Journal of Current Research in Engineering and Science
JCRES
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 7-Issue 1, January 2024
Paper: 50

An Exploration Performance Efficiency and The Knowledge Enhanced Text Classification using Probabilistic Latent Semantic Analysis

Raja K
Computer Science and Engineering
SRM Institute of Science
and Technology,
Chennai, India
drkrajamit@gmail.com

Shiny Duela J
Computer Science and Engineering
SRM Institute of Science
and Technology,
Chennai, India
shinyduj@srmist.edu.in

Sayanti Roy
Computer Science and Engineering
SRM Institute of Science
and Technology
Chennai, India
sr6864@srmist.edu.in

Mopuru Sahithi Reddy
Computer Science and Engineering
SRM Institute of Science
and Technology
Chennai, India
ms7058@srmist.edu.in

Jibendu Paul
Computer Science and Engineering
SRM Institute of Science
and Technology
Chennai, India
jp7060@srmist.edu.in

***Abstract*- The unusually long feature sparsity and considerable ambiguity in the text make classification jobs extremely difficult. Recent years have seen a significant increase in study and interest in prompt learning as a useful technique for fine-tuning models for certain downstream tasks. The primary idea behind quick learning is to use the template as input to change text classification tasks into cloze-style activities that are equal to those tasks. Unfortunately, the majority of prompt-learning techniques manually extend label words or solely take the class name into account when incorporating information in cloze-style prediction, which inevitably results in bias and omissions in text classification problems. In this study, we offer a simple text categorization strategy that makes use of prompt learning based on knowledgeable expansion. The technique can increase the label words space while considering both the text itself and the class name, considering the properties of the text. The open Knowledge Graph is specifically used to extract the top N concepts connected to the entity in the text. According to experimental findings, our technique significantly outperforms previous fine-tuning prompt-learning models.**

***Keywords*- *Text classification, labeling text, word categorization, decision making, improved traceability*.**

I. INTRODUCTION

Due to the rapid development of Web services, short texts are being published at a pace that highlights both the value of learning activities and the difficulties posed by the basic characteristics of texts, such as their extremely short duration, a small number of features, and a high degree of ambiguity. The advancements in short text data processing have far-reaching effects on practical applications like Twitter, Facebook, and Microblog. In recent decades, short-text classification has received a great deal of attention and research from many fields. The approaches now in use for short text classification can be loosely divided into two classes: methods using only one source and methods using external knowledge bases. The rules or statistical data hidden in the current short texts are inserted into the feature space by the single source-based approaches. Although approaches based on external knowledge have been implemented widely in recent years, the feature space has been expanded by an external open knowledge base. The sole source methods still struggle with the severe data sparsity problem Compared with the previous fine-tuning approaches no additional neural layer is needed for prompt learning and excellent performance has been achieved even in the scenario of few-shot or zero-shot learning. The mapping from label words in prompt learning (such as sports, association, basketball et at.) to the automatic selection of label words, also known as the verbalizer, can effectively eliminate the discrepancy between text and label space, such as SPORTS [1]. It has been demonstrated that this method of building labeled word maps is effective in improving text classification performance. Other works attempt to expand label words for text classification and incorporate external knowledge. Since the special characteristics of short text, such as extremely short length, feature sparsity, and high ambiguity, are completely distinct from those of conventional text, a knowledgeable method only takes into account the class name and disregards the information about entities and concepts in the short text, resulting in unsatisfactory classification results.

II. Related Work

A. J Peng Wang, Yun Yan, et.al,. [1] The Characteristics of a Proactive Individual: centered on the forecast of proactive identities. Focused on short-answer questions and social media post content (Weibo) were gotten from 901 cluster-selected members, and specialists assessed the participants' proactive identity names. Back Vector Machine, XG Boost, K-Nearest Neighbors, Credulous Bayes, and Calculated Relapse were the five machine learning calculations utilized for classification. In expansion, progressive cross-validation and seven unmistakable pointers, such as Exactness, F1-Score (F1), Affectability (SEN), Specificity (SPE), Positive Prescient Esteem, Negative Prescient Esteem, and Zone beneath Bend (AUC), were utilized within the comprehensive assessment of the models. In conclusion, content mining innovation demonstrated to be exceptionally valuable in anticipating a person's proactive identity and, more particularly, in distinguishing individuals who have a low proactive identity. This may be exceptionally valuable for practicing career instruction in tall school and college. The most noteworthy exactness and specificity scores, individually, were 0.842 and 0.969.

B. Peng Wang, Xiang Ping Zhan, Mei Tian, et. al [2] Modern approaches were utilized to classify people based on proactive identity utilizing content mining innovation. The members were given proactive identity surveys and were inquired to supply reactions to four choices, the lion's share of which depended on self-reports. Moment, since the names of the tall and moo proactive identity categories compare to particular highlights with critical distinction, as it were the F-test was utilized for highlight determination in our past work. As a result, highlights can be recognized utilizing the F-test. In our unused work, the 2 test was too utilized for highlight choice in expansion to the F-test.

C. Lihui Zhang, Gancheng Zhu, and Peng Wang et. al. [3] They proposed combining subjective and objective estimation to survey career flexibility; that's, text-IRT mix strategy. By differentiating three analyzing techniques—, IRT, text-IRT combination method, and text categorization —we compare and differentiate the proposed strategy on three unmistakable approval sets with 300, 600, and 900 people. The result illustrates that, especially in a little test, the text-IRT combination method may be foremost viable for career flexibility expectations. The leading approach for huge bunches, especially when distinguishing people with moo career versatility, is content categorization. Illustration estimate affected precision, particularity, and the negative prescient upsides of content characterization, as well as the responsiveness of text IRT and IRT procedures.

D. Yuezhong Liu, Xiaoming Huang, Run Chen, Chaofan Wang Run Chen, and Shenggen Ju [4] Within the field of content classification, surprising results have been accomplished employing a crossover show that combines self-attention with a Convolutional Neural Organize (CNN). Content neighborhood semantics and worldwide representation play break even with parts for each input in past ponder. A brand-new CNN and self-attention half-breed demonstration are displayed in this paper. In challenging phonetic environments, MulCNN-Att can flexibly determine whether performance is more or less crucial for categorization tasks. We will get the foremost out of these two models in this way. Our crossbreed show consolidates multi-scale highlight consideration as well. The consideration can consequently select from writings compelling and task-friendly multi-gram highlights. In expansion, they create a novel misfortune work for fine-grained feeling assignments with the objectives of minimizing closest neighbor misclassification and expanding classification precision. Classification exactness makes strides by up to 2 rate focuses when compared to customary cross-breed models based on CNN and self-attention, as illustrated by tests.

E. Berihan R. Elemary, Hammam M. Abdelaal and Hassan A. Youness [5] Given the noteworthiness of the Prophet's Hadith to Muslims worldwide—it is the essential source of Islamic enactment and the moment the source of Islam after the Quran. Based on the content of hadith, the point of this investigation is to consequently classify hadith into different categories based on their substance. The target of this audit is to build a classifier to demonstrate can bunch and isolated hadith classifications, to anticipate pointing like supplication, fasting, and zakat; utilizing data mining and AI strategies. To make classification precision, numerous administered learning calculations and combination procedures just like the stacking calculation were utilized in this ponder. The three best classifiers were assessed: the Choice Tree (DT), Unpredictable Forest (RF), and Naive Bayes (NB), which finished with higher accuracy and came to up to 0.965%, 0.956, and 0.951% independently.

III. Existing Work

The address of how well prescient instability strategies work in real-world Normal Dialect Preparing, especially when it comes to content classification for numerous classes and

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 7-Issue 1, August 2023
Paper: 50

names. On six real-world content classification datasets, we conduct benchmarking tests utilizing –D convolutional neural systems and pre-trained transformers to observationally explore the reason that prevalent adaptable vulnerability makes Dropout thinks little of vulnerability. In light of later ponders on how outfits and variational Bayesian strategies explore the misfortune scene, we contend that vulnerability estimation comes about from combining back estimation strategies. By analyzing in-domain calibration, cross-domain classification, and novel lesson strength, we discover that our proposed strategy, which combines Concrete Dropout with Profound Outfit, performs way better indeed when the outfit measure is littler. The centrality of the ne-tuning dropout rate to the current content classification assignment, which impacts demonstrate vigor both separately and collectively, is bolstered by their discoveries. When a dissemination move from novel classes is anticipated, we discover that pre-trained transformers perform essentially more regrettably in an oddity discovery in removal, restricting the application of exchange learning. When endeavoring to optimize for a well-approximated (Bayesian or something else) back, current prescient instability strategies are ordinarily ineffectual and flawed. Be that as it may, since dependable robotization depends on joining and assessing the quality of vulnerability, viable and versatile arrangements are fundamental. In arrange for professionals to be able to depend on them securely and analysts to know which approaches to prioritize, vulnerability measurement requires a methodology for task-specific benchmarking. We have highlighted the understudied thinking about instability quality and show strength in practical NLP information dissemination by displaying experimental proof from benchmarking instability strategies in content classification. It is curious to note that, despite being as often as possible connected to the content classification errand, the common behavior of prescient instability strategies does not hold over different datasets. Whereas a few strategies are predominant, we are incapable to recognize a clear winning prescient vulnerability strategy generally.

IV. Proposed System

The Properties of an architecture diagram are used to bring out the finished groundwork of the software formation through relationships, problems, and boundaries amid sections are shown in fig. 4.1.

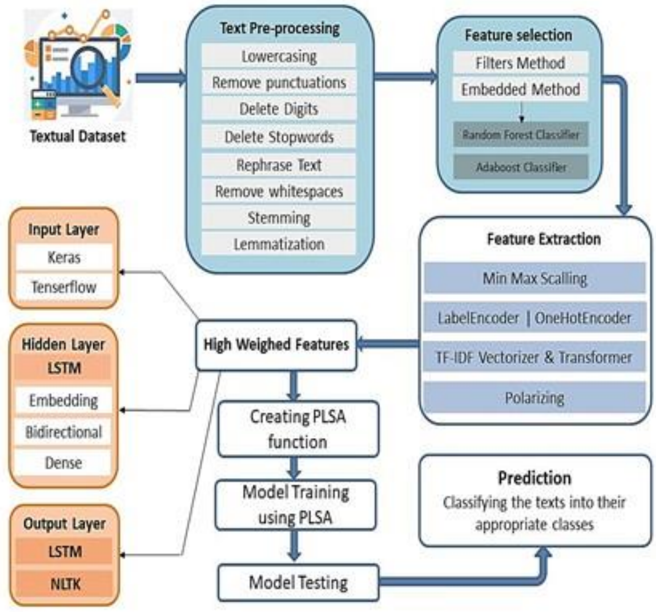

Fig. 4.1: Architecture Diagram of text classification using pLSA

The proposed work aims to investigate the performance and efficiency of pLSA for text classification tasks. In addition, the proposed work seeks to enhance the knowledge representation of pLSA through the use of domain-specific knowledge. The proposed system for text classification using pLSA would involve several components working together to investigate performance, efficiency, and classify text documents into relevant categories, and enhance the knowledge representation of pLSA through the use of domain-specific knowledge. The system would start with a pre-processing stage where the input text data is cleaned, tokenized, and converted into a bag-of-words representation. This pre-processed data would undergo feature extraction, then it is fed into a pLSA model, which would learn the underlying topics and their distribution in the text corpus.

Once the pLSA model has been trained, it would be used to classify new documents and investigate performance and efficiency with other classification algorithms. For each new document, its topic distribution would be inferred from the pLSA model. The document would then be classified into the most probable topic based on the topic probabilities.

The findings of this work could have implications for the development of more accurate and efficient text classification algorithms, particularly for domains where external knowledge can be leveraged to improve accuracy.

A detailed explanation of the architecture diagram is given as follows:

- *Text processing*: Natural language handling is the capacity of computers to get human dialect in a valuable way, whereas content handling as it alluded to the examination, control, and the era of content.

- *Feature Selection:* When making a predictive show, the method of lessening the number of input factors is known as a highlighted choice. Decreasing the number of input factors is alluring to both lower the computational taking toll on modeling and, in a few cases, boost the model's execution.
- *Feature Expansion*: Feature engineering, often referred to as feature expansion, is the act of generating new features (variables) from already-existing ones to enhance the functionality of a machine-learning model. Feature expansion can involve transforming variables, and creating interactions between variables, extracting relevant information from variables.
- *Model Training:* Model training is the process of building a Machine learning model by training it on a dataset. During model training, the algorithm learns to recognize patterns in the data by adjusting its parameters based on the input and output data. The aim of model training is to develop a model that can correctly forecast the results for brand-new, unforeseen data.
- *Analysis Model:* In text classification, an analysis model is a type of machine-learning model that is used to classify text documents into one or more predefined categories. Text categorization aims to give categories or labels to text documents relying on their content.

Algorithm: PLSA Algorithm

Input: Document-Term matrix D, Number of topics K
Output: Topic-term probabilities P(w|z), Document-topic probabilities P(z|d)

```
Initialize random values for P(w|z) and P(z|d)
Do until convergence:
  For each document d in D:
    Normalize P(z|d) = P(z|d) / sum(P(z|d))
    For each topic z:
      Update P(z|d) = (P(w|z) * P(z|d)) / sum(P(w'|z) *
P(z|d)) over all words w' in d
  For each topic z:
    Normalize P(w|z) = P(w|z) / sum(P(w|z))

Return the final values of P(w|z) and P(z|d)
```

Fig. 4.2: Pseudocode of plsa algorithm

V. Methodology

The proposed system will be implemented in the following modules:

A. Module 1- Collection Of Data:

A data set is a collection of connected data. A data set in the case of tabular data refers to one or more database tables, with each row referring to a single record in the related data set and each column referring to a distinct variable.

B. Module 2- Text Preprocessing:

Space or punctuation marks are used as tokens to represent a document. Special characters, numbers, and URLs are removed so that the document contains only words in the target language. Stop words are removed and stemming stems the remaining words. Unfiltered noisy text is filtered through a series of preprocessing steps. Noise or unfiltered text is corrupted messages or chats containing slang or trash phrases. Some of the most common pretreatment methods are stopped words, word filtering, Capitalization, Abbreviation, Noise removal, etc.

C. Module 3- Feature Extraction:

Any machine learning method must first convert text data into numerical vectors before processing it. Word embeddings use syntactic and semantic data to map vocabulary words to low-dimensional, dense, and real-valued vectors. Similar vectors are used to group words that originate from the same context. In our research, we place a strong emphasis on the word-level neurological model, which views words as the basic building blocks of text. Before we can feed text data into the model, we must convert each word in the text into a word vector. In contrast to one-hot encoding, word embeddings convert each word into low-dimensional and scattered representations. The word vectors generated by the word embedding, which is based on the distributional tenet that words that appear in related contexts have comparable meanings, may be used to visually compare the semantic similarity of words. We chose to include word embedding in our recommended method since it performs well on several NLP tasks.

D. Module 4- Model Training & Text Classification:

The data is trained with the pLSA model and an input, an output, and one or more hidden layers make up the proposed model's set of layers. The input layer gets input in the form of a matrix. Feature selection and dimensionality reduction are carried out through hidden layers. The input document is categorized by the output layer into one of the specified labels. We employ the most used performance metrics, including recall, accuracy, F-measure, and performance, to assess the classification performance of the proposed model and baseline classifiers The confusion matrix allows for the calculation of precision, recall, F-measure, and accuracy.

E. Module 5- Prediction:

After performing the algorithms, and training the datasets against the algorithm, Data has been classified into their respective classes, and the main words in the texts in the datasets have been classified into their respective classes.

VI. Working Process

A. Importing Packages:

At first, we import the necessary packages like numpy, pandas, matplotlib, seaborn, sklearn, LabelEncoder,

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 7-Issue 1, August 2023
Paper: 50

OneHotEncoder, TfidfVectorizer, TfidfTransformer, RandomForestClassifier, AdaBoostClassifier, and many more that we use in this project.

B. Collecting Dataset:

The first step in any text classification project is to collect data. The train , test , and validation datasets have been imported into the compiler for further processing. The datasets have a serial number, text, l1, l2, and l3 parameters for classification.

C. Histogram of the Train, Test, and Validation Datasets

The histogram has been plotted for token length against the number of tokens. The tokens or subsets of words taken from the datasets have been taken for the histogram analysis of the project. The dataset has been analyzed using the histogram reports. The analysis has been done for the text, l1, l2, and l3 data columns (fig. 6.1).

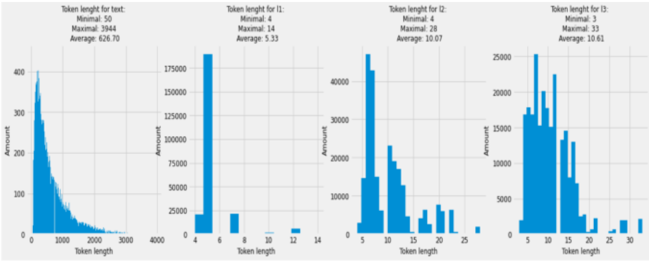

Fig. 6.1: The picture represents the Histogram of datasets.

D. Checking Datasets Category Wise Count:

Each root word has been classified into different classes and has been portrayed in a bar plot format to make it easy to understand the types of classes that have been used in the project. The l1(fig. 6.2), l2 (fig. 6.3), and l3 (fig. 6.4) categories of datasets have been represented using bar graphs as follows:

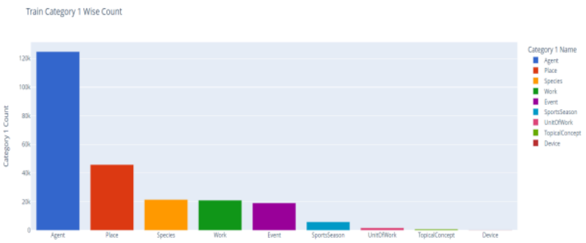

Fig. 6.2: The Bar graph represents Category 1- l1 count of data

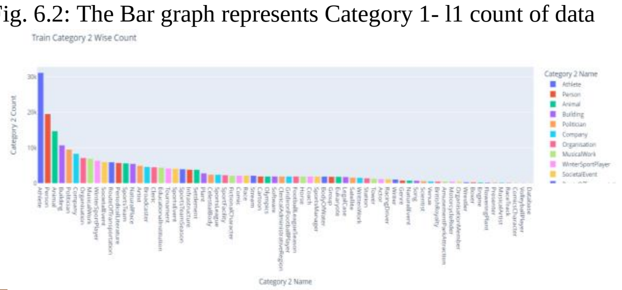

Fig. 6.3: The Bar graph represents Category 2- l2 count of data

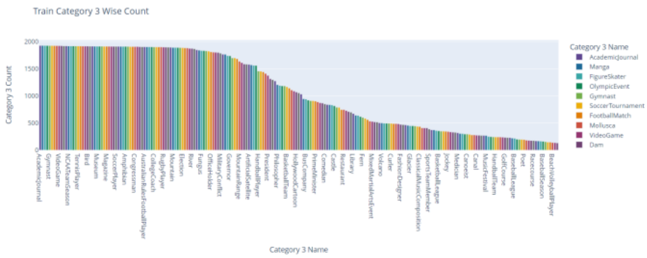

Fig. 6.4: The Bar graph represents Category 3- l3 count of data

E. Data Preprocessing

Once the data is collected, it needs to be pre-processed. This involves converting the text into a format that can be used by the model. Here the data has been pre-processed by removing the case-sensitive words, stop words, blank spaces, rephrasing texts, etc as per required in our project (fig. 6.5). The pre-processing process goes as follows:

Step 1: Define a class "text_preprocessing(df)".
Step 2: Under df take the lambda function as 'x' and specify the parameters for pre-processing.
Step 3: specify the length of 'x'.
Step 4: return df.
Step 5: apply the function to test, train, and validation datasets.

Fig. 6.5: The steps represent the Pre-processing function that has been used in the project.

F. Feature Extraction:

In this step, we extract features from the pre-processed text. The words here have been converted to vectors using the TfidfVectorizer, and TfidfTransformer.Here we are using word embeddings.

G. Model Training and Testing:

This step involves training the PLSA model on the extracted features to obtain the document-topic matrix and the topic-word matrix (fig. 6.6 and 6.7). The PLSA model essentially learns the underlying topics in the corpus of text and f1 score is calculated.

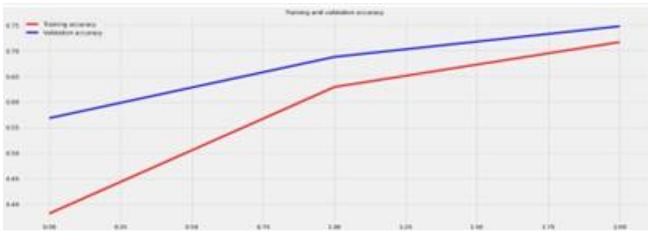

Fig. 6.6: Training and Validation Accuracy

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 7-Issue 1, August 2023
Paper: 50

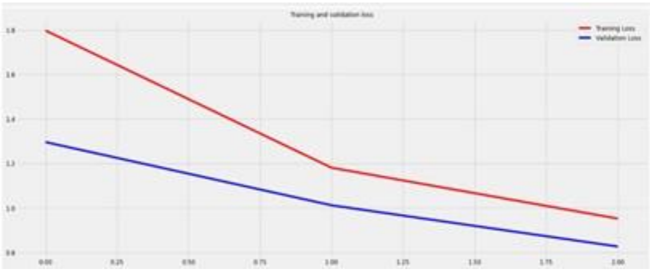

Fig. 6.7: Training and Validation Loss

H. Classification and Prediction:

Finally, the PLSA model is used to classify new documents based on their topic distributions. This can be done using a threshold-based approach or a probabilistic approach such as Bayesian classification. For example, if we have a new document and the PLSA model determines that the document has a high probability of belonging to a certain topic, we can classify the document as belonging to that topic. Three epochs, or the total number of iterations, have been chosen as the starting point. It specifies how many times the learning algorithm must process the complete data collection. Finally, the data in the paper has been effectively categorized utilizing PLSA algorithm analysis, with the phrases having been categorized and described into their appropriate classifications.

The classification report of the proposed system is as follows (Table 1)

Table.1: Classification Report of the Proposed system

	Precision	Recall	F1 score	Support
0	0.77	0.84	0.80	522
1	0.00	0.00	0.00	3
2	0.75	0.57	0.27	81
3	0.76	0.90	0.78	181
4	0.82	0.91	0.86	91
5	0.00	0.00	0.00	23
6	0.00	0.00	0.00	3
7	0.00	0.00	0.00	4
8	0.71	0.79	0.68	92
Accuracy			0.78	1000
macro avg	0.42	0.37	0.36	1000
Weighted avg	0.77	0.78	0.75	1000

Four sample input texts are taken for classification using the PLSA method (fig. 6.9). The pseudocode is as follows:

for word, i **in** tokenizer**.**word_index**.**items():
assign embedding_vector to embedding_index.get(word)
if embedding_vector is not None
then embedding_matrix equates to embedding_vector
hits += 1
else:
misses += 1
define the latent semantic model:
equate the model to sequential
add the embedding to model
add bidirectional LSTM to model
add dense len to the train model
return model

The results are as follows (fig. 6.8):

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 200)          4829800

 bidirectional (Bidirectiona  (None, 100, 512)         935936
 l)

 bidirectional_1 (Bidirectio  (None, 100, 256)         656384
 nal)

 bidirectional_2 (Bidirectio  (None, 256)              394240
 nal)

 dense (Dense)               (None, 9)                 2313

=================================================================
Total params: 6,818,673
Trainable params: 1,988,873
Non-trainable params: 4,829,800
```

Fig. 6.8: Sequential Model

For Classification,
define class sentence in sentences
equate to tokenizer
take pad_sequences
proba = np.max

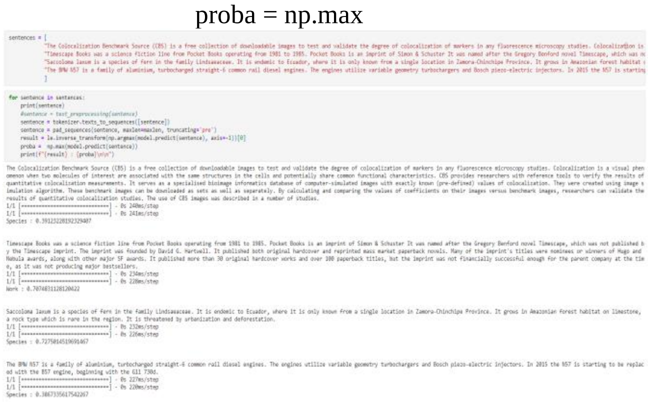

Fig. 6.10: Result of the proposed system

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 7-Issue 1, August 2023
Paper: 50

Table 2 : Performance Metrics Comparison

Method/Model	Precision	Recall	F1 Score	Accuracy
Plsa (Proposed method)	0.77	0.78	0.75	0.78
Decision Tree	0.75	0.73	0.73	0.71
K-Nearest Neighbors	0.78	0.74	0.78	0.72
Logistic Regression	0.78	0.77	0.77	0.74
Random Forest	0.76	0.76	0.76	0.76

Thus after tokenizing the words, with DF-IDF Vectorizer and Transformer, then from the epochs we get an accuracy of about 76% and then from the Matrix evaluation method, we get an accuracy of about 79% by calculating the precision, recall, f1-score and finally getting the accuracy. The training and validation accuracy graph has been portrayed in the project which show a steep increase in the accuracy of the data that has been trained against the pLSA algorithm. Finally the sentences has been described with their respective classes and the words have been classified into their respective classes successfully.

VII. Conclusion and Futute Work

The long feature sparsity and considerable ambiguity in the text make classification jobs extremely difficult. In this paper, we came up with an immediate learning strategy for text classification. During the process of expanding the space for label words, the method can take into consideration both the text itself and the class name. The open Knowledge Graph's top N concepts will be retrieved by the proposed approach, which then refines the expanded label words in embedding space.

In the future, we will expand our research in the two areas listed below. One is looking into better ways to build automatic templates and a verbalizer based on short text. The second option is to incorporate additional supplementary information from outside sources into other tasks. The third option is to increase the efficiency by using the extended model of plsa (i.e., LDA)

VIII. References

[1] P. Wang et al., "Classification of Proactive Personality: Text Mining Based on Weibo Text and Short-Answer Questions Text," in IEEE Access, vol. 8, pp. 97370-97382, 2020, doi: 10.1109/ACCESS.2020.2995905.

[2] P. Wang et al., "Predicting Self-Reported Proactive Personality Classification With Weibo Text and Short Answer Text," in IEEE Access, vol. 9, pp. 77203-77211, 2021, doi: 10.1109/ACCESS.2021.3078052.

[3] Zhang, Lihui & Zhu, Gancheng & Zhang, Shujie & Zhan, Xiangping & Wang, Jun & Meng, Weixuan & Fang, Xin & Wang, Peng. (2019). Assessment of Career Adaptability: Combining Text Mining and Item Response Theory Method. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2938777.

[4] C. Wang, S. Ju, Y. Liu, R. Chen and X. Huang, "Adaptive Feature Extractor of Global Representation and Local Semantics for Text Classification," in IEEE Access, vol. 8, pp. 202687-202695, 2020.

[5] H. M. Abdelaal, B. R. Elemary and H. A. Youness, "Classification of Hadith According to Its Content Based on Supervised Learning Algorithms," in IEEE Access, vol. 7, pp. 152379-152387, 2019,
doi: 10.1109/ACCESS.2019.2948159.

[6] M. F. Mridha, M. A. H. Wadud, M. A. Hamid, M. M. Monowar, M. Abdullah-Al-Wadud and A. Alamri, "L-Boost: Identifying Offensive Texts From Social Media Post in Bengali," in IEEE Access, vol. 9, pp. 164681-164699, 2021, doi: 10.1109/ACCESS.2021.3134154.

[7] H. Liu and Q. Qian, "Bi-Level Attention Model With Topic Information for Classification," in IEEE Access, vol. 9, pp. 125366-125374, 2021,
doi: 10.1109/ACCESS.2021.3058016.

[8] C. Nuo, G. -Q. Chang, H. Gao, G. Pei and Y. Zhang, "WordChange: Adversarial Examples Generation Approach for Chinese Text Classification," in IEEE Access, vol. 8, pp. 79561-79572, 2020, doi: 10.1109/ACCESS.2020.2988786.

[9] H. Ma, Y. Li, X. Ji, J. Han and Z. Li, "MsCoa: Multi-Step Co-Attention Model for Multi-Label Classification," in IEEE Access, vol. 7, pp. 109635-109645, 2019, doi: 10.1109/ACCESS.2019.2933042.

[10] R. Zhao and K. Mao, "Supervised Adaptive-Transfer PLSA for Cross-Domain Text Classification," 2014 IEEE International Conference on Data Mining Workshop, Shenzhen, China, 2014, pp. 259-266, doi: 10.1109/ICDMW.2014.163.