible topics for the whole text collection.

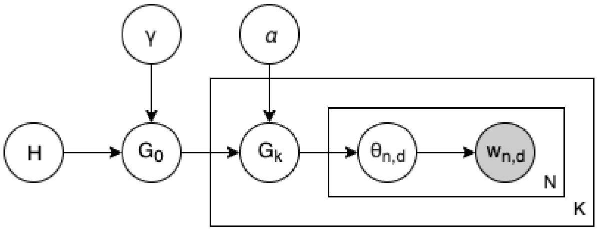


Figure 3: HDP Structure

In general, the process can be expressed as follow:

1. Choose $G_0|\gamma, H \sim DP(\gamma, H)$.
2. Choose $G_k|\alpha_0, G_0 \sim DP(\alpha_0, G_0)$ for each topic $1 \ldots K$.
3. For each word in document d:
    (a) Choose $\theta_{n,d} \sim G_k$
    (b) Choose $W_{n,d} \sim Mult(\theta_{n,d})$.

2.1.3 CTM

The limitation of LDA is mainly in its assumption of independence between topics, which derives from the near independent assumption in the Dirichlet distribution prior to model the topic proportions. This is not usually the case in real life, since there are often correlations between different topics. For instance, an article about science in genetics is very likely to infer topics like disease and health. To deal with this limited assumption in LDA, a correlated topic model(CTM) is proposed by Blei and Lafferty [4], which models the correlation between the latent topics given a text collection. The correlation between topics is the key improvement of CTM over LDA and it is introduced by drawing topic proportions from a logistic normal distribution,instead of a Dirichlet distribution as in LDA.

Specifically, the model structure is shown in figure 4. K-dimension variance Σ and mean μ are used to generate topic distribution per document η_d. With η_d, the model will assign a topic to each word in a document, namely $Z_{d,n}$ for the nth word and dth document. After choosing these parameters, we draw a word $W_{d,n}$ given the chosen topic $\beta_{Z_{d,n}}$ among K topics from the right box of the figure. The key difference here between CTM and LDA stems from the covariance matrix Σ, which contains the relationship between topics.

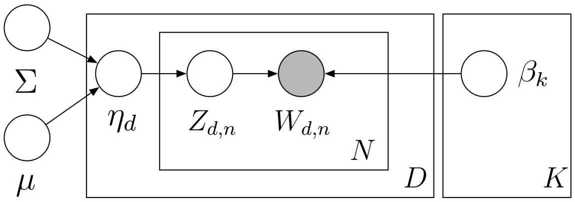


Figure 4: CTM Structure.

The mathematical form of the modeling process for an N-word document d can be described as:

1. Choose prior $\eta_d|\{\mu, \Sigma\} \sim \mathcal{N}(\mu, \Sigma)$
2. For each word in document d:
    (a) Choose topic assignment $Z_{d,n}|\eta_d$ from $Mult(f(\eta_d))$.
    (b) Choose word $W_{d,n}|\{z_{d,n}, \beta_{1:K}\}$ from $Mult(\beta_{z_{d,n}})$

, where $f(\eta_i) = \exp \frac{\eta_i}{\sum_j \exp \eta_j}$ and $\eta = \log(\frac{\theta_i}{\theta_K})$.

Comparing with LDA, CTM has many advantages. The key contribution of CTM, as mentioned above, is the use of logistic normal distributions to model relations among topics. This means two important improvements. First, words are allowed to co-occur in different topics. Second, it is possible to build topic graphs to form topic relations using the covariance matrix from the logistic normal.

Despite the many benefits brought by the more flexible assumption of topic relations, the disadvantage of CTM compared to LDA is its relatively low computation efficiency. [2] This problem stems from the logistic normal distribution as well. It is mainly because that logistic normal is not conjugate to the multinomial. This raises the cost during the posterior inference approximation process.

2.2 Deep Neural Network Models

2.2.1 VAE based Inference

As mentioned in LDA part, since the major issue for developing a new topic model is the computational cost of computing the posterior distribution, most research in topic modeling considered approximate inference methods, among which variational methods like mean field methods and Markov chain Monte Carlo(MCMC) are most popular ones. But both above mentioned methods have drawbacks that when making a small change to model assumptions, they require re-deriving the inference methods which is time-consuming. Therefore black-box inference methods especially Autoencoding Variational Bayes(AEVB)[9] is a natural choice for topic models, because it trains an inference network, a neural network that directly maps a document to an approximate posterior distribution. Different from the optimization problem stated in LDA, AEVB presented another 'variational lower bound' or 'expectation lower bound'(ELBO) as:

$$L(\gamma,\phi|\alpha,\beta) = -D_{KL}[q(\theta,z|\gamma,\phi||p(\theta,z|\alpha] + E_{q(\theta,z|\gamma,\phi)}[logp(w|z,\theta,\alpha,\beta] \tag{4}$$

The above function is the objective function we want to maximize. Similar to mean-field variational inference in LDA, the first term is KL divergence used to approximate true posterior distribution with variational posterior over latent variables. The key here is the second term measuring how good the latent variables at reconstructing the bag of words, which is analogous to the reconstruction loss in autoencoders. Thus our model can be regarded as an improved LDA model under variational autoencoders' framework.

Variational Autoencoders (VAE) is a popular deep learning based generative model aiming to optimize:

$$\mathbb{E}_{X\sim D}[\mathbb{E}_{z\sim Q}[\log P(X)]] - \mathcal{D}[Q(z|X)||P(z)] \tag{5}$$

where X is the datapoint from the dataset D, z is a vector of latent variables that we want to sample according to the probability density function $P(z)$, Q is a probability density function with which we can produce $z^{'}$ that can reproduce X under the $Q(z|X)$, and $\mathcal{D}[Q(z|X)||P(z)]$ is the Kullback-Leibler divergence (KL divergence) between $Q(z|X)$ and $P(z)$:

$$\mathcal{D}[Q(z|X)||P(z)] = \mathbb{E}_{z\sim Q}[\log Q(z|X) - \log P(z)] \tag{6}$$

The structure of VAE is shown in Figure 5, the usual choice of $Q(z|X$ is $\mathcal{N}(z|\mu(X),\Sigma(X))$, where $\mu(X)$ and $\Sigma(X)$ are learned from the data through encoder.

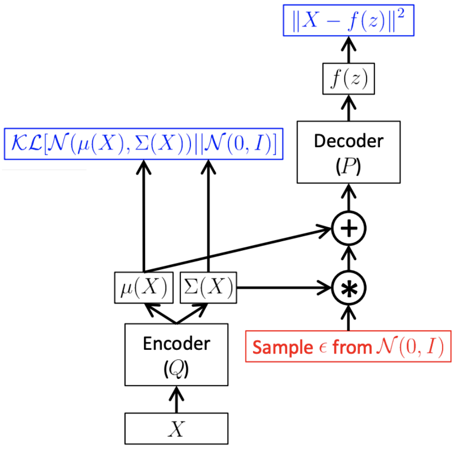

$\|X - f(z)\|^2$
$f(z)$
$\mathcal{KL}[\mathcal{N}(\mu(X),\Sigma(X))||\mathcal{N}(0,I)]$
Decoder
(P)
+
*
$\mu(X)$
$\Sigma(X)$
Encoder
(Q)
Sample ϵ from $\mathcal{N}(0,I)$
X

Figure 5: VAE Structure.

Specifically, in our topic modelling problem, the model takes the bag of words in each document as the input, and projects the input through the encoder into latent vector. Latent vectors are then used to reconstruct each document through the decoder. In this problem we optimize the loss function [10]Once we trained the variational autoencoder, we could retrieve the latent variables which should represent the topic distribution in our problem. Also, the retrieved hidden dimensions for each document can be used to calculate similar documents afterwards.

$$\mathcal{L} = \sum_{d=1}^{D}\{-\frac{1}{2}[tr(\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_0)+(\mu_1-\mu_0)\mathbf{\Sigma}_1^{-1}(\mu_1-\mu_0)-K+\log\frac{|\mathbf{\Sigma}_1|}{|\mathbf{\Sigma}_0|}]+\mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}[\mathbf{w}_d^{\perp}\log(\sigma(\beta)\sigma(\mu_0+\mathbf{\Sigma}_0^{\frac{1}{2}}\epsilon))]\} \tag{7}$$

where μ_0 and $\mathbf{\Sigma}_0$ are the prior mean and variance, μ_1 and $\mathbf{\Sigma}_1$ are the posterior mean and variance from encoder, K is the number of topics, $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_D$ are the documents in the corpus, $\sigma(.)$ is the softmax function, and $\beta = (\beta_1, \beta_2, \cdots, \beta_K)$ is the probability distribution over the vocabulary for each topic.

2.2.2 GAN

GAN is a generative deep learning framework that learns a generator distribution $P_G(x)$ that matches the data distribution $P_{data}(x)$. The generator G is trained to trick the discriminator network D, which aims to distinguish between samplings from real data and generated data. It follows that for any generator, an optimal discriminator should be $D(x) = P_{data}(x)/(P_{data}(x) + P_G(x))$. The training process is given by the following expression:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}}[\log D(x)] + \mathbb{E}_{z\sim noise}[\log(1 - D(G(z)))] \tag{8}$$

The recent work of TopicGAN proposed a structure for text generation and representation learning [11]. The model architecture is inspired by the structure of InfoGAN proposed by Chen et al [12]. The key contribution of InfoGAN is its impressive result in learning disentangle representations without supervision. The model was originally applied to MNIST dataset, which is a hand-written digit image dataset. By specifying the number of classes (10 in this case), the model correctly categorizes each digit into a class.

InfoGAN introduced a new component to GAN architecture, namely Q network, as shown in figure 6. It is designed to maximize the mutual information between generated fake data and the latent code c during the training. This strategy disentangled the use of noise z.

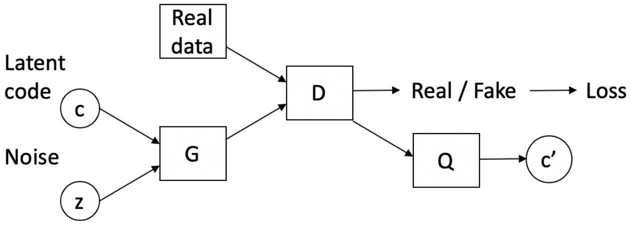


Figure 6: InfoGAN Structure. G denotes generator, D denotes Discriminator, and Q denotes categorical topic code classifier.

Based on the traditional GAN, InfoGAN proposed to solve the following problem with hyperparameter λ:

$$\min_{G,Q} \max_D V_{\text{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \tag{9}$$

where $L_I(G,Q)$ is the variational lower bound of the mutual information $I(c; G(z,c))$. The lower bound can be easy to approximate with Monte Carlo simulation. An complete optimization method can be writen as:

$$\min_{G,Q} \max_D \mathbb{E}_{x\sim P_{data}}[\log D(x)] + \mathbb{E}_{z\sim P_z, c\sim P_c}[\log(1 - D(G(z,c))) - \lambda Q(c|G(z,c))] \tag{10}$$

TopicGAN model adapted a two step progressive generation framework, which is commonly used to facilitate the training process. In the first step, it generates bag-of-words (BOW) that feeds into BOW

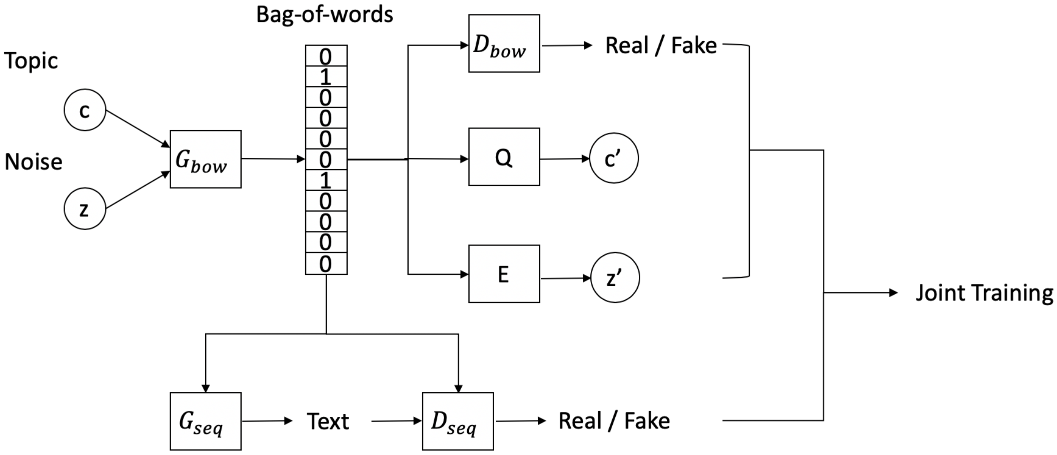


Figure 7: TopicGAN Structure. G denotes generator, D denotes Discriminator, Q denotes Topic model, and E represents the noise predictor.

discriminators to train a BOW generator. In the second step, they trained a separate LSTM generator to construct word sequence given the bag of words embedding learned in the previous step. The joint training of two components yields better result. The major nework components in the first step are:

- Bag-of-words Generator G_{bow} takes the input of discrete one-hot code c and noise input z. It generates bag-of-words with certain topical information.
- Bag-of-words Discriminator D_{bow} distinguishes whether a bag-of-words representation is from real data or generator.
- Topic model Q classifies given bag-of-words input into topic code.
- Noise predictor E predicts the noise that can reconstruct the input bag-of words.

To avoid gradient vanishing problem and mode callapse issue, TopicGAN implemented WGAN loss and clip the probability from topic model Q to a specific range α. Finally, TopicGAN redesigns the objective function and optimizes the network with the following function:

$$\min_{G,Q} \max_{D} \mathbb{E}_{x\sim P_{data}}[D_{bow}(x)] - \mathbb{E}_{z\sim P_z, c\sim P_c}[D_{bow}(G_{bow}(z,c)))] - \lambda \mathbb{E}_{x\sim P_{data}}[\min(Q(c|G_{bow}(z,c),\alpha))] \tag{11}$$

where topic model Q and noise predictor E encode real bag-of-words input into discrete code and continuous noise.

3 Experiments

In this section, we conduct experiments to implement both statistical topic models and neural network based generative models on the Bombora's extracted web content and compare the results using various evaluation metrics. Moreover, to test the robustness of our models, we also experiment models on one classic benchmark dataset in topic modeling field, 20NewsGroups, in Section 4.

3.1 Environment setup

Since our experiments involve using deep Neural network based techniques and we need to tune hundreds of hyperparameters' combination, we mainly utilized Google Cloud platform virtual machine instances with 2 x NVIDIA Tesla P4 GPU for developing Neural network models with Tensorflow 1.14.0 and PyTorch 1.2.0. We also used Google Colab when working on testing sample datasets. The LDA and HDP part were implemented under the MacOS Mojave with Python 3.6.5.

3.2 Dataset and preprocessing

Since our goal is to discover the underlying semantic topic distribution over Bombora's extracted web content corpus, for the major part of our experiments, we used the internet web browsing events dataset provided by Bombora. Bombora's dataset consists of billions of internet web browsing events which contains metrics that reflect both user's web browsing behavior and the text content extracted from virtually all urls in internet. More specifically, it has three features: url, extracted content and timestamp. For this project, we sampled roughly 10,000 records and mainly focus on the text feature, ie, extracted content. After basic cleaning, the entire vocabulary of the corpus covers 300,000 unique tokens.

To give a more concrete overview of our web content corpus, here is a wordcloud generated from the data shown in Figure 8.

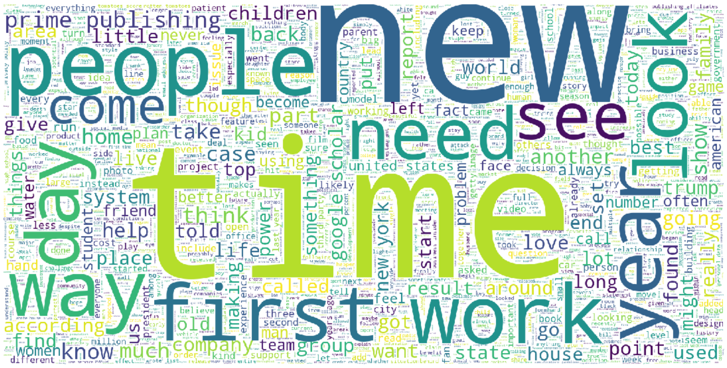


Figure 8: Wordcloud of the dataset.

In terms of preprocessing, we implemented the folllowing steps to clean and build our corpus:

- Tokenization and removal of non UTF-8 characters.
- Stopword Removal. Here we mainly used two mainstream English stopwords resource, one is from *nltk*, another from *spacy*.
- Bigram collocation detection (frequently co-occuring tokens in the corpus) using *gensim Phrases*. This is particularly useful since our corpus contains many such as ‘President Donald’, ‘Amazon prime, ‘new york’,‘Google Scholar’,etc. These bigrams would be more interpretable than the separated form.
- Lemmatization using *nltk WordNetLemmatizer*. It is a process for grouping together different inflected forms of a word so that these words with same meaning are treated as one single word. It is essentially similar to stemming method. Here we prefer lemmatization over stemming, since stemming may reduce the interpretability of words.

A comparison of one sample sentence using different preprocessing methods is listed in Table 1:

Dataset	Tokenized words
Original dataset	*[‘In’, ‘May’, ‘2018’, ‘President’, ‘Donald’, ‘Trump’, ‘awarded’, ‘retired’, ‘Navy’, ‘Master’, ‘Chief’, ‘Britt’, ‘K.’, ‘Slabinski’, ‘the’, ‘Medal’, ‘of’, ‘Honor’, ‘for’, ‘his’, ‘heroic’, ‘actions’]*
Bigram & filter out stopwords	*[‘may’, ‘president_donald’, ‘trump’, ‘awarded’, ‘retired’, ‘navy’, ‘master’, ‘chief’, ‘britt’, ‘slabinski’, ‘medal’, ‘honor’, ‘heroic’, ‘actions’]*
Lemmatize	*[‘president_donald’, ‘trump’, ‘awarded’, ‘retired’, ‘navy’, ‘master’, ‘chief’, ‘britt’, ‘slabinski’, ‘medal’, ‘honor’, ‘heroic’, ‘action’]*

Table 1: Data preprocessing comparison

3.3 Model implementation

We implemented both statistical models Latent Dirchlet Allocation, CTM, HDP, and variational autoencoding Bayes inference based VAE and experimented on sampled Bombora's web extracted corpus on Google Cloud virtual machines and tuned the hyperparameters of each model to get the best performance.

For LDA, we first applied *CountVectorizer* transformation on preprocessed words to create document-term matrix as model's input. We need to check the sparsity of the matrix and materialize into a 2D array. Then we used *GridSearchCV* to tune hyperparameters of LDA: number of topics, prior of document-topic distribution, prior of topic-word distribution, we also include others like learning decay, batch size, etc. Here, number of topics is apparently the most important parameter, We defined a range for it based on our prior knowledge about the dataset and diagnosed the performance using popular evaluation metrics: perplexity, topic coherence and log likelihood.

For CTM, in order to compare the topics and covariances it infers with those of the LDA model, we preprocessed our data in exactly the same way as in the LDA model, and we also set the number of topics to be the same as in LDA. We tuned the hyperparameters such as the prior of document-topic distribution and prior of topic-word distribution and evaluate the performance based on the perplexity and topic coherence.

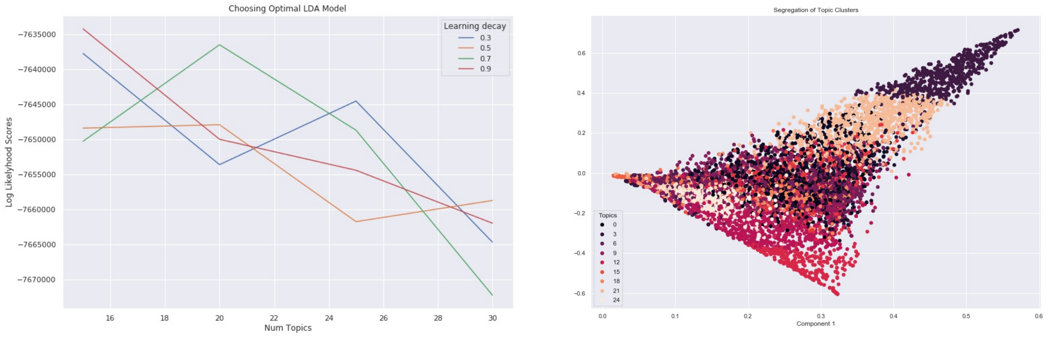
Different from LDA and CTM, HDP is a truly unsupervised model that can determine the optimal number of topics through posterior inference. We utilized *gensim* package in Python, which provides the speed of online variational Bayes with the modeling flexibility of the HDP. Hyperparameters on top and second truncation level were tuned during the training process. We used *CoherenceModel* module to get topic coherence, which serves as the evaluation metric here. In the end, we compared HDP's performance with LDA, LSI.

For VAE, we implemented mainly in *Tensorflow* with a classic variational autoencoder framework: we constructed the encoder through a two-layer neural networks with softplus as the activation function and dropout as the regularization, whose outputs pass through a fully connected layer and thus generate the posterior mean and variance after batch normalization. We can then reparameterize the latent vector, whose dimension here represents the topic distribution. The decoder is a fully connected layer, whose outputs reconstruct the documents after softmax and batch normalization. With the parameters in this structure, we can compute the loss function as illustrated in Section 2.2.1. The hyperparameters we tuned includes the hidden layer sizes, learning rate, batch size, and the vocabulary size. Adam optimizer is used in the model.

3.4 Results

In this section we mainly present the results of training and evaluating LDA, CTM, HDP and VAE model stated above on Bombora's corpus. We experimented using different data preprocessing methods and tuned hyperparameters. Figure 9 mainly shows the hyperparameter tuning result and the segregation of topic clusters after applying SVD decomposition. We found that the optimal number of topics in our dataset should be around 20 which makes sense since our corpus is largely about news and articles involving technology, business, politics, etc. We then visualize the output of best LDA model by criteria of perplexity to check how the documents are clustered into these predicted topics.

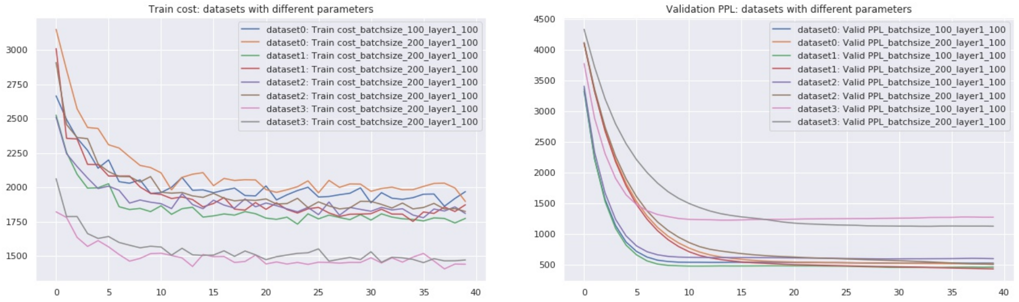
Very interestingly, we noticed while data preprocessing methods such as stopwords, removal, lemmatization and stemming help improve the performance of LDA and HDP, VAE, in contrast, underperforms when applying some data preprocessing techniques especially lemmatization and stemming, we guess it is because stemming, lemmatization, and removing stop words all involve throwing away information, which may not affect simpler model but can do harm to deep learning model. During training, we also found it is prone to be "trapped" to a local optimum closer to prior belief in early time of training. To tackle this problem, we took two measures: training with Adam optimizer using high moment weight and high learning rate. The model performance on training set and test set during training is shown as Figure 10. For most cases, the model converges after 15 epochs of training which only cost roughly 2 minutes on environment with GPU. Larger batch size and higher learning rate help model converge, while different hidden size of encoder layers affect model's prediction.



(a) LDA: Hyperparameters tuning

(b) LDA output: topic clusters

Figure 9: LDA: model performance and results



(a) Train cost: KL divergence and reconstruction loss

(b) Validation perplexity

Figure 10: VAE: model performance of training and validation set

dataset0: after bigram; dataset1: with bigram, *spacy* stopwords removal; dataset2: with bigram, lemmatization; dataset3:with bigram, filtering non English words.

Evaluation metrics The metrics we used to evaluate the performance of our models are mainly perplexity, coherence and Computational efficiency. Perplexity is a quite popular evaluation metric in NLP and can measure how well our models can predict an unseen document. It is defined as the exponential of log likelihood for average word, where D is the number of documents, N_d represents the length of the dth document, $p(X)$ represents log probability of the words in the document:

$$\exp\{-\frac{1}{D}\sum^{N_d}\frac{1}{N_d}\log p(X_d)\} \tag{12}$$

Thus, the lower the perplexity, the better the model can predict on the held-out dataset. However, according to our research, using perplexity might not be the best measure to evaluate how topic models perform since it doesn't consider context and semantic associations between words. In other words, perplexity and human judgment are often not correlated, and even sometimes slightly anti-correlated. Topic coherence, on the other hand is a better representation of human interpretability. A model with lower coherence value usually fails to decipher between similar topics and comes up with topics which might cause confusion to a human. Meanwhile, running time is also an important measurement of a model's complexity. Ideally, the best model should have relatively lower perplexity, higher coherence, and smaller running time. Figure 10a and Figure 10b show the performance of VAE model on the training and validation dataset with different hyperparameters. We also notice that although HDP has a higher perplexity due to its special structure to infer the hierarchy, it has a relatively high coherence value.

	LDA	CTM	HDP	VAE
Perplexity(*PPL*)	2927.87	2317.41	1136.20	433.31
Coherence Value	0.36	0.35	0.38	0.36
Computational Efficiency(s)	157	246	178	193

Table 2: Performance of Models

Table 2 shows our models' performance on Bombora's dataset. We found that with respect to perplexity and coherence, VAE is significantly better than others, while LDA has the best computational efficiency with running time of about 157s.

Suggested topic	Top Words that are Related to the topic
Credit Card	*credit card offer savings time available purchase store date new*
Education	*school team students job work university college student high education*
Healthcare	*health care patients medical aarp org pain drug doctor receive*
Politics	*trump president said 2019 new american country national house june*
Data	*information data use services cookies personal advertising user based*
Jurisdiction	*state law court federal states tax government case rights act*

Table 3: Top Words of Topics in LDA

The LDA model gives us the word distribution for each topic, and there are some word distribution that we can easily refer to some topics. Table 3 shows some topics we guess from the words, as well as the corresponding words that are highly related to the topics and are within the top ten words. However, there are still some topics with more ambiguous word distribution from which we can not guess about the topics.

Similarly, Table 4 shows the topics that we guess in the VAE model. We can see that compared to LDA, VAE may have more abilities to capture the keywords of a topic. This can be the potential reason why VAE outperforms LDA in our datasets. Our VAE model gives the latent variable values for each document, thus we can implement multiple lasso regressions to estimate the adjacency matrix that encodes the conditional independence relations in the multivariate normal distribution. Figure 11 is the topic graph we generated from the regressions. We can find that some topics that are highly correlated are located closely in the graph. For example, in the left side of the graph, there is a cluster whose components are related to the topics of user data, technology, and politics. Also, in the right side, there are some topics that are more related to web subscription and credit card application.

Suggested topic	Top Words that are Related to the topic
Technology	*ai, bluetooth, open_source, sensors, indicators, infrastructure*
User Data	*information_returned, people_search, find_contact, address_phone, authoritative*
Medication	*ckd, vaccine, patients, genes, carbon, outcomes, cells*
Weapons	*bullet, shield, yards, brownells, nra, carbine, glock*
Police	*police, officers, arrest, sheriff, murder, injured, custody, authorities, killed*
Politics	*irs, trump, debt, tax, warren, taxpayers, white_house*
Movie	*met_gala, via_imgur, meghan_markle, singer, photographed, romance, diana*
Marvel	*avengers_endgame, spider_man, thanos, latest_episode, comicbook_nation*

Table 4: Top Words of Topics in VAE

4 Evaluation

Since the Bombora's dataset does not have lables, to better evaluate the quality of the latent topics discovered by the models, we also test them on a human-labeled dataset called 20NewsGroups and compare their performances using C_v topic coherence. Besides, an "eye-balling" evaluation is conducted to compare the top words generated from each topic model.

In terms of the evaluation dataset, 20NewsGroups is a popular one in the field of text-related machine learning applications, especially for text classification and clustering. Therefore it is a reasonable test

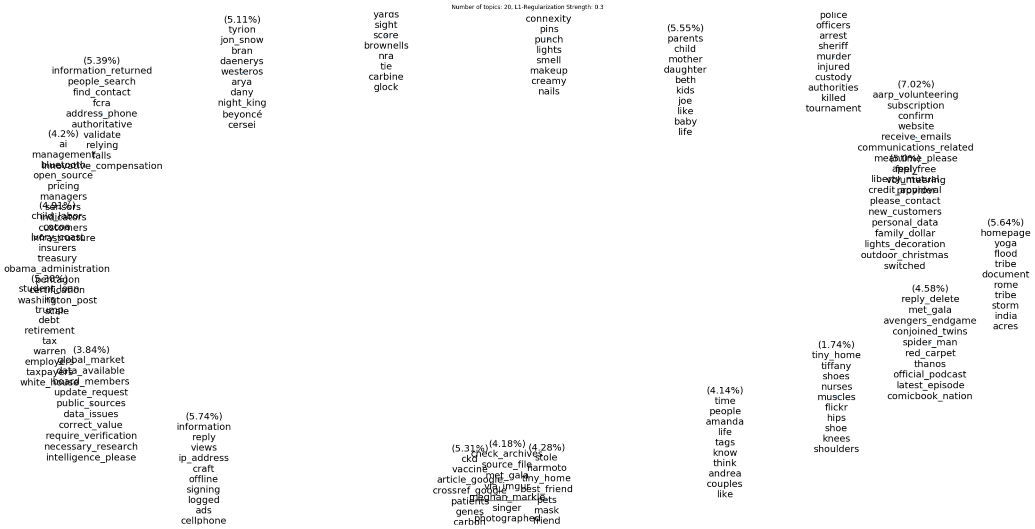


Figure 11: Topics in VAE

set for our topic model comparisons and measurement. The dataset contains approximately 20,000 newsgroup documents and are distributed evenly across 20 different newsgroups (Figure 12) such as computer, religion and sport. Among these documents, 60% is train set and 40% is test set.

For the preprocessing stage, we stay with the same process as the one discussed in Section 3.2. The dataset is tokenized excluding punctuations with stopwords removed and frequently co-occurring bigrams appended. For the simpler statistical models like LDA and HDP, we find that lemmatizing adjtive, verb and noun tokens can boost performance. However, for deep learning model, VAE, lemmatization reduces performance. This is probably becasue deep neural nets can capture the complex of different forms of tokens and lemmatization in fact cut information. Additionally, when building the tokens vocabulary we also remove the least and most common words and reduce it to about 6,000 to 8,000 tokens in total. Specifically, words appearing less than 10 times and words appearing in more than 10% of all documents are removed. During several tunings we find that removing common words helps model distinguish more between different topics.

Topic Coherence Evaluation

The reason why we choose topic coherence as the metrics is that it is shown to be more consistent to human judgement according to Chang [5]. It is surprisingly stated in their paper that predictive likelihood, or equivalently, perplexity and human judgement are usually not correlated. In fact, they are even sometimes slightly anti-correlated. Thus, for the evaluation part on human-labeled dataset, topic coherence will be our major measurement.

Topic coherence provides a score of a single topic by calculating the semantic similarity between the top scoring words in that topic. In other words, a topic is viewed as coherent if the words composing it are able to support each other. There are various types of coherence measurements,however for this paper we choose C_v with default parameter settings (110 sliding window) as the coherence score. To give a brief explanation, this method counts the co-occurrence of the given words in a sliding window and uses these counts to compute the normalized point-wise mutual information (NPMI) with each other top word. Now, each top topic word will have a vector of NPMI scores. Then, an indirect cosine similarity score combined with word segmentation is computed between each top word vector and the sum of the vectors. The final coherence score is the arithmetic average of these cosine similarities. This is better than perplexity in a sense that it is able to detect semantically interpretable topics rather than just artifacts of statistical inference.

For the training part, each model is trained over the training set of the 20NewsGroups and then evaluated over the test set to generate a coherence score. We fixed the number of topics to be 20 for each model in order to align with the number of labels of 20NewsGroups. The topic coherence results over test set are shown in the Table 5. Since CTM costs more than one day to train over this relatively large dataset, we exclude it from the evaluation stage due to its impracticality of implementation. The table displays that VAE scores highest in coherence. However, regarding as efficiency LDA costs the least time. Thus there is a little trade-off between efficiency and coherence when choosing the optimal model.

	LDA	HDP	VAE
Coherence Value	0.488	0.506	0.518
Computational Efficiency	68s	100s	390s

Table 5: Topic coherence on 20NewsGroups test data

Topic Words Evaluation

In addition to standard coherence scores, we also picked some topics and their corresponding top related words for an "eye-balling" evaluation (Table 6). Here we present some examples from LDA and VAE. For each topic we pick related top 10 words and tried to summarize them into corresponding topics, as shown in Figure 12

Generally, the topic words from these models both align with the human-labeled topic given in 20NewsGroups dataset such as computer and religion. An observation of the results is that words generated from LDA and HDP seem to be more general, while ones extracted from VAE are more specific and accurate in the sense of human interpretation. For example, looking at the topic *Computer*, words from VAE can be summarized as hardwares in computer. In contrast, words from LDA can only give an overview of the general computer topic. In this way of evaluation, VAE seems to be able to discover more concrete latent topics.

	Religion	Science	Health	Computer
LDA	god	space	patient	card
	jesus	system	study	system
	church	nasa	cause	do
	believe	mission	food	window
	word	orbit	doctor	run
	bible	earth	pain	disk
	point	also	since	driver
	love	data	disease	mac
	sin	moon	day	pc
	mean	book	case	drive
VAE	jesus	satellite	tobacco	chip
	matthew	space	public health	serial number
	prophecy	nasa	aerospace	fpu
	god	satellites	commercial space	motherboard
	holy spirit	telescope	illness	screen
	isaiah	launch	venture	card
	db	data	space technology	phones
	hanging	infrared	mariner	session key
	israel	observatory	health	monitor
	messiah	spacecraft	russia	clipper

Table 6: Topic words on 20NewsGroups

5 Conclusion and Future Work

The paper mainly demonstrated 3 statistical topic models: LDA, HDP and CTM, and one deep learning model using variational inference based generative topic model. Firstly we experimented the classic statistic topic model,LDA, as our baseline on Bombora's sampled web extracted content. We further improved the model's performance on topic coherence using HDP which serves as an extension of LDA and was designed to address the case where the number of mixture components is not known as priori. As for CTM, it breaks the near independent assumption of topics in LDA and models the topic correlations using logistic normal distribution. However, all these 3 models are under the framework of approximate inference methods to compute the posterior distribution and require high computation cost when making small changes to model assumptions. To deal with this, VAE is proposed to utilize a black-box inference method to directly map a document to an approximate posterior distribution. In this paper we implemented and evaluated these models both on the unlabeled Bombora's dataset and the labeled benchmark dataset 20NewsGroups. As a result, when evaluated on the Bombora's test set, VAE yields the lowest perplexity, which indicates its great predictive power. In terms of computational efficiency, LDA is the optimal and CTM is the worst. Lower computational efficiency could lead to higher computational cost and poorer scaling in practical applications. The evaluation on 20NewsGroups confirms that VAE model scores the highest in coherence score, and there is always a trade-off between efficiency and model coherence.

For future work, we propose to implement the architecture of TopicGAN, which could be effective in learning disentangled latent vectors. Due to the limited time and the instability during training process of GAN, we have not yet successfully trained a TopicGAN in this project. Furthermore, topic evolution is another research area that are worth investigating for the analysis of web content. There are several models such as topic over time (TOT), dynamic topic modeling (DTM), which are commonly used to detect topic evolution in the web of topic.

6 Acknowledgement

We sincerely thank Bombora and our mentor Simon for the guidance, support, time, and resources during the whole semester.

References

[1] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[2] Rubayyi Alghamdi and Khalid Alfalqi. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 6(1), 2015.

[3] Ryan Wesslen. Computer-assisted text analysis for social science: Topic models and beyond. 03 2018.

[4] David M. Blei and John D. Lafferty. Correlated topic models. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pages 147–154, Cambridge, MA, USA, 2005. MIT Press.

[5] Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chiung chu Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[7] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[8] Yee Whye Teh and Gatsby. Hierarchical bayesian nonparametric models with applications . 2008.

[9] Carl Doersch. Tutorial on variational autoencoders. *ArXiv*, abs/1606.05908, 2016.

[10] Akash Srivastava and Charles A. Sutton. Autoencoding variational inference for topic models. In *ICLR*, 2017.

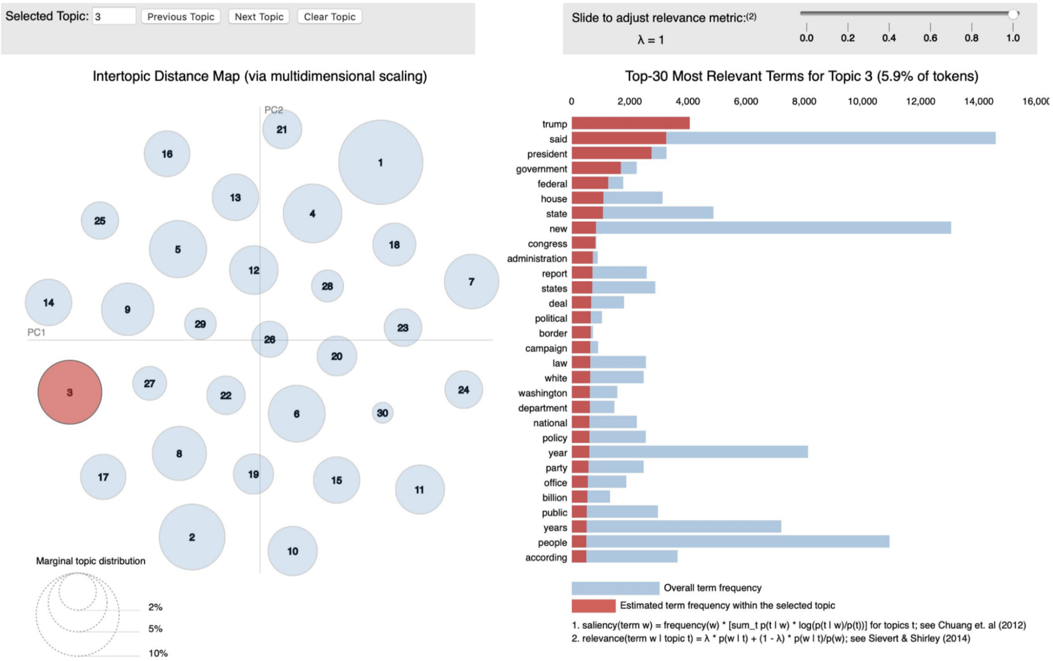
[11] Yau-Shian Wang, Yun-Nung Chen, and Hung-Yi Lee. TopicGAN: Unsupervised text generation from explainable latent topics, 2019.

[12] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2180–2188, USA, 2016. Curran Associates Inc.

7 Appendix

```
alt.atheism
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
sci.crypt
sci.electronics
sci.med
sci.space
soc.religion.christian
talk.politics.guns
talk.politics.mideast
talk.politics.misc
talk.religion.misc
```

Figure 12: 20NewsGroups Topics



Selected Topic: 3

Slide to adjust relevance metric:(2)

λ = 1

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 13: LDA output topic visualization