COMP4432 Group Project Report

LI Jinlin ZHANG Caiqi ZHANG Yubo

June 14, 2022

1 Introduction

In this project, we leverage the "A Million News Headlines" dataset to implement three natural language processing (NLP) related tasks: **topic modeling, sentiment analysis and news generation.** Our motivation is that since natural language understanding (NLU) and natural language generation (NLG) are two main fields under NLP, we aim at explore them both using the provided a million news headlines.

The remainder of this report is structured as follows. Section 2 presents some basic statistics of the dataset. Section 3 introduces how we preprocess the data. In Section 4, we apply two baseline models and two neural network based models to do the topic modeling. Section 5 and Section are about the sentiment analysis and news generation tasks conducted in the news headlines dataset.

2 Exploratory Data Analysis

# of headlines	1,226,258
# of headline tokens in vocabulary:	106,758
avg. length of headlines	41.153 characters, 6.538 tokens
range of publish date	2003/02/19-2020/12/31

Table 1: Data statistics.

The statistics of the dataset downloaded from Kaggle[1] are reported in Table 1. The dataset consists of 1226258 pieces of news headlines. Figure 1 shows the top 15 most frequent words in the dataset excluding the non-sense stop words (e.g. a, an, the).

[1] `https://www.kaggle.com/datasets/therohk/million-headlines`

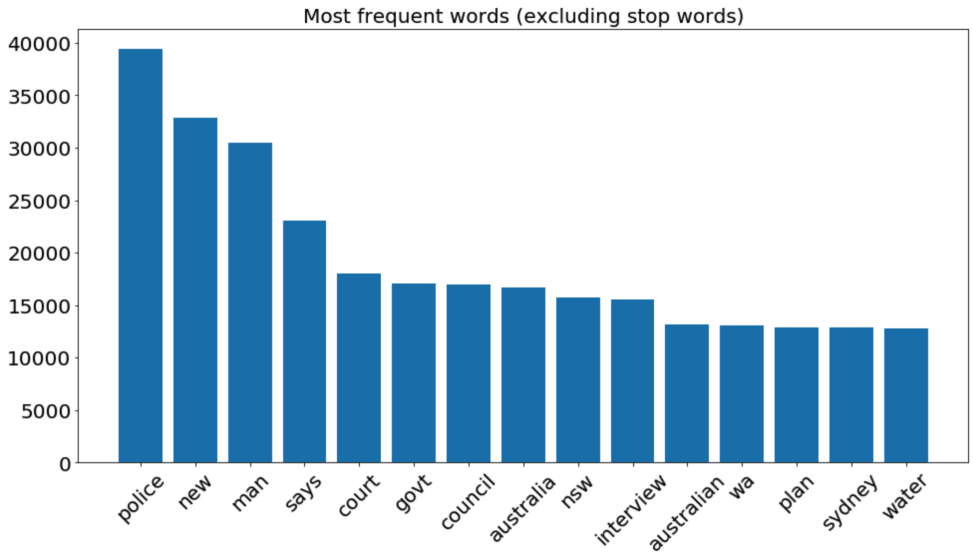


Figure 1: Word Counts

3 Data Preprocessing

We preprocess the data in the following ways:

- Deduplication: Out of more than one million data, some of them are duplicates. For the duplicate data we directly delete them, leaving only the first data.
- Stemming: Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. We apply stemming to the words to get more accurate results in topic modeling step.
- Remove stop words: A stop word is a commonly used word (such as "the", "a", "an", "in") that we ignore in the topic modeling part, since they do not have concrete meanings.

Table 2 shows some statistics after data preprocessing.

# of headlines	1,195,191
# of headline tokens in vocabulary:	78,280
avg. length of headlines	33.292 characters, 5.493 tokens
range of publish date	2003/02/19-2020/12/31

Table 2: Data statistics after data preprocessing.

4 Topic Modeling

Topic modeling is a process to cluster the topics of a group of documents. Unlike topic classification that is trained with labeled dataset, topic modeling is an unsupervised learning task that cluster texts and modeling topics among clusters.

The following mathematically illustrate general process of topic modeling task for our dataset:

1. Draw local topic proportion θ^i_{local} of each headline h_i with $\theta^i_{local} \sim Cat\,(h_i)$. With θ_{local} or other embedding methods, get embeddings e_i for headline h_i.

2. With all headlines embeddings e, clustering headlines h into K topic clusters

3. Denote each topic cluster as c_k for $k = 1, \ldots, K$, then draw global topic proportion $\theta^k_{global} \sim Cat\,(c_k)$.

4. Finally, for each cluster c_k, choose its top N global topics based on θ^k_{global}. For each headline $h_i \in c_k$, choose its own top M local topics based on θ^i_{local}.

Here, Cat $(\cdot)$ denotes the categorical distribution. In this report, we use "**LOCAL** topics" to refer topics within a single headline and "**GLOBAL** topics" referring its corresponding cluster's topics considering all headlines in the dataset.

There are many methods to model the topic. We will firstly introduce the baseline algorithm, **LSA** and **LDA**. Then we further introduce two neural topic models i.e., **NVDM-GSM** and **BERT** and compare their results.

4.1 Baselines: LSA, LDA

4.1.1 CountVectorizer

Both LSA and LDA use CountVectorizer to preprocess the data. CountVectorizer will separate the headline into words and generate the dictionary. For each headline, the CountVectorizer will get the one hot array for it according to the dictionary. This preprocessing step is necessary for LSA and LDA to convert the sentence to mathematical representation.

4.1.2 LSA

Latent Semantic Analysis (LSA) is one of the most frequent topic modeling methods analysts make use of(1). LSA is based on distributional hypothesis, which means the semantic meaning of a word can be discovered by the sentence around it. Based on this theory, LSA aims to calculate the frequency of each document in terms of words and compare its distribution. LSA uses tf-idf

methods to calculate the word frequency, which will be detailed discussed later.
Once the tf-idf frequency is calculated, LSA will generate the document frequency matrices, that for each document, it lists the corresponding frequency of all words. The document matrices can be decomposed into three matrices(USV) by singular value decomposition (SVD). Then LSA will automatically operate those matrices and gain the result.

4.1.3 LDA

LDA is also based on the theory of distributional hypothesis. It will map the document in the corpus to a set of topic that can cover most meaning in the document. LDA algorithm assumes that the document is generated by two elements, the main topic and recipe. Besides, LDA believes in that the document will confront to Dirichlet distribution. The output of the algorithm is a vector that contains the coverage of every topic for the document being modeled.

4.1.4 Result

The experiment sampled $100,000$ data from data set, then we apply LSA and LDA to get the result, that 9 topics cluster.
Firstly, the LSA gives the result:

Topic Cluster	**Top 5 words**	**# of headlines**
1	police crash death car killed	13000
2	new year laws zealand life	4000
3	man charged murder dies guilty	4500
4	says minister pm mp labor	11000
5	govt australian plan water health	45000
6	court accused face case told	3000
7	australia day world cup win	9000
8	council plan city considers land	1000
9	interview abc news home speaks	4800
10	nsw coast gold country rural	3000

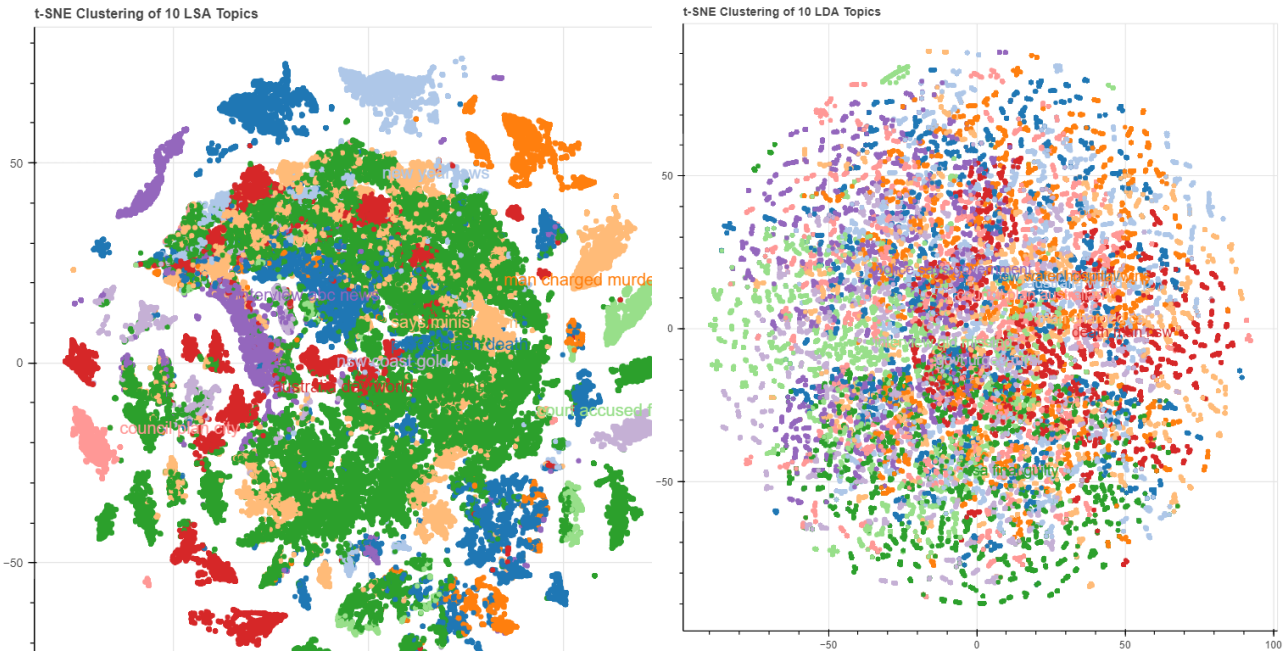
Table 3: LSA

Secondly, the LDA result is:

Topic Cluster	Top 5 words	# of headlines
1	new school country china test	12000
2	australia world cup day report	10000
3	water hospital wins help ban	10500
4	man charged car woman dies	8800
5	sa final guilty deal minister	8700
6	interview qld missing election rise	8600
7	death man nsw sydney murder	9000
8	council plan australian wa abc	8500
9	police says government man drug	10700
10	govt urged rural pm trump	9800

Table 4: LDA

In order to see the result clearly, the scattered map is shown below:



LSA Plot LDA Plot

Figure 2: It can be found that the result of LSA has a huge variation in terms of the size in cluster. By contrast, LDA has a flatten distribution among each cluster.

4.2 Neural Topic Modes (NTM)

4.2.1 NVDM-GSM (VAE + Gaussian Softmax)

NVDM-GSM is a Neural Topic Model proposed in this paper: "Discovering Discrete Latent Topics with Neural Variational Inference"(2). Its architecture is a simple VAE, which takes the BoW of a document (i.e., headlines in our dataset) as input. After sampling the latent vector z from the variational distribution $Q(z|x)$, the model will normalize z through a Gaussian Softmax layer. More details about VAE and Gaussian Softmax will be explained in the following content.

Variational Autoencoder (VAE) VAE is a kind of reconstruction autoencoder with the architecture shown in Figure 3. To be more specific, the model receives x as input and x is sampled from a parametrized distribution. The encoder compresses it into the latent space as a latent vector z. The decoder receives z as input the information sampled from the latent space and produces x' as similar as possible to x. Encoder and decoder of VAE are trained jointly and can be optimized by minimizes a reconstruction error in the sense of the Kullback–Leibler divergence between the parametric posterior (output of VAE) and the true posterior.

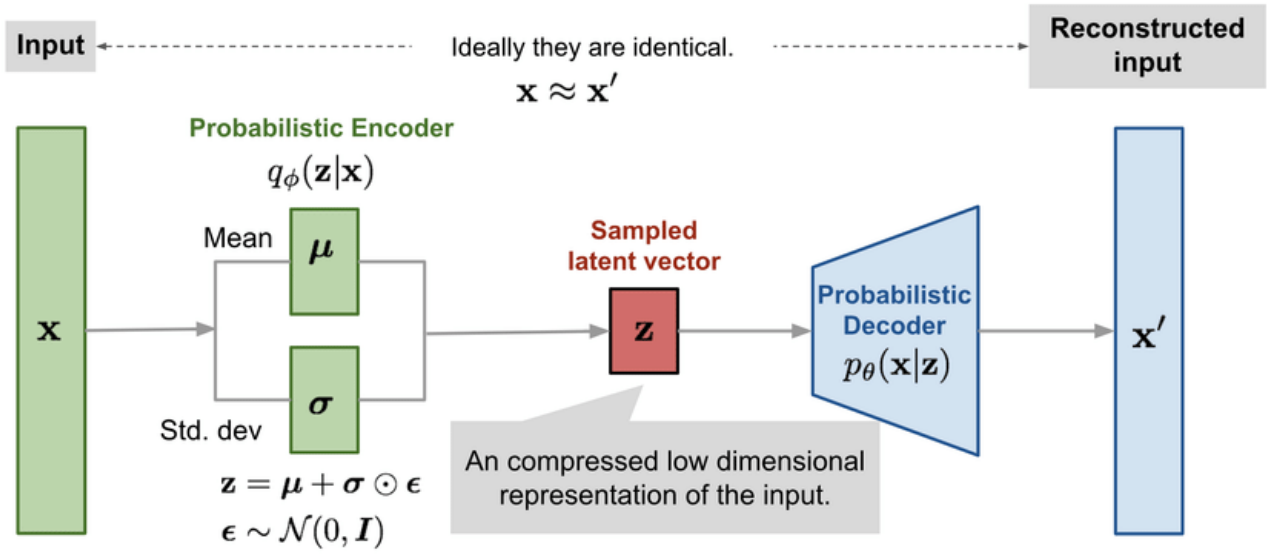


Figure 3: VAE model, figure source: (3)

Gaussian Softmax In NVDM-GSM, Gaussian Softmax distribution (GSM) aims to project output of VAE to probabilities of all available topics, which idea is common in NTM that an energy-based function is generally used to construct probability distributions. (2). NVDM-GSM passes a Gaussian random vector through a softmax function to parameterise the topic

distributions. Thus $\theta \sim G_{\text{GSM}}(\mu_0, \sigma_0^2)$ is defined as:

$$x \sim \mathcal{N}(\mu_0, \sigma_0^2)$$
$$\theta = softmax\left(W^T x\right)$$

where W is a linear transformation. μ_0 and σ_0^2 are hyper-parameters which we set for a zero mean and unit variance Gaussian.

Results For the implementation of NVDM-GSM, we use codes from a NTM tool(4) that implement NVDM-GSM via Pytorch. The results reported below are obtained through this tool on our dataset.

As recommended in NVDM-GSM papaer, we trained NVDM with 100 epochs on a Tesla T4 GPU and the following Figure 4 is the loss figure.

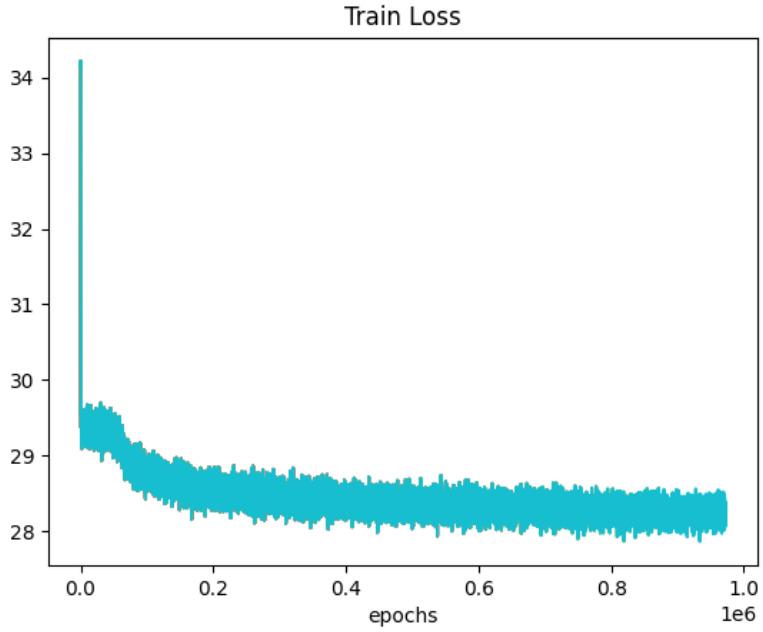


Figure 4: Training Loss for 100 epochs (1000 batch iteration per epoch). Loss function is Cross Entropy

The following Table 5 shows the top 10 topic clusters predicted by GSM and the top 5 words for each cluster. Observing from the table, there are some duplicate words among top 10 topic clusters, and one possible explanation is that VAE is more sensitive to words with high frequency in the dataset and the length of headline are too short for VAE to learn semantic interpretation

instead of word frequency. Thus, we will introduce BERT to improve the representation learning in next section.

Topic Cluster Index	**Top 5 words**
1	interview, fine, bid, team, club
2	abc, club, fine, race, target
3	bushfir, toll, threaten, spark, park
4	studi, park, bodi, question, expect
5	fall, question, bank, expect, bodi
6	bushfir, park, spark, threaten, toll
7	fall, question, bank, bodi, station
8	fall, question, bank, bodi, run
9	question, fall, bodi, run, station
10	sri, lanka, lankan, india, vs

Table 5: NVDM-GSM Predict Topics

4.2.2 BERT Topic Models

Bidirectional Encoder Representations from Transformers (BERT) (5) is a transformer-based (as shown in Figure 5) for NLP pre-training developed by Google. BERT is pretrained on a large text corpus, and then use that model to solve downstream NLP tasks. BERT outperforms previous methods because it is the first unsupervised, deeply bidirectional system for pre-training NLP. BERTopic(6), a topic model that extends BERT by extracting coherent topic representation through a class-based variation of TF-IDF. BERTopic topic modeling pipeline is:

1. Document Embeddings
2. Document Clustering
3. Topic Representation

Document Embeddings Sentence-Transformer framework (7) allows users to convert sentences and paragraphs to dense vector representations using pre-trained language models. Our selected embedding model is: SentenceTransformer('all-distilroberta-v1') i.e., a pretrained RoBERTa model(8)(RoBERTa is similar to BERT but with optimized settings). Note that we use RoBERTa to get offline embeddings without finetuning on our dataset as it has been well pre-trained and has zero-shot/transferring learning ability.

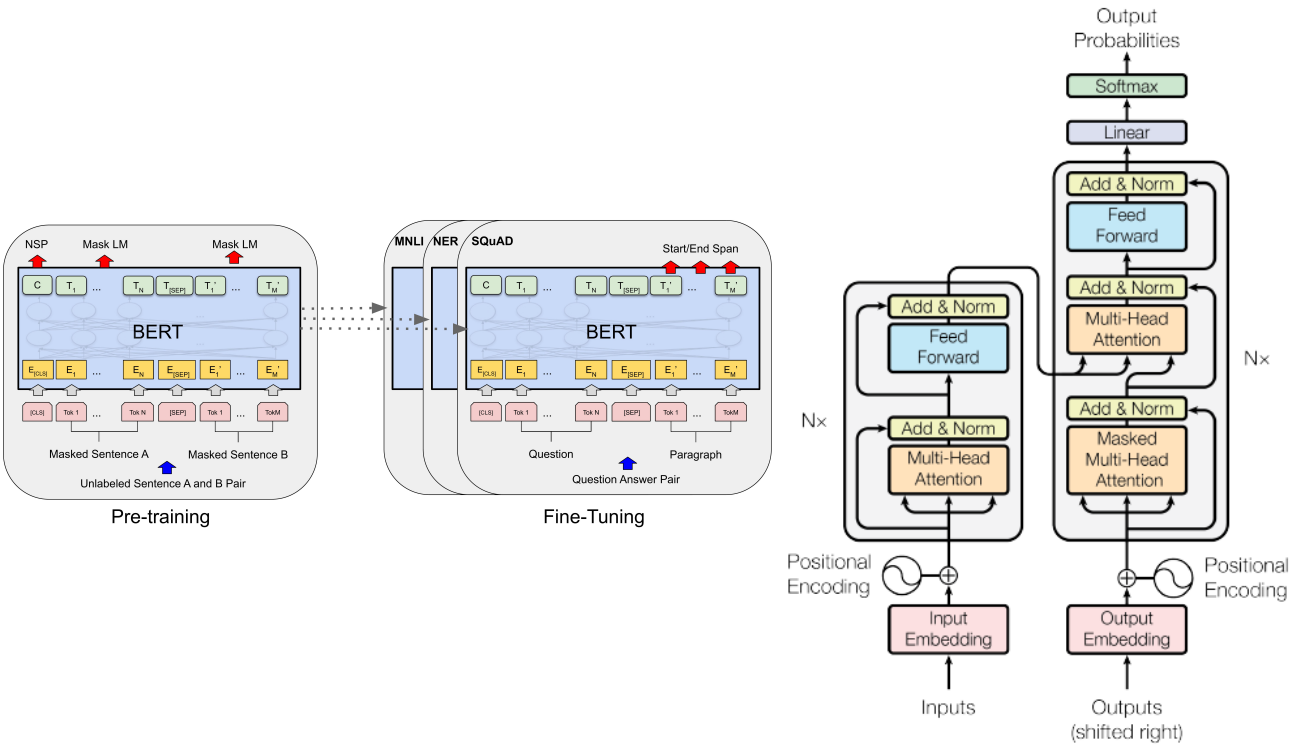


Figure 5: BERT and Transformer Architecture (BERT can be regarded as the encoder of Transformer)

Document Clustering

1. Reduce the dimensionality of embeddings: UMAP (9) preservers more of the local and global features of high-dimensional data in lower projected dimensions. Moreover, it has no computational restrictions on embedding dimensions.

2. Clustering: HDBSCAN (10) is a hierarchical clustering algorithm that finds clusters of varying densities. It models clusters using a soft-clustering approach allowing noise to be modeled as outliers. HDBSCAN works quite well with UMAP since UMAP maintains a lot of local structure even in lower-dimensional space.

Topic Representation The topic representations are modeled based on the headlines in each topic cluster. For each cluster, to know what makes this cluster based on its word distribution, we apply Class-based TF-IDF (c-TF-IDF), a modified TD-IDF measure allows for a representation of a term's importance to a cluster.

$$W_{t,c} = tf_{t,c} \cdot \log\left(1 + \frac{A}{tf_t}\right)$$

Where $w_{t,c}$ models the frequency of term t in a cluster c. Here, the cluster c is the collection of headlines concatenated into a single document for each cluster. Then, the inverse document frequency is replaced by the inverse cluster frequency to measure how much information a term provides to a cluster. It is calculated by taking the logarithm of the average number of words per cluster A divided by the frequency of term t across all clusters. Adding one to the division within the logarithm makes output only positive values.

Thus, this c-TF-IDF procedure models the importance of words in clusters instead of individual headlines. This allows us to generate topic-word distributions for each cluster of headlines. Besides, by choosing top k words with highest $W_{t,c}$, we can reduce the number of topics.

Results The following Table 6 reports the predicted Topic Clusters via BERTopic.

Topic Cluster	**# of headlines**	**Name**
-1	67080	-1_assault_murder_guilti_aussi
0	5255	0_flood_floodwat_flash_mitig
1	3983	1_adjourn_suprem_appeal_court
2	3642	2_fund_shortfal_rda_infrastructur
3	3571	3_korea_jong_korean_kim
4	3466	4_reject_govt_statehood_govern
5	3403	5_rail_derail_train_freight
6	3349	6_drought_stricken_assist_aid
7	3282	7_boat_ship_cruis_capsiz
8	2646	8_sharemarket_market_gain_share
9	2581	9_socceroo_argentina_uruguay_brazil

Table 6: Top10 most frequent topics (-1 indicates refers to all outliers which do not have a topic assigned.) Name format is based on BERTopic i.e., the top4 GLOBAL topics are separated by underline in name of each Topic Cluster.

In Figure 6, we also visualize the top5 words for Top10 Topic Clusters with c-TF-IDF scores. Insights can be gained from the relative c-TF-IDF scores between and within topics.

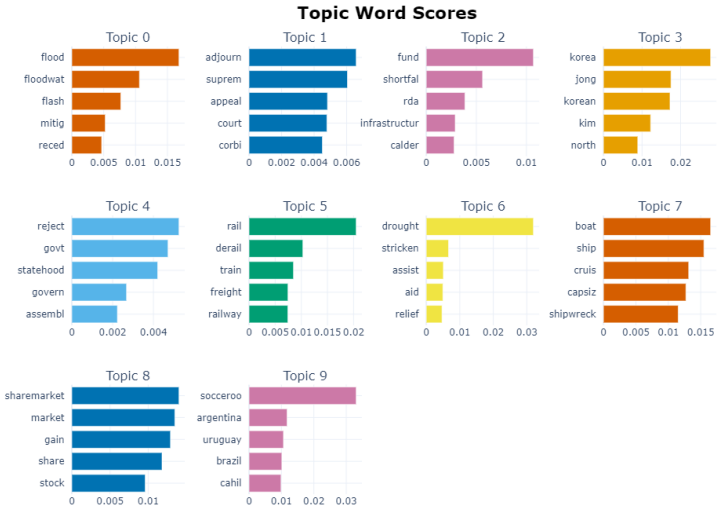


Figure 6: Top5 words and c-TF-IDF in Top10 topic clusters

The following Figure 7 shows the similarity scores among different clusters. We can observe that the top10 Topic Clusters have lower similarity which reflects a good clustering result.

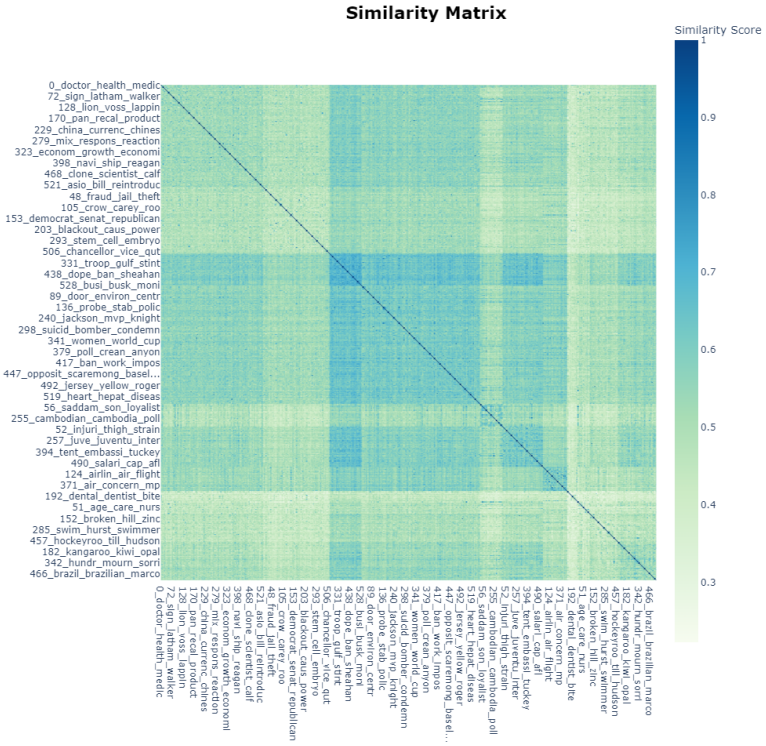


Figure 7: Similarity Matrix Among Clusters

4.3 Evaluation

To evaluate performance of the topic models, we adopt two widely-used metrics, i.e., topic coherence (TC) and topic diversity (TD). For each topic model, its topic coherence was evaluated using normalized pointwise mutual information (NPMI) (11). This coherence measure has been shown to emulate human judgment with reasonable performance. The measure ranges from [-1, 1] where 1 indicates a perfect association. As defined by Dieng(12), topic diversity is the percentage of unique words for all topics. The measure ranges from [0, 1] where 0 indicates redundant topics and 1 indicates more varied topics.

Table 7 reports the comparison results across different models.

Models	TC	TD
Baselines		
LSA	0.11	0.19
LDA	0.16	0.24
Neural Topic Models		
NVDM-GSM	0.17	0.26
BERT	**0.28**	**0.33**

Table 7: Topic Coherence (TC) and Topic Diversity (TD) comparison among models ranging from Top 10 topic clusters.

4.4 Case Study

In case study part, we choose 2 headlines and reveal the result according to different topic model.

Headline	**Model**	**Local Topics**	**Global Topics (Top 3)**
Act fire witnesses must be aware of defamation	**LSA**	fire, defamation	court, accused, face
	LDA	fire	murder, court, police
	NVDM-GSM	fire, act, aware	fire, guilty, court
	BERT	fire, witness, defamation	fire, criminal, abuse
Actors vie for divine role	**LSA**	actors	interview, abc, news
	LDA	divine	interview, qld, missing
	NVDM-GSM	actors, role	actors, show, movie
	BERT	actor, vie, role	actor, director, stage show

Table 8: Predicted Topics of 2 selected headlines. (Predicted topics are separated by;)

From the above cases, we may highlight the following observations:

- Under prior knowledge of human evaluation, the above two case studies can make sense of the result of Topic Coherence and Topic Diversity reported in Table 7.
- Compared with other models, BERT captures both local and global topics from context more semantically and syntactically (e.g., actor with vie, witness with defamation), which may because of multi-head attention mechanism in BERT.
- With n-gram c-TF-IDF's help, BERT can also concatenate sub-words into a meaningful topic. e.g., stage show instead of show.
- LSA and LDA lacks the ability to separate the topics precisely according to the semantics

5 Sentiment Analysis

Strictly speaking, news headlines state objective facts and should not carry the personal opinions of the authors or media organizations. However, in real life, we inevitably include the author's personal opinion in the headlines. Therefore, we conducted a sentiment analysis to explore the distribution of sentiment in these headlines.

To compare the results from different models, we employ three sentiment analysis models from different categories. 1) RoBERTa: This model is trained on RoBERTa large with the binary classification setting of the Stanford Sentiment Treebank. It achieves 95.11% accuracy on the test set. 2) LSTM: This model trained a LSTM binary classifier with GloVe embeddings. 3) NLTK: A classic NLP library which uses rule based sentiment analysis method. Notice that the first two models do not have the neutral category originally, so we manually set the thresholds. The results can be found in Figure 9.

	RoBERTa	LSTM	NLTK
Positive	0.359	**0.511**	0.130
Negative	**0.674**	0.181	0.145
Neutral	0.028	0.073	**0.899**

Table 9: Sentiment analysis results

From the results, we make the following findings and conclusions:

- Before the experiment, we thought that the news headlines should be neutral, but this is not the case from the results. There are times when the news uses some tendentious verbs.

For example, "U.S. forces launch military operations in Iraq" and "U.S. forces invade Iraq" may indicate the same thing, but they are expressed in different ways and thus carry the subjective emotions of the authors.

- One of the reasons has to do with the training set. The models we applied are already trained. Since both RoBERTa and LSTM are trained on data sets with strong sentiment, they are not good enough to judge neutral texts. This is also a common problem brought by the direct application of other models.

- Although the experimental results of NLTK are in line with our expectation, it has some doubts. For example, since NLTK is a rule-based model, it tends to classify news headlines as neutral for some news headlines that do not involve feature words. NLTK's Vader sentiment analysis tool uses a bag of words approach (a lookup table of positive and negative words) with some simple heuristics (e.g. increasing the intensity of the sentiment if some words like " really", "so" or "a bit" are present).

6 News Generation

In this section, we discuss another task i.e. news generation that expand previous topic modeling. News generation is a popular topic in NLG, and it aims to automatically generate news content with given topics or other initial context. GPT-2 is one of the well pre-trained models that can applied to solve this task. Compared to BERT (usually for NLU task) introduced in topic modeling, GPT-2 can be regarded as the encoder part of transformer and usually used for NLG task. GPT-2 is also a sequence-to-sequence model, for our news generation task, the input can be a headline and the output is the corresponding generated news content. **To extend usage of previous topic modeling result, we can use the predicted topics as prefix prompt added to headline to control output news within a limited topic area.** The following is an example showing a prefix prompt that is input to GPT-2:

"Given topics: [x1] [x2]...[xn] generate news for headline [y]."

The design of prompt can be various based on the pre-training tasks for NLG models. In this report, we will follow the same prefix prompt structure as above example. Because the dataset does not contain news contents, we leave the evaluation of generation with metric e.g., ROUGE to future work, and we use human evaluation to get intuitive analysis by comparing generated news with given topics from different topic models.

The following list shows several generated news for headline "Actors vie for divine role" (2nd case study) with respect to different given topics in prefix prompt.

- Without given topics: We can predict next year's election as a result of a major increase in the cost of public health...
- With BERT predicted topics: coverage by a variety of voices, including The [movie name]'s [actor name...]! The [movie name] has been running a new series, and the new series has a new cast...
- With NVDM-GSM predicted topics: with a clear purpose and are often the ultimate goal of any actor to create a believable world...
- With LDA predicted topics: the ones that are most likely to be popular with the masses...
- With LDA predicted topics: This is not the first time in the history of the genre that the roles...

Without any given topics, it is quite meaningless for generation on such short headline. We can also find the generated news is more meaningful and easier to understand when given topics generated by BERT than other given topics, which may enlighten news writers on how to write news conditioned on the specified topic.

7 Conclusion and Future Work

In this project, we pratice three NLP tasks using the "A Million News Headlines" dataset i.e., topic modeling, sentiment analysis and news generation. For topic modeling part, we not only implement the baseline models (LSA and LDA), but also propose to use Neural Topic Model like NVDM-GSM and BERT (with best performance). We also give comprehensive evaluation among models. In sentiment analysis part, we also conduct experiments to compare the performance of different models in our dataset. Finally, we attempt to apply the GPT2 model to generate news content automatically conditioned topics.

The future works are as follows: 1) For topic modeling, the dataset has no labels so that we can not finetune our best model i.e., BERT. It is better to first finetune a BERT on such topic modeling task. Besides, because both embedding and clustering of BERTopic require high computational ability, it is quite time consuming for the largest dataset. 2) In the sentiment analysis part, we only apply existing built in models instead of training a new model. This can be improved by design a new model especially for the news headlines. 3) For news generation task, although we can control generation by adding prefix prompt with topics, there are still improvement space for generation performance if given topics can be utilized with multiple prompt instead of a single prefix prompt.

References

[1] F. Pascual, "Topic modeling: An introduction,"

[2] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 2410–2419, PMLR, 06–11 Aug 2017.

[3] C. Wang, Y. Richardson, and R. Sander, "Unsupervised image clustering and topic modeling for accelerated annotation," 12 2019.

[4] L. Zhang, "Neural topic models." `https://github.com/zll17/Neural_Topic_Models`, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Bidirectional encoder representations from transformers," 2016.

[6] M. Grootendorst, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," *arXiv preprint arXiv:2203.05794*, 2022.

[7] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[9] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[10] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[11] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," *Proceedings of GSCL*, vol. 30, pp. 31–40, 2009.

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