Archiving System Optimization using Skip Gram based Neural Network as a Feature Selection

Wafaa Al Hameed (1) and Sura Khalid Salsal (2)

1 College of Information Technology, University of Babylon, Hillah, Iraq
2 College of Information Technology, University of Babylon, Hillah, Iraq

Abstract. Automatic processing of massive unstructured data and extracting useful information are one of the challenges researched so far. Hence, the techniques of data mining are of great significance in this area and text categorization (TC) is one of those common researches. The key problem in TC is due to the huge amount of data. The lack of effective description of the text of the content and the implementation of appropriate features leading to a decline in categorization accuracy. Therefore, several methods appeared to select the features that help clustering and improving their performance as it appears in our research. We suggest a good way to select a feature which support archiving system thus it supports the retrieval system.

1. Introduction

In this advanced era, companies and organizations are racing to collect, study and organize data, as machine learning has been used in many studies in order to deal with and archive giant data instead of traditional methods, and this in turn plays an important role in building a flexible retrieval system [1]. The aim of this study is to use a new method in feature selection process that realized that building such a system requires choosing characteristics that are meaningful and linked according to relationships, which in turn improves the performance of the model.

In this research, uses neural network as a bogus task in order to create an embedding matrix [2] that will employ later in the task of feature selecting using a certain method. Then using data segmentation/clustering, as a flexible alternative is possible Instead of searching for the most relevant result that matches one document at a time in all of the data, the data is divided into related groups where the search is only for the desired element in the group that relates best to its topic [3]. We used 20Newsgroups dataset [4], which were used to examine the system's efficiency through the speed and storage units that we need.

In strengthening the flexible and precise retrieval system that satisfies the user's need for this, the right clustering of textual data according to particular topics is of great importance. Studies in this respect are continuous of which A. I. Kadhim et al. (2014) propose a dimension reduction method using TF-DF SVD and then using clustering method for text categorization. This method is mathematically expensive when data is large [5], S. N. Karpovicha et al. (2020) Consider using a probabilistic modeling of the topic to construct a classification model, this
model facilitates precision tuning against the algorithm’s generality. And S. Qaiser and R. Ali (2018) in this paper, the work of the TF-IDF algorithm in text processing was explained and some possible developments were mentioned, and the failure in its work to semantically feature selection and its inability to deal with big data where it is very expensive computationally [6].

2. Problem and solutions
- There are some problems in most studies interested in creating an effective archiving system, including the establishment of a fast retrieval system from giant data with low storage and with acceptable accuracy.
- Proposing a solution by creating a new retrieval system, by building a new structure that includes employing the word embedding resulting from the embedding layer in skip gram - neural network in the process of feature extracting according to relationships. Thus building an indexing system based on the relationships between documents semantically, and this in turn helps build a flexible retrieval system in accordance with human standards.
- Using Neural networks for a better, faster, and more accurate feature-engineering task.
- Using clustering as a means of data grouping to replace traditional indexing methods, by using a hybrid method (Fuzzy C-Means algorithm for clustering and canopy for faster initialization).

3. Proposed System
In archiving system, each document in dataset is passed into three phases, which are pre-processing, embedding and clustering phases.
Phase 1: Pre-processing process
Input: text data as a document of any length.
Output: document as list of stemmed tokens.
Process: separate the text into a list of words, which is then tokenized and passed into a stemming algorithm.
Phase 2: The Embedding process:
Input: document as a list of stemmed tokens.
Output: document vector.
Process: convert all the tokens into a single numerical vector using a neural network-based.
Phase 3: The clustering process:
Input: list of all feature vectors
Output: indexing structure.
Process: this layer combines the feature vector for each and every document in a dataset to form the feature matrix (where each row corresponds to a single document), and clusters them based on their similarities so that the document can belong to one or more clusters with varying degrees of membership then create new dataset indexing structure.

3.1. The word embedding process:
Our paper focuses on this process. Word embedding is a good way to represent a word in a numerical vector. The simplest form of word embedding is a one-hot matrix, where the term represents a vector of length along the length of the vocabulary, and here several problems occur, which is a waste of time and space, which causes a curse of dimensionality. The algorithm are used to construct word2vec representation in our research the Skip-Gram based on neural network[7].

Skip gram neural network architecture:
Neural networks have dealt with many applications of natural language, and in the deep neural network it is necessary to have pre-processed and specific vocabulary. In addition, you must know that the accuracy of the system depends heavily on the size of this vocabulary. Here we
know the vocabulary size as the volume of data that is required to be trained on the system in order to get the required performance [2].

At the beginning of the training process for the network, batch generation process (a number of documents in each epoch) of training samples begins randomly for each epoch through the batch generator. Where the latter prepares the training samples on the CPU while the network starts training on the GPU to make the experiment faster. After preparing the samples, it shall enter the skip gram function that the process of preparing the diodes represents the entrance to the neural network for training is given. Let's suppose we have the word target denoted by T, the idea of skip gram is to guess the words that come with it and are considered within the context and this is done by defining a window of a certain size n, if it falls within the window then consider within context, otherwise, it is not context [8][9]. As shown in the following example with drawing:

![Figure 1. Illustrates Skip Gram description.](image1)

To prevent the system from entering an overfitting state, the network must be trained in samples that are in context and not in context.

Here a problem will appear when calculating the expected probability of each word in the event that considered the target word as the network needs to reset the weights of all words within the corpus even if they are not in context and this leads to slow training and weak system.

The solution to this problem is to use a technique called negative samples, which is that each sample entering the neural network will cause the updating of a small percentage of weights. In addition, this leads to the reduction of a very large number of computer burden in modifying these weights in case of big data and improves the quality of the word vector [1].

![Figure 2. Illustrates the skip gram explanation.](image2)
Suppose the following simple sentence:

“The tall boy runs and jumps in the marathon faster than the short boy”

Samples below showing the output form from the skip gram function and this sample are considered the entrance to the neural network:

If window size=2:
(boy (3), runs (4)) -> 1
(jumps (6), boy (3)) -> 0
(marathon (9), than (11)) -> 1
(boy (3), short (13)) -> 1
(runs (4), jumps (6)) -> 1
(short (13), tall (2)) -> 0

The objective of training the neural network is not the result of the training itself but the output of the embedding layer, it is the embedding matrix, which is our focus and which we will employ in a later work [10].

First, we enter the samples we extracted from the skip gram that are presented as pairs of integer numbers, each number represents a unique value or indexing for a specific word within the vocabulary and these pairs will pass to the input layer then through the four hidden layer, first the embedding layer.

Throughout this layer, it is essential to set two important parameters, which are the size of the vocabulary and the size of the embedding. The embedding vectors have a weights starts with random values, each of these words (the target word and the word which is in context or not) will be passed to its own embedding layer, which has matrix with 2D (vocabulary_size * embedding_size), and the result is (1 * embedding size) for each one of them [11].

Example: if we have two integer inputs each of them should pass to embedding layer with vocabulary size 300 and embedding size 10, then the embedding matrix contains 300*10 parameter and each word has a row of 10 weights, which represents the relation between the words.

Then the previous output is reshaped from 2D to a vector in a process called flattening layers, after that the output will pass to merging layer where the dot product is calculated between the two vectors and lastly we pass it in the dense layer where the sigmoid activation function is implemented[12][13].

In order to understand the accuracy of the predicted outcome, it must be compared with the real value while the loss value is calculated by special functions, including the mean square error MSE, which is calculated according to the following equation [14]:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} ||\hat{Y}_i - Y_i||^2$$

Where

N: is the number of training samples.

\(i\): denotes the i-th sample in training set.

\(Y\): denotes to actual value.

\(\hat{Y}\): denotes to predicted value.
Figure 3. Illustrates the shape of input and output from the Deep Network in case the embedding size is 100

After that should obtain the doc-vector for each document which created by combining the word-vector for every word in the document.

3.2. The Clustering process

Fuzzy c-Mean algorithm

It is one of the most popular and efficient clustering methods and is an extension of the k-mean algorithm. FCM is considered a theory that allows each element to belong to more than one cluster, where each object has a degree of belonging to each cluster, so that the word that lies on the edge of the cluster has the lowest degree of that which is located close to the center of the cluster. The FCM algorithm depends on a set of parameters as input, which must be predetermined for the algorithm, including the Fuzzier value, the number of clusters, the max iterations and the termination value. In our research, we will highlight the improvement in two parameters of the traditional FCM algorithm, which is the optimization of initial centroid values by canopy algorithm and the optimal way to choose the fuzzier value[15][16][17].

4. The dataset

To test the system, we used The 20 Newsgroups dataset. It is widely used for Applications of text mining can obtained from the link: http://qwone.com/~jason/20Newsgroups/. Ken Lang gathered it. The data set of 20 Newsgroups is a test set of about 20,000 documents of newsgroups that 1000 documents were collected from each of the newsgroups. It is split across 20 Varying newsgroups. Computers, politics, philosophy, athletics, and technology are related to the category themes. Every document it belongs to only one newsgroup, but a small portion of these documents belong to more than one newsgroups [4]:

5. The results of Implementation

| Table 1. Training parameters |
|-------------------------------|
| dataset | Vocabulary size | Embedding vector size | Total Documents | Samples Per Epoch | No. of Epoch |
|---------|-----------------|-----------------------|-----------------|------------------|--------------|
| 20newsgroups | 57368         | 100                   | 18828           | 10000            | 80           |
Table 2. Skip-gram results and time

| Avrg Terms Frequencies Time In MS | 545 |
|----------------------------------|-----|
| Skipgram Matrix Size (Row*Col)   | 19627 * 100 |
| Skipgram Matrix Size In Ram      | 7.85 Megabyte |
| Avrg Creation Time In MS         | 776 |

Table 3. Clustering Parameters, Results, and Time

| No. of clusters | 5 |
|-----------------|---|
| Max allowed iterations | 100 |
| Termination threshold | 0.00002 |
| Predicted Fuzzier values | 1.02497855731842 |
| Actual termination value | 0.000010 |
| Actual used iterations | 65 |
| Clustering Time In MS | 463 |

6. Conclusion
In our research, the feature selection process was carried out meaningfully according to human standards and not as usual in traditional methods such as TF-IDF that rely on frequency only. In addition, the use of the neural network for the model gives speed and high accuracy, which helps build an effective archiving system, which in turn contributes to supporting the retrieval system. Flexible matching query request according to user requirements. The use of NN in building an archiving system, despite the enormity of the data set, but it was collected and organized in a short time and few storage units without the need to reduce the dimensions of the data.

7. References
[1] D. Jurafsky and J. H. Martin, “Vector Semantics and Embeddings,” Speech Lang. Process., pp. 1–31, 2019. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/6.pdf.
[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., pp. 1–12, 2013.
[3] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, “An overview of clustering methods,” Intell. Data Anal., vol. 11, no. 6, pp. 583–605, 2007, doi: 10.3233/ida-2007-11602.
[4] K. Albishre, M. Albathan, and Y. Li, “Effective 20 newsgroups dataset cleaning,” Proc. - 2015 IEEE/WIC/ACM Int. Jt. Conf. Web Intell. Intell. Agent Technol. WI-IAT
2015, pp. 98–101, 2016, doi: 10.1109/WI-IAT.2015.90.

[5] A. I. Kadhim, Y. N. Cheah, and N. H. Ahamed, “Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering,” Proc. - 2014 4th Int. Conf. Artif. Intell. with Appl. Eng. Technol. ICAIET 2014, pp. 69–73, 2015, doi: 10.1109/ICAIET.2014.21.

[6] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” Int. J. Comput. Appl., vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.

[7] X. Rong, “word2vec Parameter Learning Explained,” pp. 1–21, 2014. [Online]. Available: http://arxiv.org/abs/1411.2738.

[8] P. M. Radiuk, “Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets,” Inf. Technol. Manag. Sci., vol. 20, no. 1, pp. 20–24, 2018, doi: 10.1515/itms-2017-0003.

[9] I. Kandel and M. Castelli, “The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset,” ICT Express, 2020.

[10] S. Wunderlich, M. Ring, D. Landes, and A. Hothro, “Comparison of System Call Representations for Intrusion Detection,” Adv. Intell. Syst. Comput., vol. 951, pp. 14–24, 2020, doi: 10.1007/978-3-030-20005-3_2.

[11] E. Tsironi, P. V. A. Barros, and S. Wermter, “Gesture Recognition with a Convolutional Long Short-Term Memory Recurrent Neural Network,” in ESANN, 2016.

[12] A. Wanto, A. P. Windarto, D. Hartama, and I. Parlina, “Use of Binary Sigmoid Function And Linear Identity In Artificial Neural Networks For Forecasting Population Density,” IJISTECH (International J. Inf. Syst. Technol.), vol. 1, no. 1, p. 43, 2017, doi: 10.30645/ijistech.v1i1.6.

[13] H.-C. Shin, M. Orton, D. J. Collins, S. Doran, and M. O. Leach, Organ Detection Using Deep Learning. Elsevier Inc., 2016.

[14] L. Gjestebøy et al., “Deep Neural Network for CT Metal Artifact Reduction with a Perceptual Loss Function,” fifth Int. Conf. image Form. X-ray Comput. Tomogr., pp. 439–443.

[15] V. Schwämmle and O. N. Jensen, “A simple and fast method to determine the parameters for fuzzy c-means cluster analysis,” Bioinformatics, vol. 26, no. 22, pp. 2841–2848, 2010, doi: 10.1093/bioinformatics/btq534.

[16] W. Dai, C. Yu, and Z. Jiang, “An Improved Hybrid Canopy-Fuzzy C-Means Clustering Algorithm Based on MapReduce Model,” J. Comput. Sci. Eng., vol. 10, no. 1, pp. 1–8, 2016, doi: 10.5626/JCSE.2016.10.1.1.

[17] A. McCallum, K. Nigam, and L. H. Ungar, “Efficient clustering of high-dimensional data sets with application to reference matching,” Proceeding Sixth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 169–178, 2000, doi: 10.1145/347090.347123.