2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) | 978-1-6654-2642-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/I-SMAC52330.2021.9641021

TOPIC RECOGNITION AND CORRELATION ANALYSIS OF ARTICLES IN COMPUTER SCIENCE

Hitha K C
PG Student, Dept. of Computer Science and Eng.
NSS College of Engineering, Palakkad

Kiran V K
Asst. Professor, Dept. of Computer Science and Eng.
NSS College of Engineering, Palakkad

***Abstract*— Topic identification and similarity detection are two related essential task in data mining, information retrieval, and bibliometric data analysis, which aims to identify significant topics and to find similarity between text collections.It is an essential activity to identify research papers according to their research topics to enhance their retrievability, help create smart analytics, and promote a range of approaches to evaluating the research environment and making sense of it.The proposed frame work deals with three main steps: text extraction, topic identification, and similarity detection.The PyPDF2 module is used to extract text from pdf file. CSO classifier is used for topic identification and similarity between documents is calculated using different models, such as Tf-Idf, Bert, Glove, Word2vec, and Doc2vec.and compared these models with respect to cosine similarity and Eucleadian distance obtained from these models.**

***Index Terms*— PyPDF2 module, CSO Classifier, Tf-Idf, Bert,Glove, Word2Vec, Doc2Vec**

I. INTRODUCTION

Topic analysis is a Natural Language Processing (NLP) technique that allows us to automatically extract meaning from text by identifying recurrent themes or topics.Its merely too difficult to perform manually when dealing with large volume of text data. It's also time consuming,Inconvenient, and costly. Manually sorting through large amounts of data is more likely to lead to mistakes and inconsistencies. Plus, it doesn't scale well.

Text mining is a computational field of study that explores and analyses huge volumes of unstructured text material using tools and techniques from machine learning, data mining, and statistics[8]. To apply text mining methods to text collections, conventional natural language processing-based preprocessing tasks must be used, such as the identification of similar terms, noise removal, tokenization, normalization, stemming,lemmatization, stop-words removal, sentence segmentation and word standadization.The two most common approaches for topic analysis with machine learning are NLP topic modeling and NLP topic classification.

Topic modeling is an unsupervised machine learning technique.This implies it can infer patterns and group similar phrases without the requirement to specify subject tags or train data. This sort of method can be implemented fast and simply, but it has one drawback: it is inexact. When you have a set of text documents and wish to figure out what subjects they cover and arrange them by those topics, you use topic modelling.

Text classification, or topic extraction from text, on the other hand, necessitates prior knowledge of a text's subjects in order to train a topic classifier, which necessitates tagging data. Although topic classifiers require an extra step, they pay off in the long run and are far more exact than clustering approaches.Topic classification, unlike topic modelling, assumes that you already know what your subjects are.

Word embeddings are a form of word representation that allows for the depiction of words with comparable meanings. They are a distributed representation for text that may be one of the fundamental breakthroughs in deep learning approaches' outstanding performance on difficult natural language processing tasks.Word2Vec is a statistical approach for learning a solitary word embedding from a text corpus quickly and effectively. The Global Vectors for Word Representation, or GloVe, technique is an extension of Pennington's et alword2vec .'s approach for quickly learning word vectors. GloVe is a method for combining global statistics from matrix factorization techniques like LSA with local context-based learning from word2vec. Jeffrey Pennington et al. introduced the Global Vectors for Word Representation (GloVe) approach for learning vector representations of words quickly.. GloVe is a method for combining global statistics from matrix factorization techniques like LSA with local context-based learning from word2vec.Transformer, an attention mechanism that learns contextual relationships between words (or sub-words) in a text, is used by BERT.

In this paper Classified research papers according to their research topics according their research topics using CSO(computer science ontology) classifier. It finds all the explicitly mentioned topics in the ontology, identifying semantically related topics, returned union of syntactic and semantically generated topics, finally enriches the topics using super areas of topics according to CSO[1].Similarity between documents is calculated using different models, such as Tf-Idf, Bert, Glove, Word2vec, and Doc2vec.and compared these models with respect to cosine similarity and Eucleadian distance obtained from these models.

The paper is organized as follows.In section 2, the proposed system is described in detail. The outcomes of the experiments are detailed in section 3. Finally, section 4 provides a conclusion.

II. METHODOLOGY

There are mainly three stages text extraction, topic recognition and topic similarty.The PyPDF2 module was used to extract the topics. The cso classifier is used to recognise topics, and to find similarity between articles the Eucleadean distance and cosine similarity are calculated and evaluated using several embedding models such as Tf-Idf, Bert, Glove, Word2Vec, and Doc2Vec. The proposed system is depicted in Figure 1.

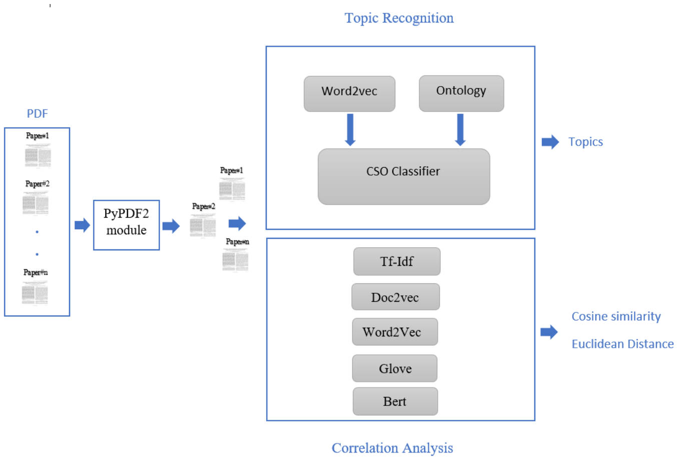

Fig. 1: Proposed system

A. *Topic Recognition*

A pdf file containing abstract, title, and keywords of n number of research papers given as input to the PyPDF2 module.The Python package PyPDF2 extracts text from PDF files.For topic identification, text comprising title, abstract, and keywords is input into the CSO classifier.The CSO Classifier is a innovative system that takes the text from a research paper's abstract, title, and keywords as input and returns a collection of relevant topics from Computer science ontology. It is divided into two sections: the syntactic module and the semantic module.The syntactic module parses the input documents and detects CSO topics that are mentioned directly in the text. The semantic module identifies potential phrases using part-of-speech tagging, then utilises word embeddings to infer semantically related subjects. Finally, the CSO Classifier integrates and enriches the findings of these two modules by incorporating appropriate super-areas.

B. *Correlation analysis*

Similarity between documents is calculated using different models, such as Tf-Idf, Bert, Glove, Word2vec, and Doc2vec.and compared these models with respect to cosine similarity and Eucleadian distance obtained from these models.It first did some basic text cleaning, removing special characters, removing stop words, and converting everything to lowercase.After that, different word embedding models, such as tf-idf, word2vec, doc2vec, Glove, and Bert, are applied to these preprocessed text documents.

Tf-Idf:-Tf-idf is a combination of term frequency and inverse document frequency. Every word in the document is given a weight, which is computed using the frequency of that word in the text as well as the frequency of papers containing that word across the whole corpus of documents.

Word2vec:-Words are embedded in vector space using word2vec.Word2vec is a system that uses a text corpus as input and produces word embeddings as output. Continuous bag of words and continuous skip gram are the two major learning algorithms used in word2vec. Google provides a pre-trained embedding model. To begin, tokenize and pad each document to the same size. The pre-trained embeddings then loaded. A 300-dimensional vector represents each word. Every word in our document corpus may be converted into a 300-dimensional vector using this embedding technique. The varied lengths of the documents would also have a detrimental effect on such functions. 6 documents and padded each document to a maximum size of 64.Using the tf-idf weights to create a weighted average of word vectors may be a better way to go. This solves the problem of fluctuating length to some extent, but it does not preserve the semantic and contextual meaning of words.

GloVe:-Word Vectors at a Global Level GloVe is a word vector representation method that is learned unsupervised. Training is based on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations emphasise significant linear substructures of the word vector space. Stanford's pre-trained glove embeddings were employed. All of the stages would be the same as they were for word2vec embeddings. It's simply that we utilised the Glove pre-trained model in this situation. Because of the huge size of the embedding file, we're employing 100-dimension glove embeddings.

Doc2vec:-Doc2vec is an unsupervised learning algorithm that produces vector representations of sentences, paragraphs, and documents. This is an adaptation of Word2vec. Doc2vec is capable of converting a whole document into a vector. To build a document vector, we don't need to take an average of word vectors. I use gensim to train a Doc2vec model on corpus and create vector representations of documents.

Bert:-BERT (Bidirectional Encoder Representation from Transformers) is a Google-developed state-of-the-art approach for natural language processing pre-training. Unlabeled text, such as Wikipedia and the Book corpus, is used to train BERT. To learn word embeddings, BERT employs transformer architecture, an attention model. Masked Language Modelling (MLM) and Next Sentence Prediction are the two pre-training processes in BERT (NSP). Text is represented using three embeddings in BERT training: token Embeddings, segment Embeddings, and position Embeddings. To incorporate our corpus, I utilised a pre-trained BERT model from Huggingface. The BERT basic model, which contains 12 layers (transformer blocks), 12 attention heads, 110 million parameters, and a hidden size of 768, is now being loaded. Finally, pairwise similarities are calculated using cosine and Euclidean distance.

III. EXPERIMENTAL RESULTS

Topic identification is done using CSO classifier and pypdf2 module. Pypdf2 modlue extracted text from pdf file. Classified research papers according to their research topic using CSO classifier.Set of topics identified is printed. Results include fist it return syntactic topics that are explicitly referred in the computer science ontology. Then it return semantic topics after that classifier return union of syntactic and semantic topics. finally it return enhanced topics with super areas of Computer science ontology. Precision, recall, and f-measure are used to manually calculate the performance of this method. When categorising a given paper p, the values of precision pr(p) and recall re(p), as well as the f-measure, are calculated using the equations 1, 2, 3 below. where c(p) is the number of topics returned by the classifier, and g(p) is the number of topics manually returned for that article. The harmonic mean of precision and recall is the f-measure (F1). Table 1 depicts the system's performance.
Precision:

$$pr(p) = \frac{c(p) \cap g(p)}{c(p)} \tag{1}$$

Recall:

$$re(p) = \frac{c(p) \cap g(p)}{g(p)} \tag{2}$$

F-Measure:-

$$H = \frac{2 * pr(p) * re(p)}{pr(p) + re(p)} \tag{3}$$

TABLE I: Performance of this system.

PRECISION	RECALL	F-MEASURE
72.58%	74.38%	73.47%

Using several embedding models such as Tf-Idf, Word2vec, Bert, Glove, and Doc2vec, cosine similarity between documents is computed.The glove embedding model performs better, whereas the Tf-Idf model performs worse.Word2vec and Bert produce nearly identical results.

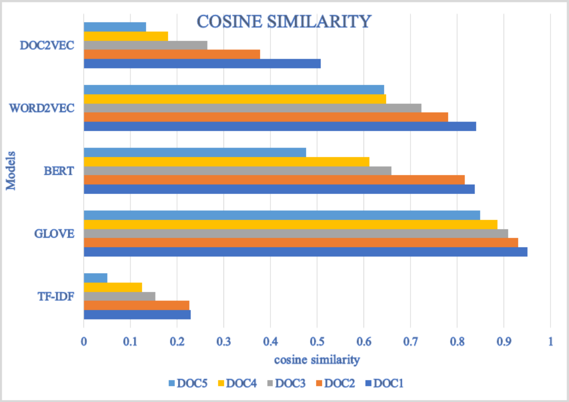

Fig. 2: cosine similarity of different models

Using several embedding models such as Tf-Idf, Word2vec, Bert, Glove, and Doc2vec, Euclidean distance between documents is computed.The Bert embedding model performs better, whereas the Tf-Idf and Word2vec models performs worse.Doc2vec and Glove models produce nearly identical results.

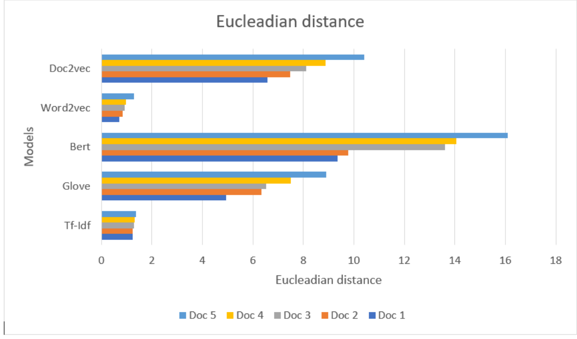

Fig. 3: Euclidean distance of different models

IV. CONCLUSION

Topic identification and similarity detection is a significant task for identifying research articles based on their research topics to improve retrievability, aid in the creation of smart analytics, and encourage a variety of ways to assessing and making sense of the research environment. The topic analysis model allows us to filter through enormous amounts of data and quickly discover the most common topics and similarities across documents. Topic analysis can be used for the following applications Ontology Creation, Semantic Search, Theme Generation for News, Resume analytics, Content Overloading Problem solution, User Interest Discovery, Product catalogue generation, Clustering Sampling, Intent Classification. PyPDF2 module extracted text from pdf file. The CSO Classifier examines research paper content on a syntactic and semantic level (title, abstract, and keywords) and produces a list of relevant CSO research topics. Performance of this system evaluated manually.Using several embedding models such as Tf-Idf, Word2vec, Bert, Glove, and Doc2vec, cosine similarity and Euclidean distance between documents is computed.The glove embedding model performs better in terms of cosine similarity, but the Tf-Idf model performs worse. The results of Word2vec and Bert are nearly identical.In the case of Eucleadian distance, the Bert embedding model outperforms the Tf-Idf and Word2vec models in terms of performance. The Doc2vec and Glove models yield findings that are nearly comparable. The proposed system can be extended in other domains of the Ontology.

REFERENCES

[1] Angelo A. Salatino, et.al;"The CSO Classifier: Ontology-Driven Detection of Research Topics in Scholarly Articles ";pp 296-311,2019, Springer.

[2] Raghavendra Pappagar,et.l;"joint verification identification in end to end multiscale cnn framework for topic identification",2018,IEEE.

[3] Zakky Nilem Sanjifa, et.al;"Community Feedback Analysis Using Latent Semantic Analysis (LSA) To Support Smart Government",2019 International Seminar on Intelligent Technology and Its Applications (ISITIA),2019,IEEE.

[4] David M Blei, et.al "Latent dirichlet allocation," Journal of machine Learning research, vol. 3, no. Jan, pp. 993–1022, 2003.

[5] ytug onan, et.al; "Two-Stage Topic Extraction Model for Bibliometric Data Analysis Based on Word Embeddings and Clustering" :special section on artificial intelligence and cognitive computing for communication and network , volume-7, 2019, IEEE.

[6] guixian xu, et.l; "Research on Topic Detection and Tracking for Online News Texts";special section on artificial intelligence and cognitive computing for communication and network,2019,IEEE.

[7] K.Hofmann, et.al;"A comparative study of features for key phrase extraction in scientific literature," in Proc. 18th ACM Conf. Inf. Knowl. Manage., Hong Kong, pp. 1725–1728,2009,ACM.

[8] A. Bagheri, M. Saraee, and F. de Jong, "ADM-LDA: An aspect detection model based on topic modelling using the structure of review sentences,"J. Inf. Sci., vol. 40, no. 5, pp. 621–636, 2014.

[9]

[10] A. Bagheri, et.al ;"ADM-LDA: An aspect detection model based on topic modelling using the structure of review sentences," J. Inf. Sci., vol. 40, no. 5, pp. 621–636, 2014, JIS.

[11] A. Bougouin, et.al; "TopicRank: Graph-based topic ranking for Keyphrase extraction," in Proc. Int. Joint Conf. Natural Lang. Process., Nagoya, Japan, pp. 543–551, 2013.

[12] A. Ferrara, et.al; "Ten challenges in modeling bibliographic data for bibliometric analysis," Scientometrics, vol. 93, no. 3, pp. 765–785, Dec. 2012, Springer.

[13] Y. Zhang, et.al; "Term clumping' for technical intelligence: A case study on dye-sensitized solar cells," Technol. Forecasting Social Change, vol.85,pp. 26–39, 2014, Elsevier.

[14] Haotian Xu, et.al; "Text classification with topic-based word embedding and convolutional neural networks.," in BCB, pp. 88–97, 2016, ACM.

[15] Yang Liu, et.al "Topical word embeddings.," in AAAI15: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2418–2424, 2015.

[16] Dheeraj Mekala, et.al; "Scdv: Sparse composite document vectors using soft clustering over distributional representations," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 670–680, 2017.

[17] A. Onan, et.al "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," Inf. Process. Manage., vol. 53, no. 4, pp. 814–833, 2017, Elsevier.

[18] A. Onan, et.al "Ensemble of keyword extraction methods and classifiers in text classification," Expert Syst. With Appl., vol. 57, pp. 232–247, 2016, Elsevier.

[19] T. Velden, et.al;"Comparison of topic extraction approaches and their results," Scientometrics, vol. 111, no. 2, pp. 1169–1221, 2017, Springer.

[20] Osborne F., et.al;" Mining semantic relations between research areas. In: Cudré Mauroux, P., et al. (eds.) ISWC 2012. Lecture Notes in Computer Science", vol. 7649, pp. 410-426, 2012, Springer.

[21] Manoharan, J. Samuel. "Capsule Network Algorithm for Performance Optimization of Text Classification" Journal of Soft Computing Paradigm (JSCP) 3, no. 01 (2021): 1-9.

[22] Mugunthan, S. R., and T. Vijayakumar. "Design of Improved Version of Sigmoidal Function with Biases for Classification Task in ELM Domain" Journal of Soft Computing Paradigm (JSCP) 3, no. 02 (2021): 70-82.

[23] Smys, S., and Jennifer S. Raj. "Analysis of Deep Learning Techniques for Early Detection of Depression on Social Media Network-A Comparative Study" Journal of trends in Computer Science and Smart technology (TCSST) 3, no. 01 (2021): 24-39.