A Novel Approach of Neural Topic Modelling for Document Clustering

Sandhya Subramani
Telecommunication Engineering
BMS College of Engineering
Bengaluru, India
sandhyasubramani14@gmail.com

Vaishnavi Sridhar
Computer Science Engineering
BMS Institute of Technology
Bengaluru, India
vaishnavi.rao2@gmail.com

Kaushal Shetty
Computer Science Engineering
Sri Jayachamrajendra College of Engineering
Mysore, India
kaushalshetty@outlook.com

***Abstract*— Topic modelling is a text mining technique to discover common topics in a collection of documents. The proposed methodology of topic modelling used artificial neural networks to improve the clustering mechanism of similar documents by modelling probabilistic relations between the topics, documents and vocabulary. Currently, while topic modelling and clustering are considered to be manifestations of unsupervised learning, and neural networks on the other hand are used for supervised learning problems, Neural Topic Modelling reformulated topic modelling into a supervised learning task by defining an objective function whose loss function had to be minimized. Custom input embedding layers were designed in order to extract the semantic relationships between the words in the corpus, and the output of the model presented a topic probability distribution for each document. The documents with similar distributions were then bucketed together based on the criteria of meeting the threshold value of a simple distance based similarity metric, such as cosine similarity. The model was implemented using Keras with TensorFlow backend and the effectiveness of the clustering was validated on the IMDB Movie dataset and the News Aggregator dataset from UCI. On comparison with other commonly used clustering mechanisms in combination with traditional topic models, the proposed model delivered an improved Silhouette Co-efficient Score and Davies-Bouldin Index, along with an increased data handling capacity, thereby making the solution scalable.**

Keywords—Deep Learning; Artificial Neural Networks; Topic Modelling; Clustering; Neural Topic Modelling;

I. INTRODUCTION

Internet and Big Data technologies enable capturing and storing data from numerous sources. But much of this data such as web-content, chat transcripts, application logs, etc. is unstructured. Natural Language Processing (NLP) [1] provides a set of techniques to derive structured and meaningful information which lead to actionable insights and decision making. One such NLP technique is topic modelling which seeks to discover topics from a corpus. A topic indicates the main theme or the underlying subject matter present in the documents. Besides, topic models discover topics from unlabeled or untagged text in an unsupervised fashion, which allows their application to the vast majority of documents.

Topic models have important applications such as recommendation of relevant articles, assembling of a text corpus and abstraction of the corpus in latent variables fewer than the original vocabulary, also known as dimensionality reduction. These capabilities in turn demonstrate immense scope in tackling real world issues [2][3] across a variety of domains, right from field of Biology, in the clustering and subsequent categorization of various flora and fauna given their features, to City Planning, in the identification of similar civil structures to result in appropriate price charting, and even in the various areas of Environmental Science such as Earthquake Studies, that necessitate the pattern analysis of observed earthquake epicenters to better understand danger zones, Ecosystem Monitoring that requires continued analysis of habitat in order to establish regulatory laws to govern waste dumping, illegal deforestation, poaching, and more. Topic models also play an important role in commerce, in aiding consumer segmentation, fraud identification, product recommendation, and beyond. Thus, it is imperative to design efficient and scalable topic models.

II. LITERATURE SURVEY

While the concept of using the topic distribution of documents coupled with distance based similarity metric such as cosine distance [4] to identify document similarity is not fairly new [5], traditional clustering mechanisms primarily involved the implementation of K-Means [6] and similar algorithms on the vectorized representation of text. This came from the basic definition of clustering – a mechanism to identify intrinsic groups within datasets of unlabeled data. It follows the principle [7] that well grouped clusters are identified by their property of having maximum intra – cluster similarity and minimum inter – cluster similarity in such a way that the individual groups contain objects that share the same properties. However, while statistically accurate, this technique was found to be linguistically ineffective because of the lack of consideration of the semantics in the text corpus. Thus, the idea of identifying similar documents by first determining the various subjects that it talks about, and then grouping it with other documents that exhibit similar subject content, came into play.

A topic model seeks to discover a predefined number of topics $T=\{t_1, t_2, \ldots, t_M\}$ from a collection of documents represented as $D=\{D_1, D_2, \ldots, D_N\}$ having an overall vocabulary of $W=\{w_1, w_2, \ldots, w_V\}$ and generates two

978-1-5386-9276-9/18/$31.00 ©2018 IEEE

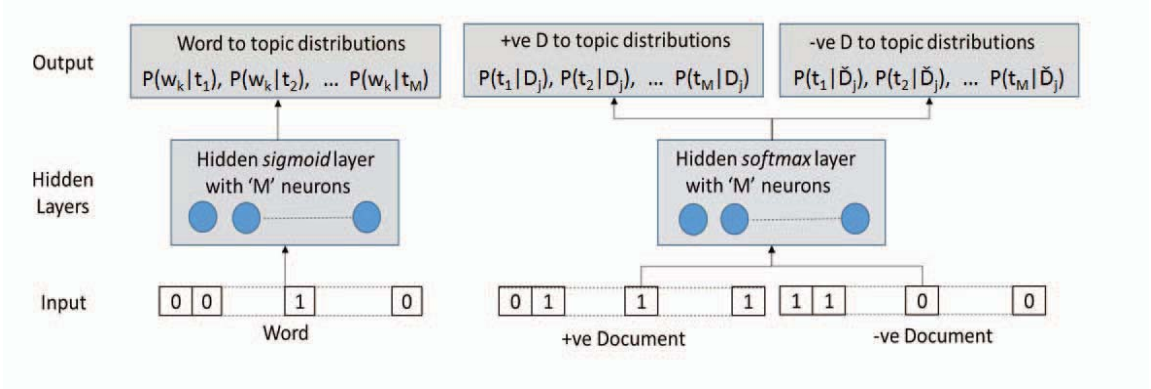

Fig. 1. Conceptual Depiction of Neural Topic Model

important outputs: (i) probabilistic relation between topics and documents – $P(t_i|D_j)$ which is the probability of t_i present in D_j; and (ii) probabilistic relation between words and topics – $P(w_k|t_i)$ which is the probability of w_k present in t_i. Probabilistic latent semantic indexing (pLSI) was introduced by Hofmann, 1999 [8] to capture these relations as multinomial distributions. Blei *et al.*, 2003 [9] generalized pLSI through Latent Dirichlet Allocation (LDA) which uses a term (w) – document (D) matrix1 to infer the probabilities. LDA has been by far the most commonly used topic model for small to medium sized corpus until the advent big data and the need to analyze larger corpuses. This made storage and computation of term – document matrices intractable. Thus, a Neural Topic Modelling (NTM) approach is proposed to address the limitations of LDA.

III. Methodology

NTM was motivated by earlier work on neural word embedding [10] which uses a neural network to capture semantic similarity between words and documents through latent variables. The main advantage of neural models such as word embeddings and NTM is the scalability for large corpuses. These neural models allow batch-wise model training by breaking down a large corpus into smaller batches which are easy to fit and compute in memory. The model is gradually updated over the individual batches and the whole process is repeated on the entire data for several passes. This approach forgoes storing and computing the entire data as a single object in memory and hence suits for big data.

Additionally, neural models can capture language semantics via word context rather than word count as used in LDA. As stated earlier, topic modelling is unsupervised and supposed to work in the absence of predefined document labels. However, neural network is a supervised learning method and requires input data to have labels. Therefore, it is imperative to design a neural topic model that can reformulate topic modelling as a supervised learning task.

Let the word w_k present in the document D_j. A topic model gives the conditional probability $P(w_k|D_j)$ of finding w_k in D_j as the following combination of word-topic distributions and topic-document distributions.

$$P(w_k \mid D_j) = \sum_{i=1}^{M} P(w_k \mid t_i) P(t_i \mid D_j) \quad (1)$$

Furthermore, consider the document $\tilde{D}_j$ which does not contain w_k. Applying (1), $P(w_k|\tilde{D}_j)$ can be computed which has the probabilistic notion of w_k being present in a negative document $\tilde{D}_j$. Similarly, $P(w_k|D_j)$ carries the notion of w_k being present in a positive document D_j. For a topic model to correctly capture the underlying probability distributions, $P(w_k|D_j)$ is required to be high and $P(w_k|\tilde{D}_j)$ to be low. This notion is expressed as follows:

$$P(w_k \mid D_j) - P(w_k \mid \tilde{D}_j) \geq \tau \quad (2)$$

which requires the difference to be a minimum of τ. Therefore, for an input quadruplet of $(w_k, D_j, \tilde{D}_j, \tau)$, NTM seeks to minimize the objective function.

$$C(w_k, D_j, \tilde{D}_j, \tau) = \tau - P(w_k \mid D_j) + P(w_k \mid \tilde{D}_j) \quad (3)$$

As for the inputs to the neural network: w_k, D_j and $\tilde{D}_j$ can have numeric forms such as one-hot encoding, word count etc. For simplicity, the target variable (or the pseudo target variable), τ can be same for all input quadruplets $(w_k, D_j, \tilde{D}_j, \tau)$.

Thus, NTM has reformulated the unsupervised problem to a supervised task. The parameters of the neural network along with the calculations of probabilities $P(w_k|D_j)$ and $P(w_k|\tilde{D}_j)$ are explained using Fig. 1. It shows a neural network with two parallel input, hidden layer and output streams. Word-Topic layer has a sigmoid activation and takes word (w_k) as input and gives the topic to word distributions as output. The sigmoid in this layer is required to obtain many-to-one probability mappings between M topics and a single word. In contrast, the Document-Topic layer has a softmax activation to generate one-to-many probability mappings from a single document to multiple topics. Besides, the Document-Topic

layer is shared by the two parallel input streams of positive and negative documents. Thus, the neural network is used to compute the necessary probability distributions to evaluate the objective function in (3). Now, using the input, the network parameters i.e., weights of the hidden layers and the output, the back propagation algorithm is applied to minimize the objective function for all words found in the corpus. The input quadruplet $(w_k, D_j, \widetilde{D}_j, \tau)$ can be generated by first taking a word w_k and a positive document D_j and then randomly picking r negative documents $\{\widetilde{D}_j\}^r$ where the word is absent. Here r : 1 is the negative sampling ratio.

IV. Implementation

A. Architecture

The architecture is as shown in Fig. 2. Three input layers, namely the n-gram, positive document and negative document embedding, were defined. These were fed to the n-gram topic layer and shared document topic layer which in turn gave the conditional probabilities of the n-gram given all the topics, as well as the conditional probabilities of the topics given the document. Then, given the positive and negative documents, the n-gram positive and negative document layers were computed with the help of the objective function which was minimized using a custom loss function. The final outputs were determined after sending all the training parameters through a softmax layer.

```
model.summary()
```

Layer (type)	Output Shape	Param #	Connected to
n_gram_embedding (InputLayer)	(None, 1291)	0	
pos_doc_embedding (InputLayer)	(None, 1291)	0	
neg_doc_embedding (InputLayer)	(None, 1291)	0	
n_gram_topic_layer (Dense)	(None, 500)	646000	n_gram_embedding[0][0]
shared_doc_topic_model (Model)	(None, 500)	646000	pos_doc_embedding[0][0] neg_doc_embedding[0][0]
merge_1 (Merge)	(None, 1)	0	n_gram_topic_layer[0][0] shared_doc_topic_model[1][0]
merge_2 (Merge)	(None, 1)	0	n_gram_topic_layer[0][0] shared_doc_topic_model[2][0]
merge_3 (Merge)	(None, 1)	0	merge_1[0][0] merge_2[0][0]

```
Total params: 1,292,000
Trainable params: 1,292,000
Non-trainable params: 0
```

Fig. 2. NTM Model Architecture

B. Algorithm

The model was implemented in Python using the Keras [11] Library with TensorFlow [12] running in the backend. Both Keras and TensorFlow are open source libraries. The pseudo-code for training the NTM is depicted in Fig. 3.

Input: T = {(n, d^p)} where $d^p \in D$ is a document containing the n-gram n;
For NTM, {label(d)} where label(d) is the correct label of d.
Pre-train;
repeat
  FOREACH (n, d^p) ∈ T:
    Sample d^n where (n, d^n) ∉ T;
    IF c (n, d^p, d^n) > 0:
      Update W_1, W_2 using backpropogation;
  IF NTM Then:
    FOREACH d ∈ D:
      Compute label error;
      Update W_1, W_3 using backpropogation;
until convergence

Fig. 3. NTM Training Pseudo Code

C. Dataset

The model was tested on the IMDB [13] and UCI Newsgroup [14] data, which are two widely used benchmark datasets for text categorization. The data was pre-processed to remove stop words and lemmatize verbs into their base form post their Parts of Speech tagging. Tokenization and n-gram word representations were also included as part of the custom embedding layers. After topic modelling, the various document similarities were calculated using cosine distance against the topic distribution of each document, and highly similar documents were grouped together to form clusters.

a) IMDB Top 1000 Movie Reviews Dataset: The IMDB dataset contains the 1000 most popular movies from 2006 to 2016 on IMDB. The dataset has attributes ranging from title, genre and synopsis description, to director, actor names, year of release, and even includes content on runtime, rating, votes, revenue and metascore. The aim of the task was to identify clusters of highly similar movies based on the description attribute which could be a combination of multiple topics.

Sample Topics	Top Words in Each Topic				
Topic 1	race	thrill	adventure	lost	ship
Topic 2	laughter	fun	party	humour	girlfriend
Topic 3	love	marriage	relationship	attraction	date
Topic 4	battle	invasion	nuclear	threat	fight
Topic 5	wrestle	athlete	boxer	champion	olympic
Topic 6	death	sorrow	disaster	accident	injure
Topic 7	alien	space	government	mission	lab
Topic 8	empire	slavery	war	king	prison
Topic 9	demon	dead	wood	dark	haunt
Topic 10	psycho	kill	hospital	police	therapy

Sample Clusters	Top Topics in the Cluster			Sample Movies in the Cluster		
Cluster 1	Topic 5	Topic 4	Topic 6	The Blind Side	Southpaw	Bleed for This
Cluster 2	Topic 8	Topic 6	Topic 3	Marie Antoinette	La da Baarovai	The Other Boleyn Girl
Cluster 3	Topic 9	Topic 10	Topic 7	Funny Games	Shut In	1408

Fig. 4. IMDB: Sample Clusters and Topic Composition

The dataset was modelled for 100 topics, of which a few of the topics and their top unigram words are displayed in Fig. 4. Each cluster comprised of unique documents containing various topics, and it was found that all the documents belonging to a given cluster had highly similar topic distribution probabilities. The total number of best clustered buckets was 52. The largest cluster size was 21 while the smallest cluster contained 4 documents.

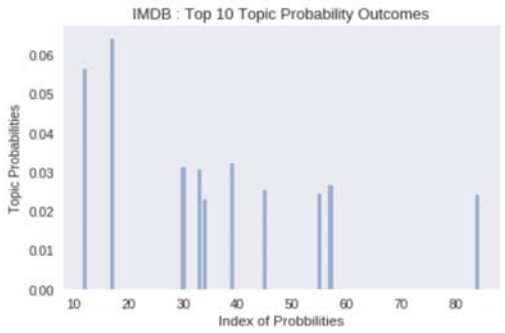

Fig. 5. IMDB: Top Ten Topic Probability Outcomes

Fig. 5. shows the highest ten topic probability outcomes from the total number of 100 topics.

b) UCI News Aggregator Dataset: The UCI newsgroup contains 422,937 news stories and their headlines, making it a large dataset. The data points included are id, title, url, publisher, category, story, hostname and timestamp. Business, science and technology, entertainment and health are the different category labels mentioned in this dataset. The aim of the task was to bucket similar headlines into clusters, beyond the scope of the category labels, based on the different unlying topics that each heading could contain.

Sample Topics	Top Words in Each Topic				
Topic 1	profit	stock	export	sale	economy
Topic 2	crash	bankrupt	bitcoin	currency	dollar
Topic 3	cinema	music	award	season	autograph
Topic 4	disease	diagnosis	weight	outbreak	treatment
Topic 5	google	ibm	apple	facebook	cisco
Topic 6	US	asia	europe	china	japan
Topic 7	heart	cancer	mumps	stroke	alzheimer
Topic 8	elephants	dinosaur	dog	cat	butterfly
Topic 9	valentine	bachelorette	rose	couple	wedding

Sample Clusters	Top Topics in the Cluster			Sample Headlines in the Cluster		
Cluster 1	Topic 1	Topic 6	Topic 7	Alibaba's Q4 revenue climbs 66% ahead of US IPO	Asian Stocks Rise On IBM's Earnings, China Data	EUR/USD trims gains on solid US factory, labor market data
Cluster 2	Topic 7	Topic 4	Topic 8	Ohio Mumps Outbreak Jumps To 23	US reports: Saturated fat may not cause heart disease	Canada reports claim Alzheimer's more likely than breast cancer in women over 60
Cluster 3	Topic 2	Topic 7	Topic 1	Mt. Gox files bankruptcy in US and faces more hacks	Following the Japan Filing, Now Mt. Gox Files For Bankruptcy Protection in the US	Japanese bitcoin exchange files US bankruptcy case (Update)

Fig. 6. News Aggregator: Sample Clusters and Topic Composition

The dataset was modelled for 100 topics, of which a few of the topics and their top unigram words are displayed in Fig. 6. Every cluster bucket comprised of unique documents having similar topic distribution. Examples of clusters with their headlines and topic composition are also shown in Fig. 6. The total number of best clustered buckets was 382. The largest cluster had 1467 documents while the smallest cluster contained 6 documents. Fig. 7. shows the highest ten topic probability outcomes from the total number of 100 topics.

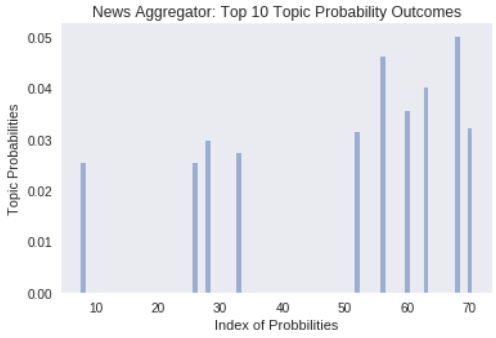

Fig. 7. News Aggregator: Top Ten Topic Probability Outcomes

V. RESULTS

A. IMDB Dataset

The Silhouette Score and Davies – Bouldin Index values for the various clustering methods against which the NTM model was compared with, is tabulated in Table I. The Silhouette Co-efficient values range from -1 to 1, the value 1 indicative of best clustering, and the Davis – Bouldin Index ranges from 0 to 1, the best value being 0. The optimal number of clusters for K-means was chosen by using the elbow method, and the clustering threshold for cosine distance similarity measure was set at 0.8 as it was found to give the best cluster results, after varied experimentation trials.

The NTM topic modelling model has the highest silhouette score of 0.8361, as well as the lowest Davies – Bouldin Index of 0.1833, thereby making it the best clustering mechanism amongst its peers.

TABLE I. CLUSTERING RESULTS OF IMDB DATASET

Sl No.	IMDB Dataset		
	Architecture	***Silhouette Co-efficient***	***Davies-Bouldin Index***
1	TFIDF + cosine similarity	0.732	0.355
2	TFIDF + K-means clustering	0.454	0.407
3	Word2Vec + cosine similarity	0.411	0.565
4	Word2Vec + K-means clustering	0.644	0.378
5	TFIDF + LDA + cosine similarity	0.794	0.239
6	TFIDF + NTM + cosine similarity	0.836	0.183

B. UCI News Aggregator Datset

When the NTM model architecture was compared against the various other models as tabulated in Table II, it was observed that the LDA algorithm gave a memory error while clustering the huge corpus of data. All the models were run on the same 256 GB RAM processor to ensure a head-to-head comparison. While the NTM architecture gave a high silhouette score of 0.7664, the TFIDF + cosine similarity

based model gave a slightly better result of 0.8673, and a comparable Davies – Bouldin index value.

TABLE II. CLUSTERING RESULTS OF IMDB DATASET

Sl No.	UCI News Aggregator Dataset		
	Architecture	*Silhouette Co-efficient*	*Davies-Bouldin Index*
1	TFIDF + cosine similarity	0.867	0.232
2	TFIDF + K-means clustering	0.685	0.421
3	Word2Vec + cosine similarity	0.403	0.612
4	Word2Vec + K-means clustering	0.495	0.397
5	TFIDF + LDA + cosine similarity	Memory error	Memory error
6	TFIDF + NTM + cosine similarity	0.766	0.211

VI. CONCLUSION

This paper presents the implementation of a neural network based topic modelling architecture for text corpus, along with its usage in the task of clustering by identifying highly similar documents. Unlike traditional topic models, neural topic models provide a scalable and unsupervised learning framework for automatically discovering topics from a corpus. While its clustering efficiency on small datasets is second to none, NTM truly outperforms its counterparts when dealing with large corpora such as the News Aggregator dataset from UCI. This is because not only does the model allow training in batches thereby making it highly flexible, but it also takes into consideration the semantic meanings of the words ensuring the usefulness of the topics. Therefore, the proposed method of topic modelling finds a variety of real world applications, from movie recommendations and news clustering to gene pool matching in Biotechnology.

REFERENCES

[1] Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc

[2] T. Soni Madhulatha, An Overview of Clustering Methods. Alluri Institute of Management Sciences, Warangal, 2012.

[3] Swaminathan Gurumurthy, Yongchao Jin, Lantao Yu, Weiping Li, Fei Fang, Chenyan Zhang, Xiaodong Zhang, Carnegie Mellon University, USA, 2018.

[4] Li B., Han L. (2013) Distance Weighted Cosine Similarity Measure for Text Classification. In: Yin H. et al. (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2013. IDEAL 2013. Lecture Notes in Computer Science, vol 8206. Springer, Berlin, Heidelberg.

[5] Kun Zhang, Zhikun Zhang, Jiji Zhang, ACM Transactions on Intelligent Systems and Technology, Vol. 7, No. 2, Article 25, Publication date: January 2016

[6] S. Na, L. Xumin, G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm", 3rd IEEE International Symposium on Intelligent Information Technology and Security Informatics (IITSI), pp. 63-67, 2010.

[7] Priyanka Sharma "Comparative Analysis of Various Clustering Algorithms", 2015

[8] Hofmann, T. (1999, August), Probabilistic Latent Semantic Indexing, in Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 50-57) ACM.

[9] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003), Latent Dirichlet Allocation, in Journal of Machine Learning Research, 3(Jan), 993-1022.

[10] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013), Distributed Representations of Words and Phrases and their Compositionality, in Advances in Neural Information Processing Systems (pp. 3111-3119).

[11] Keras website, https://keras.io/

[12] TensorFlow website, https://www.tensorflow.org/

[13] IMDB dataset website, https://www.kaggle.com/PromptCloudHQ/imdb-data

[14] UCI News Aggregator, https://archive.ics.uci.edu/ml/datasets/News+Aggregator

ABOUT THE AUTHORS

[1] 

Sandhya Subramani at the time of writing this paper is an Associate Software Engineer in Machine Learning and is working in the finance industry. She received her B.E in Telecommunication Engineering from BMS College of Engineering, Bengaluru, in 2016. Her areas of interest are Artificial Intelligence, Natural Language Processing and Object Oriented Programming.

[2] 

Vaishnavi Sridhar at the time of writing this paper is an Associate Software Engineer in Machine Learning and is working in the finance industry. She received her B.E in Computer Science Engineering from BMS Institute of Technology, Bengaluru, in 2016. Her areas of interest are Artificial Intelligence, Embedded Systems and Cognitive Computing.

[3] 

Kaushal Shetty at the time of writing this paper is an Associate Software Engineer in Machine Learning and is working in the finance industry. He received his B.E in Computer Science Engineering from Sri Jayachamrajendra College of Engineering, Mysore, in 2016. His areas of interest are Deep Learning, Natural Language Generation and Artificial Intelligence.

.