Topic Modeling For Research Articles

Kalgee Anand Kotak, kkotak, CSC 522 Jainam Ketan Shah , jkshah , CSC 522
Venkata Gnana Vardhani Kanamarlapudi, vkanama, CSC522

Abstract

Researchers approach huge online files of scientific and research articles. As an outcome, discovering important articles has become increasingly troublesome. Topic modeling is an approach to give a clear badge of ID to investigate articles which works with suggestion and search measure. On the off chance that a few words are happening more in certain records, it implies they have similar theme or topic. Topic modeling for research articles is finding a group of words which belongs to the same topic from a collection of documents or an abstract. Multiple methods for topic modeling are explored and analysed for research article topic classification.

1 Introduction

Researchers have access to electronic format of the research papers and articles. To have a better way of managing the explosion of huge electronic documents archives these days , it requires new tools or technologies that can automatically organize searching , indexing and browsing large collections. So that using such tools or technologies finding articles or research paper would become more and more easier. As we know artificial intelligence and deep learning is changing today's world to a large extent. Natural language processing(NLP) is an area of artificial intelligence which has developed new technique for processing of natural language that can be interpreted by the computers. NLP based models are able to identify patterns of words in document collection.The main importance of topic modeling is to discover word usage patterns and how to connect documents that share similar patterns. Therefore, the idea of the topic model is to process the terms of documents, which are a mixture of topics, where the topic is the probability distribution of words. In other words, the topic model is a generative model of the document. It specifies a simple probabilistic process through which documents can be generated. Despite the advancements in field of deep learning, the performance of deep learning models in NLP is less when compared with the performance of deep learning in Computer Vision. One of the main reasons for this can be because of the absence of large labelled text data sets. There has been great progress in the field of Natural Language Processing most of which involve supervised learning techniques for topic modeling and text classification. These supervised techniques require large amount of labeled data for training process. It is difficult to obtain topic of the text data specific to each task. Given a hypothesis natural language inference is used to determine the topic it belongs to.

2 Background

2.0.1 Literature Survey

Convolutional Neural Networks, are a type of neural network that uses a multi-layer perceptron variant and is designed for minimal preprocessing. CNN is also called translation-invariant or spatially invariant artificial neural network (SIANN), based on a shared weight architecture and invariant translation characteristics.CNN has proven to be very effective in areas such as image recognition or

classification. However, these days, there are many CNN studies that use text as processed data. Text mining and natural language processing are currently the most interesting research areas. The results show that CNN has not solved some problems in the field of text mining and NLP. This happens because CNN is used to solve problems in every situation, such as sentiment analysis, document classification, or NLP situations (such as entities and their relationships) or semantic representation. CNN, which is proficient in image classification, has proven its ability to process text. Appropriate data representation and methods have brought this success.(2)

LSTM is a recurrent neural network that can be used to capture long-range dependencies in a sequence. The LSTM model has multiple LSTM units, and each LSTM unit models the digital memory in the neural network. It has gates that allow the LSTM to store and access information over time. For example, the input/output gate controls the input/output of the cell, and the forget gate controls the state of the cell. The model takes words encoded in 1-hot encoding from the input, converts them to an embedding vector, and consumes the word vectors one at a time. Given the sequence of words that have been seen, the model is trained to predict the next word.To adapt the LSTM cell that takes words to a LSTM cell that takes as input both words and topics, The modification of representing equations for the operation of the LSTM cell to add topic vector to the input gate , forget gate cell and output gate. (3)

The previous methods used LSTM to model the relationship between the target word and its context. However, LSTM is difficult to parallelize, and the backpropagation with truncated time makes it difficult to remember long-term patterns. To overcome this we used the BERT model. BERT is a new language representation model, which uses a bidirectional Transformer network to pre-train a language model on a large corpus, and fine-tunes the pre-trained model on other tasks. The task-specific BERT design is able to represent either a single sentence, or a pair of sentences as a consecutive array of tokens. For a given token, its input representation is constructed by summing its corresponding token, segment, and position embeddings. For classification tasks, the first word of the sequence is identified by a unique token, a fully connected layer is connected at the position of the last encoder layer, and the last softmax layer completes the classification of sentences or sentence pairs. Hence the BERT model learns the context of the sentence for classification in a modified way. There are various methods that can be used in topic modeling, TF-IDF, LDA, LSA. Bertopic is a topic modeling approach that extends Bert embeddings and a class-based TF-IDF to create dense clusters allowing for easily interpretable topics while keeping important words in the topic descriptions.(5)

3 Method

3.1 CNN (Convolution Neural Network)

A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data. Embedding Layer of CNN can convert unprocessed documents into meaningful numeral matrix based on the dimension/size of words. Abstract data is represented as a sequence of word vectors through data preprocessing is fed in. In convolutional Layer contextual information of words present is captured. We have used three 1-dimensional Convolutional Layers followed by Batch Normalization and Max Pooling Layer. We performed Max pooling Operation on specific kernel to obtain maximum features from convolutional layer and Batch Normalization is performed to stabilize and perhaps accelerate the learning process. Droupout layers are added to improve the robustness of model. Next is Flattening Layer where we flatten our entire matrix into a vector like a vertical one so, that it will be passed as an input to 2 Fully Connected Layers with rectified linear unit (ReLU) and Softmax activation functions. We use Categorical Cross Entropy as loss function as this is multiclass classification problem. ith training to Validation samples ratio as 80 :20.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 668, 128)	5564032
conv1d (Conv1D)	(None, 661, 64)	65600
dropout (Dropout)	(None, 661, 64)	0
batch_normalization (BatchNo	(None, 661, 64)	256
conv1d_1 (Conv1D)	(None, 654, 256)	131328
dropout_1 (Dropout)	(None, 654, 256)	0
batch_normalization_1 (Batch	(None, 654, 256)	1024
conv1d_2 (Conv1D)	(None, 647, 512)	1049088
dropout_2 (Dropout)	(None, 647, 512)	0
conv1d_3 (Conv1D)	(None, 640, 1024)	4195328
dropout_3 (Dropout)	(None, 640, 1024)	0
batch_normalization_2 (Batch	(None, 640, 1024)	4096
max_pooling1d (MaxPooling1D)	(None, 320, 1024)	0
flatten (Flatten)	(None, 327680)	0
dense (Dense)	(None, 1024)	335545344
dense_1 (Dense)	(None, 29)	29725

Total params: 346,585,821
Trainable params: 346,583,133
Non-trainable params: 2,688

Figure 1: Simple CNN Model for text modeling

3.2 LSTM (Long Short-Term Memory)

LSTM is a reccurent neural network with gates mechanism learning order dependence in sequence. Usaually RNN have problem if the input sequence is large and do not know to remember the context of the tokens used. Where as LSTM uses gates to memorize or forget the particular token based on the context. LSTM with LSTM layers with 1024 and 128 memory units. 1 Fully Connected Layers with rectified linear unit (ReLU) and Softmax activation functions. We use Categorical Cross Entropy as loss function as this is multiclass classification problem.With training to Validation samples ratio as 90 : 10.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 3583, 100)	5000000
spatial_dropout1d (SpatialDr	(None, 3583, 100)	0
lstm (LSTM)	(None, 3583, 1024)	4608000
lstm_1 (LSTM)	(None, 128)	590336
dense (Dense)	(None, 29)	3741

Total params: 10,202,077
Trainable params: 10,202,077
Non-trainable params: 0

Figure 2: LSTM Model for Text modeling

3.3 Bert(Bidirectional Encoder Representations from Transformers)

One of the biggest challenges faced in the field of Natural Language Processing is the lack of enough training data. In general there is a huge amount of text data available, most of which is not annotated. For domain specific tasks to be performed it is necessary to create task-specific data sets. This results in us having only small amounts of labelled usable data. In order to achieve good performance, deep learning models require huge data. To reduce gap present in available training data, researchers have developed various techniques for training general purpose language representation models using unannotated text available on the web which is known as **pre-training**.These general purpose models are first pretrained on huge corpus and then they are fine-tuned on smaller task-specific datasets.This approach results in great accuracy improvements compared to training on the smaller task-specific datasets from scratch. Bert is one of the recent additions to these techniques for NLP pre-training.

Bert (Bidirectional Encoder Representations from Transformers) is a Natural Language Processing Model which achieves state-of-the-art accuracy on many NLP tasks like Question Answering and Reading Comprehension. Pretraining of Bert was done on Wikipedia and Bookcorpus.

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 150)]	0	
input_mask (InputLayer)	[(None, 150)]	0	
segment_ids (InputLayer)	[(None, 150)]	0	
keras_layer (KerasLayer)	[(None, 768), (None,	109482241	input_word_ids[0][0] input_mask[0][0] segment_ids[0][0]
tf.__operators__.getitem_5 (Sli	(None, 768)	0	keras_layer[5][1]
dense_11 (Dense)	(None, 1024)	787456	tf.__operators__.getitem_5[0][0]
dropout_6 (Dropout)	(None, 1024)	0	dense_11[0][0]
dense_12 (Dense)	(None, 29)	29725	dropout_6[0][0]

Total params: 110,299,422
Trainable params: 110,299,421
Non-trainable params: 1

Figure 3: Bert Model for Text modeling

3.3.1 Bert architecture

Bert comprises of an encoder stack of transformer architecture. A transformer architecture is an encoder-decoder network which uses self-attention on the encoder side. Attention on the decoder side of the network. Bert model was released in two sizes - $Bert_{BASE}$ and $Bert_{LARGE}$. These two sizes differ in number of parameters and hidden layers and their specifications are mentioned below:

- $Bert_{BASE}$ – 12 layers, 768 hidden size, attention head size of 12, and a total parameter size of 110 million.
- $Bert_{LARGE}$ – 24 layers, 1024 hidden size, attention head size of 16 and a total parameter size of 340 million.

Bert's functionality relies on the working of a transformer that employs attention mechanism. This helps the Bert model to learn the context dependent relationships between words in the training data. A Transformer comprises of an encoder module and a decoder module. The encoder is used for reading the input text and the decoder's job is to generate predictions specific to the task. As the goal of Bert is to create language representations, only the encoder part is used. A sequence of tokens are given as input to the encoder for Bert, which is first converted into vectors and then processed in the neural network. But before processing, Bert needs the input to be pre-processed (to change to a convenient format which the model understands).

The following pre-processing steps are performed on the input text:

- **Token embeddings:** In this two tokens are used. The first is the special classification token - [CLS]. This [CLS] is will be added at the beginning of the sentence 1. The second is the separator token - [SEP]. This [SEP] token is inserted at the end of each sentence so that the model is able to recognise where the sentence ends.
- **Segment embeddings:** In this, a marker is added to each token to specify whether that token belongs to Sentence A or Sentence B. Because of this, the encoder will be able to distinguish between sentences.
- **Positional embeddings:** To specify the position of each token present in the sentence a positional embedding is added.

Bert follows a technique which is called as Masked Language Modelling (MLM). In this technique a few words are randomly masked in the sentence and those masked words have to be predicted. In order to predict the masked word it uses the full context of the sentence, taking into account the words present in both the left and right side of the masked word. Unlike the previous other language models, the tokens of the previous and next words are used at the same time.

3.3.2 Pre-Training Bert

The pre-training phase of Bert is done using two unsupervised tasks, one is masked language modelling and the other is next sentence prediction.

Masked Language Modelling(MLM) The task of predicting the next word given a sequence of words is called Language Modelling. Instead of predicting every next token, a percentage of input tokens are masked at random and only these tokens have to be predicted. This is known as "masked LM" (MLM) because of the masking which is done. The final hidden vectors corresponding to the mask tokens are fed into a softmax layer which has outputs over the vocabulary, as it is in the case of standard LM. In all of our experiments, we will mask 15% of over all tokens in a sequence at random and predict those masked words instead of reconstructing the entire input.

A drawback of this is that a mismatch is created between fine-tuning and pre-training, because the [MASK] token will never appear in the fine-tuning stage. To overcome this issue, we do the following instead of replacing the masked words with token [MASK].

- $[MASK]$ token is used to replace 80% of tokens.
- 10% of the time a random token is used to replace tokens.
- Rest of the cases tokens are left without making any change.

4 Experiments

4.1 Dataset Description:

The data set consists of abstracts from various research articles and the topic or genre it belongs to. There are 31 columns, with I'd, Abstract and one hot representation of 29 topics the abstract belongs to. Multiple topics labelled are Computer Science, Mathematics, Physics, Statistics, Analysis of PDEs, Applications, Artificial Intelligence, Astrophysics of Galaxies, Computation and Language, Computer Vision and Pattern Recognition, Cosmology and Non Galactic Astrophysics, Data Structures and Algorithms, Differential Geometry, Earth and Planetary Astrophysics, Fluid Dynamics, Information Theory, Instrumentation and Methods for Astrophysics, Machine Learning, Materials Science, Methodology, Number Theory, Optimization and Control, Representation Theory, Robotics, Social and Information Networks, Statistics Theory, Strongly Correlated Electrons, Superconductivity, Systems and Control. It has Training samples of 14004, Testing samples of 6002, articles which are classified into respective topics.
`https://www.kaggle.com/abisheksudarshan/topic-modeling-for-research-articles?select=Train.csv`

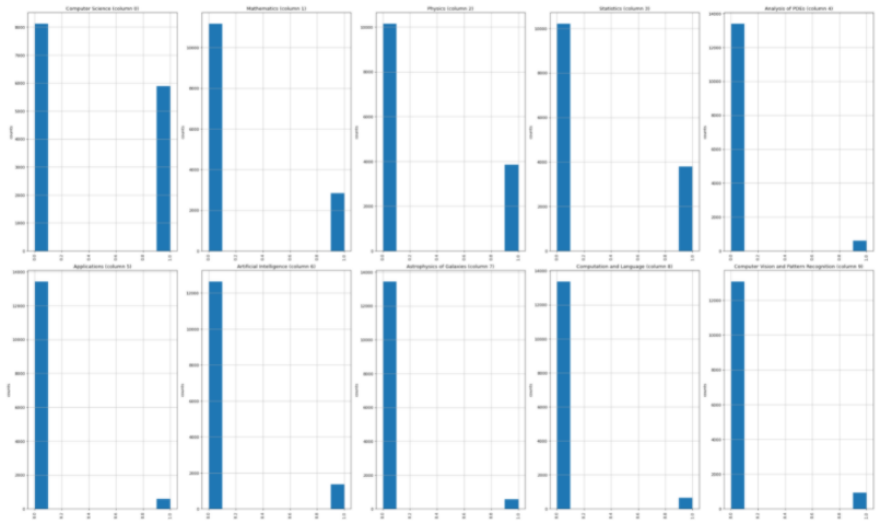

Figure 4: Multiple label Counts.

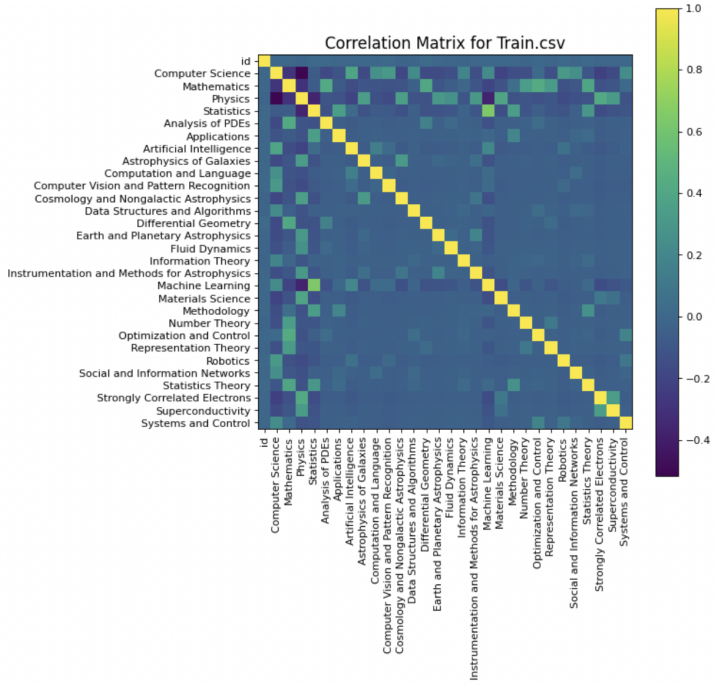

Figure 5: Correlation Matrix

4.2 Data Preparation and Pre-processing:

The data from the data set is taken from a csv file and processed further. Exploratory Data Analysis (EDA) is performed to better understand the data. Correlation matrix for the data helped us identify the correlated topic For example, Machine Learning and statistics are highly correlated topics in this dataset. Checked for the Null values and duplicates to eliminate them and clean the data. Regular expression is used to remove any punctuation in the abstracts. Then we will lowercase the words. Abstract from the data set is tokenized for text, followed by stop word removal and lemmatization i.e. converted the word to their dictionary base words. Finally we converted the text samples into vectors for analysis. We also reduced the text data length to a maximum length of 200.For Bert model to

accept the input data it should be encoded in specific format. The training data is formatted to Bert encodings and them given for model.

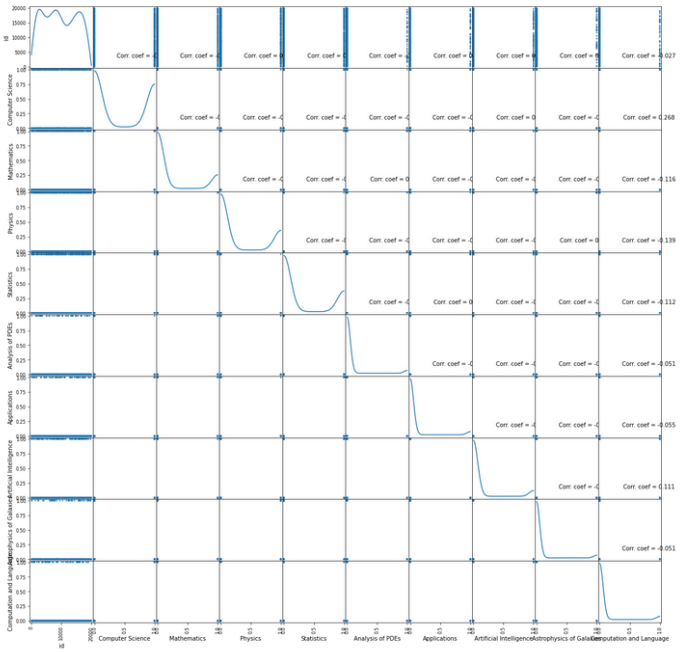

Figure 6: Correlation between features

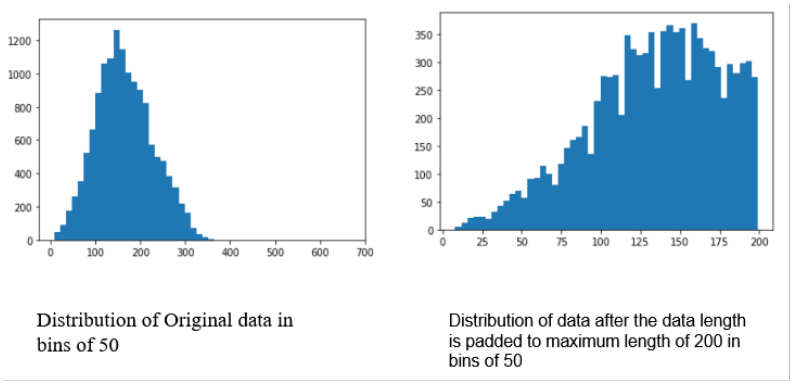

Figure 7: Tokenized Sequence histogram.

4.3 Performance Analysis:

We evaluated the performance of the classifiers using CNN and Bert models. The CNN classifier is trained over 15 epochs for batch size of 64 and for each abstract sequence of maximum length 668. Analysed CNN model with various changes in hyper parameters, the improved accuracy is with 40 epochs. LSTM model is incorporated with 10 epochs While the Bert model is trained over 5 epochs for batch size of 32 and for each abstract sequence of maximum length 150. Validation accuracy and further metrics are mentioned in the table below.

Table 1: Overview of model performance

Model	No. of epochs	Layers	Validation Accuracy
CNN	15	5	26.45%
CNN	40	5	42.2 %
LSTM	10	5	43.47 %
Bert	5	14	78.43 %
Bert_preprocess	10	14	83.2 %

5 Conclusion:

Topic modeling on research article to classify which topic the given abstract belongs to is the basic problem we are trying to solve. Bert, LSTM and CNN models are explored with various parameters. Experimented CNN and LSTM models for various epochs and variable layers.
Experimental data shows that Bert model with complete data pre-processing gives the better performance among all the evaluated models. Bert model performed better as it has better contextual understanding and the encoding are fine tuned and classified into their respective topics.
The context of the words used in a specific research abstract is captured well using BERT model when compared to LSTM and CNN models. So using BERT model for this problem gives better results for topic modeling.
Github link: `https://github.ncsu.edu/kkotak/engr-ALDA-fall2021-P14`

References

[1] Convolution Neural Network for Text Mining and Natural Language Processing ,N I Widiastuti 2019 IOP Conf. Ser.: Mater. Sci. Eng. 662 052010.

[2] L. Yao and Y. Guan, "An Improved LSTM Structure for Natural Language Processing," 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), 2018, pp. 565-569, doi: 10.1109/IICSPI.2018.8690387.

[3] P. Anupriya and S. Karpagavalli, "LDA based topic modeling of journal abstracts," 2015 International Conference on Advanced Computing and Communication Systems, 2015, pp. 1-5, doi: 10.1109/ICACCS.2015.7324058.

[4] Z. Gao, A. Feng, X. Song and X. Wu, "Target-Dependent Sentiment Classification With BERT," in IEEE Access, vol. 7, pp. 154290-154299, 2019, doi: 10.1109/ACCESS.2019.2946594.

[5] Tutubalina, E., Nikolenko, S. Exploring convolutional neural networks and topic models for user profiling from drug reviews. Multimed Tools Appl 77, 4791–4809 (2018). https://doi.org/10.1007/s11042-017-5336-z

[6] T. Patten and P. Jacobs, "Natural-language processing," in IEEE Expert, vol. 9, no. 1, pp. 35-, Feb. 1994, doi: 10.1109/64.295134.