Semantic Annotation of Documents

Sakala Venkata Krishna Rohit[1] Sharvil Katariya[2] Nikhil Chavanke[3]

***Abstract*— There are many articles and many documents in the internet that are been generated daily. But, there lacks an efficient mechanism to categorize them. We have designed a methodology to solve this issue. We have concentrated on a specific domain in this report. We have designed a model which classifies any new ACM research paper into the corresponding ACM classification tree 2012 categories. We have used the state-of-art technologies and classifiers for this purpose which makes our model robust. This can later be extended to any document (domain) provided the efficient classifier. Everyone is moving towards Semantic Web i,e Web 3.0 and so, we have stressed and made sure our algorithm classifies based semantics. Our analysis includes MAP, NDCG scores to tell the efficiency of the algorithms we have used.**

I. INTRODUCTION

We have used the state-of-art technologies to build our model. One being word2vec and another the extension of it doc2vec. Let me first describe about these technologies.Word2vec is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus.The purpose and usefulness of Word2vec is to group the vectors of similar words together in vectorspace. That is, it detects similarities mathematically. Word2vec creates vectors that are distributed numerical representations of word features, features such as the context of individual words.Doc2vec is an extension of word2vec that learns to correlate labels and words, rather than words with other words.The first step is coming up with a vector that represents the meaning of a document, which can then be used as input to a supervised machine learning algorithm to associate documents with labels.

The next section of the paper will be methodology where we will explain about each process in detail. After that we will have pictorial representations of the analysis that we have made and we will also reason about the results achieved. Finally, we will define the scope of the project. We will talk about how to extend the paper to achieve more better results.

II. METHODOLOGY

This section will give you the detailed analysis of each process involved in the project. Each sub section is mapped to one of the stages in the project.

A. Data Pre-Processing

The pre-processing stage involves removal of stop-words, stripping of excess white-spaces, removing punctuation's, tags etc from data-set to transform it into clean data-set. After this is done, the data-set is divided into training and testing sets so as to calculate ac-curacies later. As we have specified above, we are testing our model on ACM dataset. Here, our task is to map a new research paper to its ACM classification tree category. We have used 247543 abstracts for training the datasets. As the categories for this aren't mapping with ACM classification one's. We have mapped the 25 labels of this dataset with 4 high-level categories. The code is written in a generic manner. No restrictions on carnality of high-level categories. Testing data is taken as 10 percent of the total dataset.

B. WORD2VEC

Word2vec can make highly accurate guesses about a words meaning based on past appearances. Those guesses can be used to establish a words association with other words or cluster documents and classify them by topic. Those clusters can form the basis of search, sentiment analysis and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management.The output of the Word2vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep-learning net or simply queried to detect relationships between words.Word2vec trains words against other words that neighbor them in the input corpus.It does so in one of two ways, either using context to predict a target word (a method known as continuous bag of words, or CBOW), or using a word to predict a target context, which is called skip-gram. So, using this we have transformed each word of the dataset into vector space model with the help of gensim[1] libraries.

C. DOC2VEC

In Doc2Vec a document label is to be provided for a given piece of text.This method is almost identical to Word2Vec,except we now generalize the method by adding a paragraph/document vector. Like Word2Vec,there are two methods: Distributed Memory (DM) and Distributed Bag of Words (DBOW). DM attempts to predict a word given its previous words and a paragraph vector. Even though the context window moves across the text, the paragraph vector does not (hence distributed memory) and allows for some word-order to be captured. DBOW predicts a random group

*This work was supported by International Institute of Information Technology

[1]Sakala Venkata Krishna Rohit is a research student in Computer Science and Humanities at IIIT Hyderabad, India.

[2]Sharvil Katariya is a student in Computer Science at IIIT Hyderabad, India.

[3]Nikhil Chavanke is a student in Computer Science at IIIT Hyderabad, India.

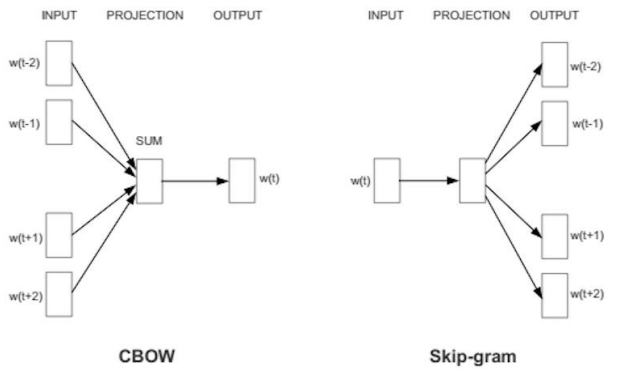

Fig. 1. The two architectures of Word2vec with their respective neural network implementations.

of words in a paragraph given only its paragraph vector.Once it has been trained, these paragraph vectors can be fed into a sentiment classifier without the need to aggregate words. This method is currently the state-of-the-art when it comes to sentiment classification on the IMDB movie review data set, achieving only a 7.42 percent error rate. So, we have used doc2vec in the process of training of classifier. This is also done using gensim[1] library.

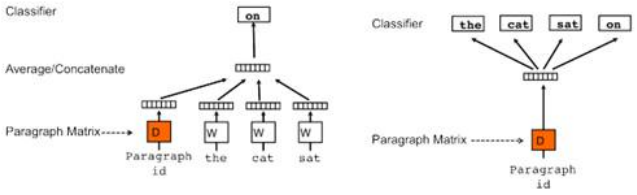

Fig. 2. The two architectures of Doc2Vec with their respective neural network implementations.

III. Training the classifier

For topic modeling, on needs to do supervised learning. We need features for each data point as an input to the classifier. We give the features as the concatenation of the dimensions of the vectors of words. We train the classifier and finally achieve the model. The task now remaining is to test it with cross validation. We have created vector space models for both the models in doc2vec i.e DM and DBOW.

IV. Testing the classifier

Testing the classifier includes cross validation and reporting the accuracy. We have taken ten percent of the data as the testing data and reporting all the necessary scores.

V. Analysis

For analyzing the efficiency of the system we are using the MAP and NDCG score.

A. Mean Average Precision (MAP)

Average precision score is the average of the precision values at the points at which each relevant document is retrieved.

Mean average precision (MAP) for a set of queries is the mean of the average precision scores for each queries.

If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero.

B. Normalized discounted cumulative gain (NDCG)

Normalized discounted cumulative gain (NDCG) measures the performance of a recommendation system based on the graded relevance of the recommended entities. It uses graded relevance as a measure of the usefulness, or gain, from examining a document.

Gain is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks. DCG values are often normalized by comparing the DCG at each rank with the DCG value for the perfect ranking. It varies from 0.0 to 1.0, with 1.0 representing the ideal ranking of the entities.

With the Testing Size of 10% and having trained on 10 Epochs.

TABLE I
BEST SCORES (WITH PORTER STEMMER)

Measure/Algorithm	Logistic Regression	SVM	Neural Network
DM - MAP	0.700075742274	0.699328418501	0.711400053861
DM - NDCG	0.973485165503	0.973485165503	0.860783052554
DBOW - MAP	0.694430418097	0.693854776813	0.711400053861
DBOW - NDCG	0.887297887051	0.887297887051	0.860783052554

TABLE II
BEST SCORES (WITHOUT STEMMING)

Measure/Algorithm	Logistic Regression	SVM	Neural Network
DM - MAP	0.724765367266	0.724112300545	0.715448394264
DBOW - MAP	0.727152763751	0.726495657443	0.734235507978

VI. Graphs

On observing the pictorial representation, we see the comparison between the two architectures and also the comparison between algorithms in the two architectures and also the comparison between architectures, along with when (Porter) Stemming is applied or not.

VII. Results

MAP is used for Binary relevance. Whereas the NDCG, is designed for situations of non-binary notions of relevance. Also, MAP is macro-averaged, i.e, each query counts equally. However, gain is discounted, for lower ranks in the case of NDCG value.

So, on obtaining the results of the NDCG and MAP values of different classifiers such as SVM, Logistic Regression and

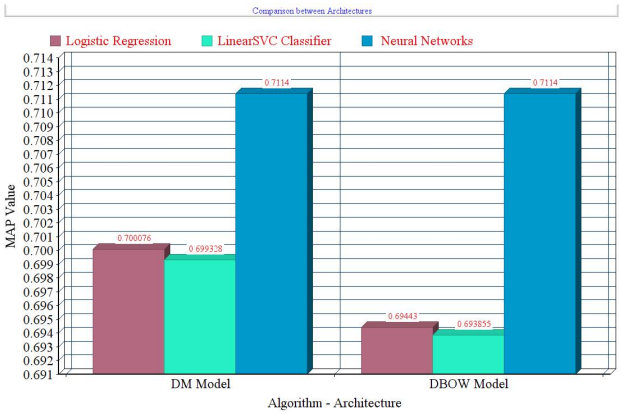

Fig. 3. Comparison Graphs MAP - Different Architectures (ACM)[with stemming]

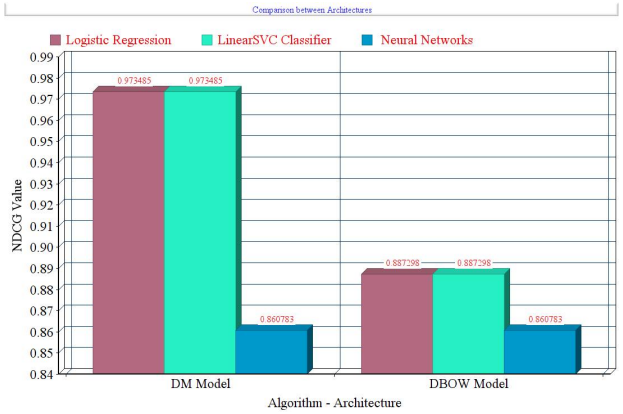

Fig. 4. Comparison Graphs NDCG - Different Architectures(ACM)[with stemming]

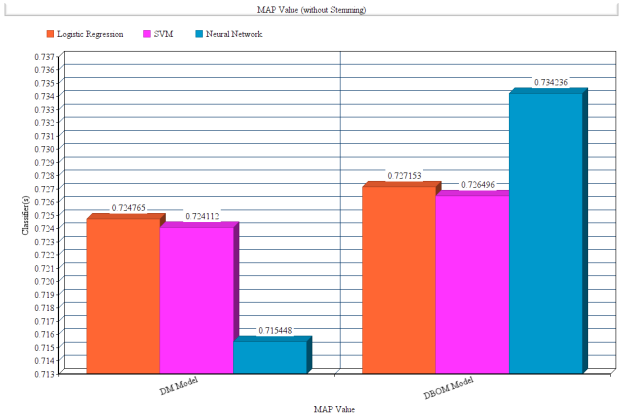

Fig. 5. Comparison Graphs MAP - Different Architectures (ACM)[without stemming]

Neural Network, we see that the Neural Network, seems to be performing the best, on average.

We see that in general better results are obtained using DM Model, when Stemming is applied. However, without Stemming the Evaluation Parameter Scores increase significantly. Stemming increases recall while harming precision. For example, in the case of öperational and research änd öperating and system, an stemmer would generally convert operational and operating to öperïn both cases, that may lead to catastrophic results.

From the results, it is safe to say that additional hidden layer(s) and the tensor models improve upon the original PV-DM and PV-DBOM formulation. With Stemming, DM Model was shown to perform marginally better, whereas, in the case of without applying stemming, DBOW model was found to perform better. This may be because of the number of dimensions used, as with Stemming was tested with 300 Dimensions as compared with 100 dimensions used in the case where Stemming was applied.

ACKNOWLEDGMENT

We would like thank our Dr. Priya Radhakrishna for mentoring our project and introducing us to the new state-of-art technologies and helping us at every stage of this project. We would also like to thank Dr. Vasudev Varma, our course instructor for Information Retrieval and Extraction. This project is part of this course.

REFERENCES

[1] https://github.com/piskvorky/gensim
[2] https://radimrehurek.com/gensim/models/doc2vec.html
[3] http://rare-technologies.com/Doc2Vec-tutorial
Quoc V. Le, and Tomas Mikolov, Distributed Representations of Sentences and Documents ICML, 2014
Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.