International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

Large Scale Hierarchical Natural Language Text Classification using Deep Graph-CNN

N. Rohini[1], Neethu Subramanian[2], C. Sunitha[3], Amal Ganesh[4]
[1,2]PG Scholar, Dept. of Computer Science & Engg., Vidya Academy of Science & Technology, Thrissur, India
[3]Associate Professor, Dept. of Computer Science & Engg., Vidya Academy of Science & Tech., Thrissur, India
[3]Assistant Professor, Dept. of Computer Science & Engg., Vidya Academy of Science & Tech., Thrissur, India

***Abstract*: Text classification is the task of classifying un- labelled natural language documents into a predefined set of categories. Task of classification can depend on various factors like structure of data, size of data processed etc. Many real world problems however need to consider a huge amount of data to be classified from many sources. In large scale text classification, no. of classes can run into thousands and in some cases, each document may only belong to single class while in others to more than one. Hierarchical relations can offer extra information to a classification system which can improve scalability and accuracy. The work aims at presenting a new Graph based document representation and classification based on Convolutional Neural Network architecture.**

***Keywords*: natural language processing, large scale text classification, deep graph convolutional neural network, hierarchical relation, multilabel classification**

1. Introduction

Text classification deals with the problem of assigning documents to a predefined set of classes. For example, consider the case of binary classification where there is just one class and each document either belongs to it or not. Spam filtering is such an example, where emails must be classified as desirable or not. In machine learning a classifier can be trained using positive and negative instances in order to perform the classification automatically, but was found rarely to be 100% correct even in the simplest case. In large scale text classification, the number of classes can run into thousands. In some cases, each document may only belong to a single class (single-label classification), while in others to more than one (multi-label classification). Also sometimes, the volume of documents to be processed is also very large (hundreds of thousands or even millions), leading to a high vocabulary (unique different words in the documents, also known as types). One of the aspect of Multi label classification is that the classes are connected each other. Thus this can be a parent child relation composing a hierarchy. A class taxonomy is important for two reasons. First it offers extra information to a classification system, which in theory can be exploited either to improve scalability or to improve accuracy or even both. Second it can affect the evaluation of a classification system. One of the main problems that was faced in the early stages of the research was the unavailability of open-access to large-scale hierar- chical text datasets. This fact prompted the researchers to initiate a series of challenges on Large Scale Hierarchical Text Classification (LSHTC). The LSHTC challenges is conducted in four editions from December 2009 until 2014, which attracted more than 150 teams from around the world (USA, Europe and Asia). The results of the challenges were presented in subsequent workshops at the conferences ECIR 2010, ECML 2011, ECML 2012 and WSDM 2014 and the discoveries of the workshops were presented. The aim of The LSHTC initiative was to assess the performance of classification systems in large-scale classification using a large number of classes. It included tracks of various scales in terms of classes, multi-task classification and unsupervised classification. Two corpora from Wikipedia and from the ODP Web directory data were mainly used in the workshop which may be downloaded from the permanent LSHTC website. This motivated many researchers to do extended works in the text classification area.

2. Graph CNN for text classification

Studies have been done on applying neural networks on sequence of words for the task of classification of text for several years. However, since text contains a relationship among their words, latest studies are based on graph convolutional neural networks.

A. *Graphical representation Of Document*

The task of representing a given document in a form which is suitable for data mining system is referred as document representation. Graph based portrayal is proper method for representation of content record and enhanced the aftereffect of investigation over customary model for various content applications. Document is modeled as Graph where term represented by vertices and relation between terms is represented by edges:

G = {Vertex, Edge Relation}

There are generally five different types of vertices in the Graph representation: Vertex = {F,S,P,D,C} ,where F-Feature term,S-Sentence, P-Paragraph, D-Document, C-Concept.

Edge Relation = {Syntax, Statistical, Semantic}

Edge relations between two feature terms may be different on the context of Graph.

1. Word occurrence together in a sentence or paragraph or section or document.
2. Common words in a sentence or paragraph or section or document.
3. Co-occurrence on the fixed window of n words.
4. Semantic relation Words have similar meaning, words spelled same way but have different meaning, opposite words.

B. CNN For text classification

CNN have been used largely in image processing and was found almost succeeded in it. However, when CNN deals with NLP, for which the inputs are documents or sentences represented as a matrix, each row of the matrix corresponds to one token, typically a word, but it could be a character. That is, each row is vector that represents a word. Typically, these vectors are word embeddings (low- dimensional representations), but they could also be one-hot vectors that index the word into a vocabulary. For a 10-word sentence using a 100-dimensional embedding we would have a 10100 matrix as our input.

For eg., consider a sentence classification using CNN model depicted in the figure 1[12], Here three filter region of sizes: 2, 3 and 4 are depicted, each of which has 2 filters. Feature maps of variable length are generated by performing filter convolution on the sentence matrix. Then 1-max pooling is performed over each map, i.e., the largest number from each feature map is recorded. Thus a univariate feature vector is generated from all six maps, and these 6 features are concatenated to form a feature vector for the penultimate layer. The final softmax layer then receives this feature vector as input and uses it to classify the sentence; here binary classification is assumed and hence depict two possible output states

3. Related works

Automatic document categorization has become more challenging over the last several few years since the corpus size and fields and sub fields of documents have been in- creased. Liang Yao, Chengsheng Mao, Yuan Luo from North- western University Chicago IL 60611 done their research on Graph Convolutional Networks for Text Classification [1]. Their text GCN is initialized with one-hot representation for word and document, it then jointly learns the embeddings for both words and documents, as supervised by the known class labels for documents. GCN is a multilayer neural network. that operates directly on a graph and induces embedding vectors of nodes based on properties of their neighborhoods. They build a large and heterogeneous text graph which contains word nodes and document nodes so that global word co-occurrence can be explicitly modeled and graph convolution can be easily adapted. 5 datasets were used for their study such as 20 NG, R8, R52, Ohmsumed, MR. Their work proved that Text GCN can capture global word co- occurrence information and utilize limited labeled documents well. A simple two-layer Text GCN demonstrates promising results by outperforming numerous state-of-the-art methods on multiple benchmark datasets.

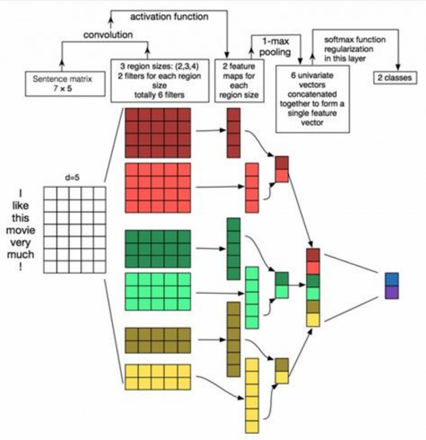
activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Fig. 1. Illustration of a Convolutional Neural Network (CNN) architecture for sentence classification

Deep Neural Network for Hierarchical Extreme Multi- Label Text Classification [2] by Francesco Gargiulo, Stefano Silvestri, Mario Ciampi, Giuseppe De Pietro from University of Naples is an article published in the journal, “Applied Soft Computing” presents a methodology named Hierarchical Label Set Expansion (HLSE). It is used to regularize the data labels, and an analysis of the impact of different Word Embedding (WE) models that explicitly incorporate grammatical and syntactic features. their work was evaluated using PubMed scientific articles collection, where a multi- class and multi-label text classification problem is defined with the Medical Subject Headings (MeSH) a hierarchical set of 27, 775 classes. Their experiments proved a direct correlation between the vector size and the corresponding number of network parameters label set.

Recently, techniques for applying convolutional neural networks to graph-structured data have emerged. Bayesian graph convolutional neural networks for semi-supervised classification [3] was a method proposed by Yingxue Zhang, Soumya sundar Pal, Mark Coates and Deniz stebay from Montreal, QC Canada. Their focus was on a specific random graph model, the assortative mixed membership block model. which address the task of semi-supervised classification of nodes and examine the resilience of the derived architecture to random perturbations of the graph topology. They suggested that the performance of the GCNNs can be improved by incorporating attention nodes leading to the graph attention network (GAT). A GCNN performs the task of classification by performing graph convolution operations within a neural network architecture. Their studies have a clear conclusion that the Bayesian formulation can provide better performance when there are very few labels available during the training process. Typical consideration of graph topology is the hierarchies. Hierarchical Attention Networks for Document Classification

[4] proposed by Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy from Carnegie Mellon University and Microsoft Research, Redmond has two distinctive characteristics. One is that it has a hierarchical structure which reflects the hierarchical structure of the input document and second is that 2 levels of attention networks are applied both at sentence level and word level. This will help in differentially to highly and lower important matter while creating the text representation. 6 different datasets were used for the evaluation of the system to prove that their proposed method was better than many of the previous methods.

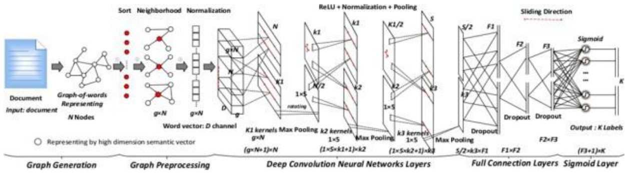

Fig. 2. Deep Graph CNN Architecture

4. Proposed method

In this paper we propose a Deep Graph CNN (DGCNN) based model for the classification of natural language documents. Here the long documents are first converted into graph format based on the word co-occurrence mentioned in section 2.1.i.e, if 2 words occur in the same window, we consider there is an edge between them. Then each node of the graph is converted to vectors using a pretrained neural network called Word2Vec which makes our input as a graph of vectors. The typical architecture of the proposed DGCNN is depicted in fig. 2.

1. *Graph Generation:* Here we will observe how we convert a document into a graph. We define the set of labels as L= {l_i i=1,2,....K} where K refers to the number of labels. Since the proposed method considers the hierarchical relation among the documents, we assume a parent-child relation among the labels. Thus we also denote label $l^{(j)}$(j=1,...k_i) as children of l_i. Also k_i is the number of children of K_i. Stanford Core NLP tool is used to split each document in to sentences and then to tokens. Then a fixed sized sliding window (here size=3) is used to get the word co-occurrences. We apply convolution masks on the sub-graphs of the word co-occurrence graph. For consistency, we normalize the sub- graphs. The following two steps are done to obtain the normalized sungraph so that it can be given for further processing.
    - Nodes in the graph are sorted according to their degree and degrees as calculated based on the number of neighborhood nodes. Then we choose the N most relevant nodes and then Breadth First Search algorithm is applied on each of these N nodes. Thus we get N Subgraphs of size g which is pre-defined.
    - Now to make the subgraphs consistent, we normalize the subgraphs. The labelling starts from roots and we apply BFS to rank the nodes. If the subgraphs have more than g nodes, then we use rank filters on them. If less than g nodes, then we add some dummy nodes to disconnected nodes.
2. *Graph of embeddings:* Rather than representing each node in a graph as an ID, more semantic content can be captured is they are represented as embeddings. Wored2Vec trained using large corpus is used to create the graph of embedding's.
3. *Convolutional layers:* Feature space of size N*g*D is the input to the first convolutional layer. Here N is the number of selected normalized subgraphs, g is the size of subgraphs and D is the dimension of word embeddings. The N*g*D input tensor is convolved using a g D kernel which serves as a composition of each subgraph.

Then for all the N input sub-graphs, we use k1 kernels to convolve the same way to generate a N*k1 matrix. Then max-pooling layer is applied to generate N/2*k1 matrix. This means we select half of the subgraphs which can further be used for better processing.

Now the N/2 subgraphs are convolved using 5*1 kernel which will generate the k2 dimention to obtain higher level of semantics. Thus a k1*k2 matrix is created. Then we use a max polling layer after k1*k2 matrix to generate a k1/2*k2 matrix, followed by k3 1*5 kernel. The ReLu activation function is used throughout the process for speed up and to avoid overfitting. The output layers of the proposed DGCNN architecture is 3 fully connected layers which mostly deals with non- linearity in classification. Here we apply dropout to avoid overfitting where the dropout rate is set to be 0.5.

4. *Dataset:* Any supervised text classification experiments are said to be successful by the accuracy of the prediction. However, the accuracy depends a lot more on the dataset used for the training of the system. Here we have used 2 datasets. One is an IMDB movie genre dataset and other is the 20NewsGroup dataset both of which are available freely. IMDB dataset contained 4456 documents collected from IMDB website and they are labelled into 19 classes. 20NG is a collection of 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.

5. Observation

The work aimed at labelling of large scale text data in natural language based on their hierarchical relationship. The evaluation was done on 2 classic datasets. IMDB dataset was not uniform, i.e., the no. of documents belonging to each class was not uniform. However, in 20NG, the distribution of documents was almost uniform. Also 20NG had more no. of

Table 1
Observation table

Dataset	No. of documents	Distribution of documents	Accuracy
IMDB	4456	Non-uniform	62.3%
20NG	Approx. 20,000	Uniform	83.4%

documents well spitted into both training and testing however IMDB had only lesser no. of documents samples. It was observed that the system accuracy on a more uniform 20NG dataset was improved compared to a lesser uniform IMDB dataset. Observation is tabulated in table 1.

6. Conclusion

Text Classification assigns one or more classes to a document according to their content. Classes are automatically selected from a previously established classes to make the process superfast and efficient. Deep learning has been used extensively in natural language processing (NLP) because it is well suited for learning the complex underlying structure of a sentence and semantic proximity of various words. Also graph representation of words can capture more semantic content that word representation of natural language data. Also considering the hierarchical relationship among labels can contribute a lot to the efficiency of multi-label classification to a greater heights.

References

[1] Liang Yao, Chengsheng Mao, Yuan Luo, Northwestern University Chicago IL 60611 Graph Convolutional Networks for Text Classification, Nov. 2018.

[2] Francesco Gargiulo, Stefano Silvestri, Mario Ciampi, Giuseppe De Pietro, Deep neural network for hierarchical extreme multi-label text classification, Applied Soft Computing Journal, March 2019.

[3] Yingxue Zhang, Soumyasundar Pal, Mark Coates, Deniz stebay, Montreal, QC Canada Bayesian graph convolutional neural networks for semi-supervised classification, 2018.

[4] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy, Carnegie Mellon University Hierarchical Attention Networks for Document Classification, June 12-17, 2016.

[5] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang ,Yangqiu Song, and Qiang YangLarge-Scale Hierar- chical Text Classification with Recursively Regularized Deep Graph-CNN: International World Wide Web Conference Committee), pub- lished under Creative Commons CC BY 4.0 License, ACM, ISBN 978-1-4503-5639-8/18/04.

[6] Rajni Jindal, Ruchika Malhotra, Abha Jain, Techniques for text classification: Literature review and current trends: Webology, Volume 12, Number 2, December, 2015.

[7] Jian-lin LI, A Text Classification Algorithm Based on PCA:2017 2nd International Conference on Computer Science and Technology (CST 2017)

[8] Ksh. Naresh Kumar Singh, H. Mamata Devi, Anjana Kakoti Mahanta, Document representation techniques and their effect on the document Clustering and Classification: A Review: International Journal of Advanced Research in Computer Science, Volume 8, No. 5, May-June 2017.

[9] Wang, B.K., Huang, Y.F., Yang, W.X., Li, X Short text classification based on strong feature thesaurus: Journal of Zhejiang University Science C (2012).

[10] Xu, J.S, “A new method of text categorization,” Proceedings of the Sixth IEEE International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August (2007).

[11] Yang, Y., Pedersen, J. O, “A comparative study on feature selection in text categorization,” in proceedings of the Fourteenth International Conference on Machine Learning ICM, 412-420:

[12] Zifeng, C., Baowen, X., Weifeng, Z., Junling, X, “A new approach of feature selection for text categorization,” in Wuhan University Journal of Natural Sciences, 11(5), 1335-1339, 2006.

[13] Mikael Henaff, Joan Bruna, and Yann LeCun, “Deep Convolutional Networks on Graph-Structured Data,” 2015.

[14] Yoon Kim, “Convolutional Neural Networks for Sentence Classification,” 2014.

[15] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao, “Recurrent Convolutional Neural Networks for Text Classification,” in AAAI. 2267-2273, 2015.

[16] Sam Scott and Stan Matwin, “Feature Engineering for Text Classification,” in ICML. 379-388, 1999.

[17] Min-Ling Zhang and Zhi-Hua Zhou, “A review on multi-label learning algorithms,” in IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, pp. 1819-1837, 2014.