Vectorization of Text Documents for Identifying Unifiable News Articles

Anita Kumari Singh[1], Mogalla Shashi[2]
Department of Computer Science & Systems Engineering, Andhra University, India

***Abstract*—Vectorization is imperative for processing textual data in natural language processing applications. Vectorization enables the machines to understand the textual contents by converting them into meaningful numerical representations. The proposed work targets at identifying unifiable news articles for performing multi-document summarization. A framework is introduced for identification of news articles related to top trending topics/hashtags and multi-document summarization of unifiable news articles based on the trending topics, for capturing opinion diversity on those topics. Text clustering is applied to the corpus of news articles related to each trending topic to obtain smaller unifiable groups. The effectiveness of various text vectorization methods, namely the bag of word representations with *tf-idf* scores, word embeddings, and document embeddings are investigated for clustering news articles using the k-means. The paper presents the comparative analysis of different vectorization methods obtained on documents from DUC 2004 benchmark dataset in terms of purity.**

***Keywords*—Vectorization; news articles; tf-idf; word embeddings; document embeddings; text clustering**

I. Introduction

Recent developments in the world wide web have paved way for sharing different forms of information seamlessly on any platform. Among all the various sources of data, the textual representation of data continues to be the most widely used for communication and hence attracts the attention of the researchers to focus on developing automated tools for understanding as well as synthesizing textual information.

In this era of information outflow where billions of bytes of information are being created and shared worldwide for various purposes, it is essential to use the computing power of the machines for uncovering the unseen insights by transforming data to the way the machine understands. The realm of natural language processing has many possibilities for further research in making the interactions between human and machines much transparent. The initial step towards making the text documents machine-readable is vectorization.

Transforming textual data to meaningful vectors is a way to communicate with the machines for performing any Natural Language Processing tasks and solve problems mathematically. Researchers in the domain had proposed different vectorization models that range from a very simple to sophisticated ways helpful in solving NLP problems. A straightforward but ineffective way to build the vectorization table is, mapping all the words in the vocabulary to some integer value.

The work presented in the paper focusses on studying the impact of different vectorization methods for clustering text documents using the k-means algorithm. It is part of a framework which collects news articles based on URLs mentioned in social media posts, clusters the articles into smaller unifiable groups and automatically summarizes the multiple articles of each group for preparing a comprehensive news story related to the most trending topics and capturing the opinion diversity for the topics.

The organization if the paper is as follows, Section II presents the Related Literature, Section III introduces the framework for Identification and Hybrid Summarization of Unifiable News Articles. In Section IV, the paper details various Vectorization methods studied for the work; Section V briefs the Unifiable News Articles Identification process using Text Clustering with k-means. Section VI gives the details of the Dataset, Experimentation, Evaluation and Results. Section VII presents the Conclusions and Future Work.

II. Related Literature

A brief introduction to the prevailing text vectorization methods and contemporary word embedding models is as follows.

TF-IDF: Term Frequency-Inverse Document Frequency [1] is the most commonly used method in NLP for converting text documents into matrix representation of vectors. Tf-idf representation reflects the prominence a word in a collection of documents to the individual document. Successful search engines could be developed based on the potential of tf-idf scores for representing the prominence of words in the text to capture the relevance of the document to a given search query. However, the Inverse Document Frequency (idf) score calculation is vocabulary specific and hence hinders the applicability of tf-idf scores for dynamically changing corpora.

GloVe: Is a count-based model which constructs a global co-occurrence matrix where each row of the matrix is a word while each column represents the contexts in which the word can appear. The GloVe [2] scores represent the frequency of co-occurrence of a word with other words. GloVe learns its vectors after calculating the co-occurrences using dimensionality reduction. Other benefits of GloVe are its parallelizable implementation and ease of training over the large corpus.

Deep learning techniques were applied to process enormously large collections of text to extract word embeddings without confining to specific vocabulary or

corpora. Hence, word embedding has become the better choice for converting the text to machine-readable vectors for a wider variety of applications, including document summarization, language translation, question answering, and others.

Word embedding is a collection of different language modelling and feature learning techniques in NLP domain. The words or phrases in the vocabulary are mapped to vectors of real numbers, usually to a high-dimensional representation of words based on the context in which they appear.

Word2Vec: Word2Vec [3] builds a distributed semantic representation of words in the document. The model could be trained in the context of each word, such that similar words have similar numerical representations. Word2Vec is a predictive model that learn its vectors for reducing their loss of predicting the target words, from the given context words.

SentenceToVec: SentenceToVec is an extension to Word2Vec representation where feature representations at sentence level or the complete document are learned instead of words, by averaging the vector representations of all words in the sentence. Skip-Thought Vectors [4] released in 2015 have made good progress in sentence-level embeddings.

Doc2Vec: Doc2Vec [5] is an extension of Word2Vec or rather SentenceToVec as sentences are a part of documents, and the procedure of obtaining the Doc2Vec embeddings is similar to that of SentenceToVec.

While the single-level word embeddings discussed above are undoubtedly the most used of the word embeddings, they are still limited to capturing only the syntactic and semantic information of words from the sizeable collection of unlabeled text. While these methods suffice for document clustering, they cannot effectively handle more complex NLP tasks like Question-Answering, Textual Entailment, Named Entity Resolution, Sentiment Analysis and other as they produce context-independent embeddings with limited capability for Word Sense Disambiguation (WSD).

ELMo and BERT are the recent advancements for generating context-dependent word embeddings at multiple levels to be incorporated into various layers of deep learning models for solving complex NLP tasks successfully. Context-dependent embeddings like ELMo and Language Understanding models like BERT have taken the field to a different level.

ELMo: Embedding from Language Model (ELMo) [6] is a bidirectional Language Model (biLM) whose vectors are pre-trained using a large corpus to extract multi-layered word embeddings. ELMo Learns conceptualized word representations that capture the Syntax, Semantics and Word Sense Disambiguation (WSD). ELMo could be coupled with existing deep learning approaches for building supervisory models for a diverse range of complex NLP tasks to improve their performance significantly.

BERT: Bidirectional Encoder Representations from Transformers (BERT) [7] is based on the bidirectional idea of ELMo but uses a Transformer [8] architecture. BERT is Pre-trained to learn bidirectional representations by jointly conditioning the contexts of the corpus in both directions for all the layers. The pre-trained vectors could be used in complex NLP tasks and can achieve state-of-the-art results with only one additional layer at the output.

III. Framework for Identification and Hybrid Summarization of Unifiable News Articles

The proposed framework for Identification and Hybrid Summarization of Unifiable News Articles collects its inputs from a popular website, Trends24, which publishes the trending topics from Twitter on an hourly basis. Trends24 publishes real-time twitter trends in the form of hashtags and topics at multiple levels of granularity worldwide or in certain countries or cities. It has got a development page where the trending hashtags and topics of the entire day in a 24-hour frame is available. This page helps us to easily scrape the contents and extract all the information required for further processing. Figure 1 depicts the different phases in the framework.

After scraping the trending topics from trends 24, one can choose any of the popular and trending topics of interest based on how long it has been in the top trends. The trending hashtags or topics identified in the first part of the framework are used to gather all the tweets associated with those topics/hashtags. Twitter allows access to their data using APIs after establishing proper authentication using OAuth. OAuth is a standard for access delegation; it is used by websites or applications to access information from other websites without having to reveal any of the access credentials [9]. Twitter APIs like the Search API, Streaming API, and the REST API can help us obtain the publicly available data for free. Apart from these, there are other ways in which you can get more substantial chunks of data for real-time applications. Most of the popular programming languages provide built-in libraries for collecting and analyzing the tweets. A plethora of third-party tools is available on the web for performing various levels of analytics using social media [10].

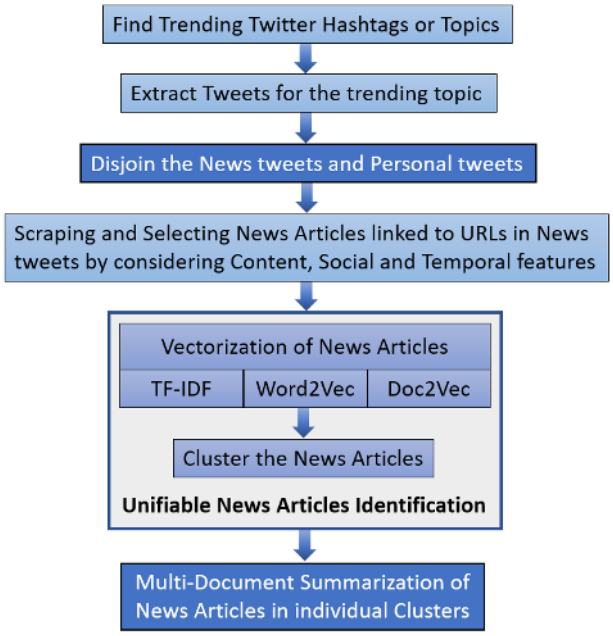

Fig. 1. Identification and Hybrid Summarization of Unifiable News Articles.

After obtaining enough number of tweets for a given topic, the next step in the framework is to separate and filter the News tweets from the collection. The method for identifying the news tweets from the collection works in three steps. In the first step, the *screenNames*(pseudonyms) are correlated with a set of news specific patterns to isolate the news tweets. The second step looks for news-related keywords by matching the tweet *tex*t with an extensive collection of news words to generate another partial set of tweets. Tweets extracted from both the streams are merged, followed by removing all duplicates, from this combined set, tweets only of the form *<TEXT, URL>* without any profanity words in the tweet text are selected to form the final set of news tweets. The details of the methodology for the Automatic Identification of News Tweets is elaborated in the paper [11]. All the identified news tweets have *URLs* in their text, which are the links to the actual news article where they are originally published.

The next part of the framework scrapes the actual news articles linked to the URLs in the News tweets. News articles are scraped from its original news sources and stored along with the references of the respective news sources. Opinion diversity is expected among the news articles collected for each topic as they might have discussed the topic in different perspectives. Relevant news articles are selected for each topic considering the features related to the content, social context and temporal aspects for further processing. For example, the news articles collected for one of the trending topic on twitter, "Nipah Virus" have included discussions on Nipah virus from perspectives like, the preparedness of hospitals and medical staff to deal with it, preventive steps through public awareness, different treatment procedures, and some articles included the statistics related to this epidemic, and other. Hence the news articles related to each perspective should be segregated from the others as a unifiable group for better comprehension through summarization of each group.

The news articles are grouped based on their semantic similarity into smaller clusters using clustering techniques. The paper investigates the effectiveness of vectorization for capturing the semantics of the documents using different state-of-the-art methods. The unifiable news article identification phase of the proposed framework studies *TD-IDF, Word2Vec* and *Doc2Vec* vectorization methods in detail and clusters the articles using the *k-means* clustering [12]. We elaborate on the proposed method for identifying Unifiable News Articles in Section V.

Each cluster thus obtained consist of multiple documents which discussed the topic in a specific perspective and hence are unifiable for a summary generation. The documents in each cluster are summarized using a Hybrid Multi-Document Summarization methodology proposed by the authors, details of which are elaborated in the paper [13]. The Hybrid Multi-Document Summarization is implemented using Deep Learning architecture with a cascade of Abstractive and Extractive summarization approaches.

The final summaries generated for the individual clusters could be used to build the underlying stories of the most trending topics on Twitter. Each statement in the final summary contains the references to the original news source, useful for further study, possibly, for resolving any conflicts.

IV. Vectorization of Text Documents

The section presents three commonly used methods for converting text documents to a vector representation. The proposed work studies tf-idf, Word2Vec, and Doc2Vec vectorizations in detail and experiments the approaches for clustering news articles using the k-means algorithm.

A. Term Frequency-Inverse Document Frequency (tf-idf)

The tf-idf score increases proportionally by the count of a particular word appearing in a given document (term frequency) and is neutralized by the count (inverse-document frequency) of the total number of documents in the corpus. The tf-idf matrix transforms all documents into rows, with all words in the documents stored as column vectors. The product of ***tf*** and ***idf*** is used to calculate the ***tf-idf*** score,**.**

$$tfidf(t,d,D)=tf(t,d)\times idf(t,D)$$

where ***t*** denotes the terms; ***d*** denotes each document; ***D*** denotes the collection of documents.

Term Frequency (**tf**):

tf(t,d)= (Number of times the term t appears in a document) / (Total number of terms in the document, d).

Inverse Document Frequency (**idf**):

$idf(t,D)= log_e$*(Total number of documents, D / Number of documents with term t in it).*

B. Word2Vec

Bengio et al. [14] first introduced the term word embedding in the year 2003. Collobert and Weston [15] were the first to depict the advantage of pre-trained word embeddings in 2008 and proposed the neural network architecture used in most of the recent approaches. Mikolov et al. [3], created Word2Vec model that revolutionized the use of word embeddings by introducing a toolkit that allows seamless training to the models and use of its pre-trained embeddings. Pennington et al. [2] in 2014, released GloVe, a competitive set of pre-trained word embeddings without using neural networks, signalling that word embeddings had reached the mainstream.

Word2Vec is a predictive neural-based word embedding model that provides probabilities to the words rather than frequencies. Word2Vec models process large text corpus to produce the output vectors using shallow neural network architectures. Though Word2Vec is a shallow neural network, the resulting vector representations are used for sophisticated language modelling by Deep learning architectures. Word2Vec is a combination of two models, the continuous bag of words (cbow), where the context of the word is used to predict the actual word and the skip-gram(sg), where the word is used to predict the target context. Skip-gram(sg) model could be used on large datasets to produce more accurate results.

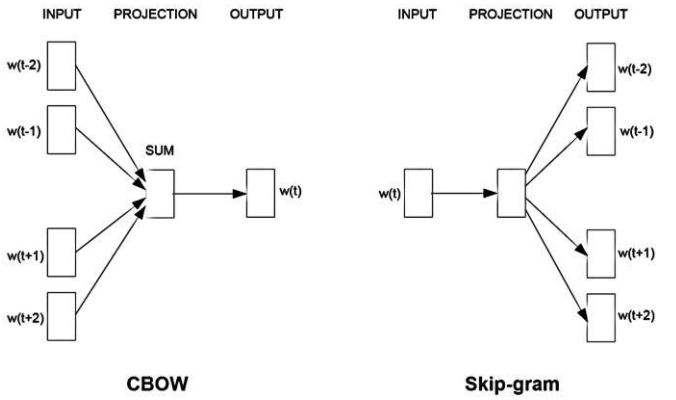

Fig. 2. CBOW and Skip-Gram Models, Adapted from Mikolov et al. 2013.

A snapshot of the Word2Vec models proposed by Mikolov et al. [3], is shown in figure 2, continuous bag of words methods predicts the word w(t) by taking the corresponding words as input. Also, the skip-gram techniques can predict the context words given the input word w(t).

C. Doc2Vec

Dealing with longer sentences, paragraphs or documents of varying lengths requires macro-level embedding techniques and Doc2Vec is devised for such scenarios. Doc2vec is an extension to the Word2Vec algorithm for learning continuous representations of larger chunks of text like sentences, paragraphs or the entire document in terms of constituent word embeddings. An additional sentence/paragraph token is added to obtain the document vectors.

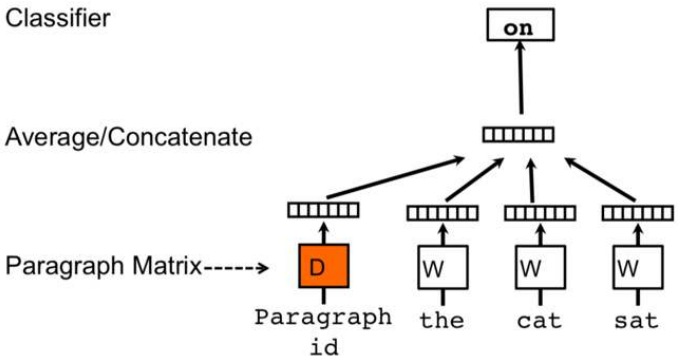

Fig. 3. Distributed Memory Paragraph Vectors(dmpv), Adapted from Mikolov et al. 2014.

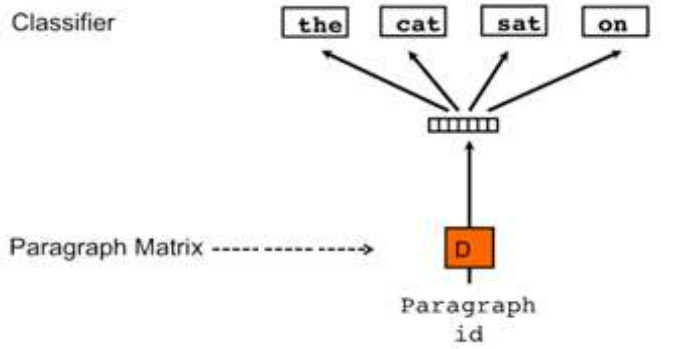

Fig. 4. Distributed Bag of Words(dbow), Adapted from Mikolov et al. 2014.

The Doc2Vec [5] model architecture also has two underlying algorithms the distributed memory paragraph vectors(dmpv) as shown in figure 3 and the distributed bag of words (dbow) shown in figure 4.

V. Identifying Unifiable News Articles using K-Means Clustering

Clustering algorithms for analyzing text, places documents into groups called subsets or cluster that are internally coherent and externally coupled. Text clustering is a procedure for cluster analysis of textual documents useful for machine learning (ML) and natural language processing (NLP) applications.

The procedure for text clustering includes a series of transformations of the original document, before obtaining the vector representation of the text. The documents are first pre-processed to remove all unwanted characters like punctuations, numbers, and other symbols, as they are in no way helpful for the clustering task. Later methods like stop words removal and stemming are applied to refine the text even more. After pre-processing, the normalized vectors of the text documents can be produced using any of the previously explained vectorization methods. The vectorized form of the input data is used to performs a k-means clustering over the set of documents and produce smaller groups of documents based on the given k value. The proposed work experiments with all the three vectorization methods discussed above to generate the vector representations for clustering the news articles using k-means.

k-means is an unsupervised learning algorithm that allows to group or clusters data points within your data based on some similarity. k-means is a grouping technique that groups the data into k clusters and assigns each data point to a particular cluster based on the similarity or distance measure to its centroid. The *k* in the k-means implies to the number of clusters and certain techniques like the “elbow method”, help in choosing the optimal number of clusters for large documents. The steps for k-means clustering are as follows:

1) Randomly chosen *k* data points act as the cluster centroids as the starting point, the remaining data points get assigned based on the nearest centroid within the cluster using any of the distance or similarity measures.

2) Reassign the respective centroids, after calculating the mean of all the data points in the individual clusters.

3) Repeat steps 1 and 2 until no new centroids constitute.

The clusters obtained using k-means on the news articles are chosen as the Unifiable groups that segregate the articles in the corpus based on the similarity of the news articles in capturing the different perspectives of the news articles. The identified groups of unifiable news articles can be summarized to get the underlying story for the trending topics in different perspectives.

VI. Dataset and Experimentation

The proposed work is to study different vectorization models and access the performance of k-means clustering on news documents. Three most commonly used vectorization

techniques were studied using DUC 2004 corpus for identifying clusters of unifiable news articles that could be summarized later in the proposed framework. tf-idf, Word2Vec and Doc2Vec vectorization methods were applied on the same data and clustered using k-means.

A. Dataset

Document understanding conference [16] (DUC) 2004, consist of 500 documents organized in 50 clusters, each with approximately 10 news articles related to a specific news topic form NEWSWIRE. This structured organization of the dataset helps us in estimating the purity of the clusters formed using k-means as each folder of DUC 2004 inherently is a cluster of unifiable news articles where all files in the folders relate to the same news topic.

Figure 5 displays the file size distribution of all the files from DUC 2004 in kilobytes. About 58% of the total files are of size 2KB and 3KB, and the rest are either too big or too small to form unifiable clusters.

B. Results and Evaluation

For interpreting the results and calculating the purity of the clusters, the proposed work is repeated for five times randomly selecting 10 folders and experimented with all the three vectorizing models. The averaged values of purity for all the five runs, for each of the vectorization methods, were tabulated and compared. The tf-idf vectorization model produced the best results in terms of purity of the clusters. Table 1 shows the purity scores of the three models on 10 randomly chosen folders containing 8 files each, with files whose size is in the range of 2KB or 3KB.

The experimental results indicate that the tf-idf vectorization method has produced more appropriate clusters for the articles with high purity value compared to Word2Vec and Doc2Vec vectorization methods. However, the tf-idf score depends on document frequencies for the words in the vocabulary and hence should be refreshed upon the arrival of a new chunk of news articles. Hence, tf-idf vectorization is preferable for handling the static collection of documents.

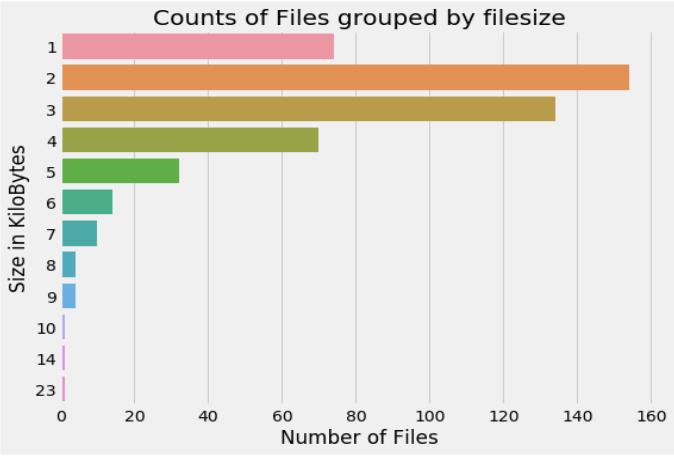

Fig. 5. Distribution of Files in KiloBytes.

TABLE I. PURITY SCORES FOR DIFFERENT VECTORIZATION

	TF-IDF	Word2Vec	Doc2Vec
Purity Score	**0.98**	0.89	0.95

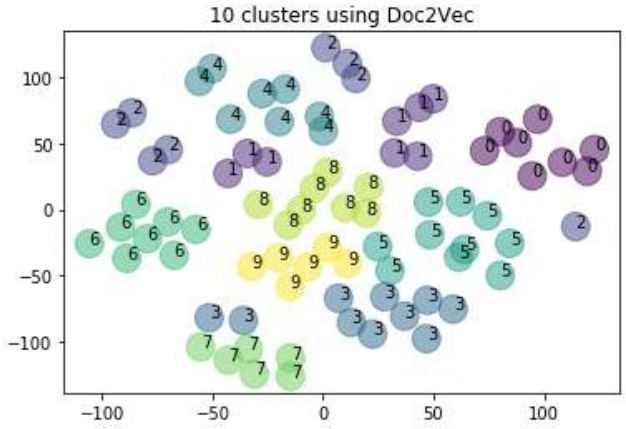

Fig. 6. Clusters using Doc2Vec.

Due to the limitation of tf-idf vectorization in handling corpora with continuously changing vocabularies, vectorization using Word2Vec and Doc2Vec embeddings offer better alternatives. Word2Vec and Doc2Vec could be used when there is an incremental set of news articles, and among the two methods, Doc2Vec approach produced high-quality clusters. Figure 6 depicts the Multi-Dimensional Scaling (MDS) visualization of 10 clusters produced by k-means with Doc2Vec vectorization of news articles.

VII. CONCLUSIONS AND FUTURE WORK

The proposed work introduces the framework for the Identification and Hybrid Summarization of news articles related to trending topics on social media. Different news articles related to a topic may have different perspectives and hence should be segregated into different unifiable groups based on their semantic similarity. The effectiveness of three vectorization methods, namely tf-idf, Word2Vec and Doc2Vec, for capturing the semantic similarity of news articles for identifying unifiable groups was investigated by clustering the vectorized news articles using the k-means algorithm.

The results obtained upon experimentation using documents available in DUC 2004 benchmark dataset are in favour of tf-idf vectorization with high purity cluster formation for static datasets. However, Doc2Vec vectorization is suggestable for handling news articles on trending topics as they require dynamically changing vocabularies.

The authors of the proposed work investigated the effectiveness of the existing text vectorization methods that generate single level word embedding of dynamically changing vocabularies required for clustering of news articles. As a future extension to this work, the authors propose to apply multi-level word embeddings using ELMo and BERT for building deep learning models for clustering news articles.

REFERENCES

[1] Sparck Jones, Karen. "A statistical interpretation of term specificity and its application in retrieval." Journal of documentation 28.1 (1972): 11-21.

[2] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[3] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[4] Kiros, Ryan, et al. "Skip-thought vectors." Advances in neural information processing systems. 2015.

[5] Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. 2014.

[6] Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

[7] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[8] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

[9] Whitson Gordon. "Understanding OAuth: What Happens When You Log into a Site with Google, Twitter, or Facebook". Retrieved 2016-05-15.

[10] Singh, Anita Kumari, and Mogalla Shashi. "Research Aids for Social Media Analytics." International Journal of Computer Science and Network 6.6 (2017): 753-759.

[11] Anita Kumari Singh and Shashi Mogalla, “Automatic Identification of News Tweets on Twitter”. International Journal of Computer Engineering and Technology,9(4), 2018, pp. 140-147.

[12] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297.

[13] Anita Kumari Singh, M Shashi "Deep Learning Architecture for Multi-Document Summarization as a cascade of Abstractive and Extractive Summarization approaches." International Journal of Computer Sciences and Engineering 7.3 (2019): 950-954.

[14] Bengio, Yoshua, et al. "A neural probabilistic language model." Journal of machine learning research 3. Feb (2003): 1137-1155.

[15] Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." Proceedings of the 25th international conference on Machine learning. ACM, 2008.

[16] Nenkova, Ani. "Automatic text summarization of newswire: Lessons learned from the document understanding conference." (2005).