Plagiarism Checker X Originality Report

Similarity Found: 8%

Date: Saturday, April 13, 2024
Statistics: 194 words Plagiarized / 2296 Total words
Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

MACHINE LEARNING WEB APPLICATION WITH FLASK Prabhjot Singh Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University Mohali, India. 20bcs6895@cuchd.in Diya Goel Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University Mohali, India. 20bcs6887@cuchd.in Ujjwal Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University Mohali, India. 20bcs6849@cuchd.in Kumar Saurav Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University Mohali, 20bcs6856@cuchd.in Aadi Pratap Singh Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University Mohali, aadi.e15043@cuchd.in Abstract- With the development in the science field, more and more research and developments have been made in the past few years. Accurate tools are needed to study and analyze long research papers accurately and in less time.

So we have created a web application with Flask that can accurately classify the data written in research papers into different categories for quick analysis used for sequential data classification. The web application is made using Machine learning and applying NLP methods to it with the model trained using sequential neural networking.. The application takes the data from the research paper and classifies the text inside it. The dataset that we are using for this purpose is named as PUBMED 200K RCT which consists of around 200,000 abstracts of randomized controlled trials, totaling 2.3 million sentences. The dataset is quite large compared to other datasets for similar purposes.

The model we have created is accurately trained and its computation time is also less which is a great advantage of this model. The project culminates in the development of a user-friendly Flask web application, enabling users to interact with the trained model.

Deployment is executed on a server or cloud platform for broader accessibility.
Keywords Flask, NLP, machine learning, classification, pre processing. Neural network I.

INTRODUCTION In the realm of human-computer interaction, there exists a compelling need for more natural and intuitive interfaces that can seamlessly bridge the gap between users and technology. In a world where new research and technical advancements are made at each step, it becomes of utmost importance that research articles written on them are accurately and thoroughly analyzed. So the primary objective of the model created by us is to categorize sentences, facilitating a deeper understanding of medical research findings.

The project encompasses comprehensive data pre- processing, model development, evaluation, and deployment stages. Leveraging state-of-the-art Natural Language Processing (NLP) techniques, the project addresses the challenge of analyzing vast amounts of textual data in the biomedical field. more user-friendly interfaces, where users can upload their data and perform sequential task classification. Despite the promise of delivering accuracy, adaptability, and speed, the existing systems have faced many challenges.

Our model focuses on and will deal with these problems by using appropriate training and preprocessing of the dataset. The dataset used is quite new and advanced comparable to other datasets used for the same purpose. The dataset consists of approximately 200,000 abstracts of randomized controlled trials, totaling 2.3 million sentences. Each sentence of each abstract is labeled with their role in the abstract using one of the following classes: background, objective, method, result, or conclusion. The purpose of releasing this dataset is twofold.

First, the majority of datasets for sequential short-text classification (i.e., classification of short texts that appear in sequences) are small: we hope that releasing a new large dataset will help develop more accurate algorithms for this task. Second, from an application perspective, researchers need better tools to efficiently skim through the literature. Automatically classifying each sentence in an abstract would help researchers read abstracts more efficiently, especially in fields where abstracts may be long II.

LITERATURE REVIEW Existed System In the existing landscape of natural language processing (NLP) research, the analysis of medical research abstracts has emerged as a crucial area of focus. Prior to the introduction of the PubMed 200k RCT dataset, researchers faced challenges due to the lack of large-scale datasets tailored specifically for sequential sentence classification in medical texts. While some smaller datasets existed, they were either not publicly available, focused on non-RCT abstracts, or limited in size, hindering the development of accurate algorithms for sequential sentence classification tasks.

One of the most notable challenges in the existing system was the absence of structured abstracts in a significant portion of published RCTs. Unstructured abstracts made it difficult for researchers to quickly locate relevant information, slowing down tasks such as literature review and evidence-based medicine. Proposed System The introduction of the PubMed 200k RCT dataset presents a novel solution to the limitations of the existing system. This dataset, consisting of approximately 200,000 abstracts of randomized controlled trials (RCTs) and totaling 2.3

million sentences, addresses the need for large-scale datasets in sequential sentence classification tasks within the medical domain. The proposed method involves the meticulous labeling of each sentence in the dataset with one of five classes: background, objective, method, result, or conclusion. This labeling scheme provides researchers with a structured framework for analyzing and categorizing information within medical abstracts.

Furthermore, the dataset is split into training, validation, and test sets, enabling researchers to develop and evaluate algorithms for sequential sentence classification effectively. By providing a comprehensive resource with a sizable volume of meticulously labeled data, the proposed system aims to facilitate the development of accurate algorithms for tasks such as evidence-based medicine, literature review, and information retrieval in the medical domain. III. METHODOLOGY The created model has accuracy above 75% and we are working on improving it. Here is the information about the dataset we are using i.e.

"PubMed 20k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts",where Size of training set : 180040 #distribution of labels in training data METHODS: 59353 RESULTS: 57953 CONCLUSION: 27168 BACKGROUND: 21727 OBJECTIVE: 13839 Size of validating set: 30212 Size of testing set: 30315 Here are the following steps used while creating our Machine learning web application: 1. Install and Import Dependencies: TensorFlow, Sklean, and Pandas are just a few of the libraries and frameworks that must be installed and imported initially.

Flask is utilized for making the machine learning model to display on the web, while TensorFlow is implemented for model construction and training. 2. Data Collection and Pre-processing: Data is imported using Pandas and several pre-processing steps have been taken. Some irrelevant columns were removed. Preprocessing with SpaCy and Lametization: • SpaCy is used to tokenize the text and perform initial preprocessing tasks. • Lemmatize the tokens to reduce them to their base forms and normalize the text.

• Remove stop words to eliminate common, non- informative words. • Joined the preprocessed tokens back into a single string, which will serve as the input for further processing. 3. TF-IDF Vectorization: Utilized TF-IDF vectorization to convert the text data into numerical features which assigns weights to terms based on their frequency in a document relative to their frequency across all documents, capturing their importance in distinguishing between documents. 4. Training and Testing the data:The data is split into training and testing sets, where 10% of the data i.e. 20,000 is used for testing and trained using train_test_split. 5.

Apply Multinomial Naive Bayes classifier: To efficiently classify text documents into multiple categories based on the frequency of occurrence of words, while making the simplifying assumption of feature independence. Naive Bayes classifiers are efficient and effective for text classification, especially when dealing with high-dimensional data like TF-IDF vectors. 6. Apply Logistic regression: To predict the probability of each class label for a given text document, enabling efficient classification into one of the five categories. 7.

Apply SVM classification model: Used to classify text documents by finding the optimal hyperplane that separates different classes in a high-dimensional space, maximizing the margin between them. 8. Cross Validation: Split the TF-IDF matrix into training and validation sets for performance analysis. Performed k-fold cross-validation to train and evaluate the model multiple times on different subsets of the data. 9. Define a revised neural network model: We have implemented this model through a neural network consisting of multiple dense layers with ReLU activation, batch normalization layers for stabilization, dropout layers for regularization, and a softmax activation function in the output layer for multi-class classification. 10. Adam optimization and early stopping: Used to efficiently minimize the loss function during training by adapting the learning rate for each parameter individually.

Performed early stopping to prevent overfitting. 11. Label Encoding: Used to encode categorical target labels into numerical values, allowing the model to interpret and process the target variable. It transforms class labels into integers. 12. Sort the indices: Sort the indices of the sparse matrices and then of the training and validation sets. 13. Performance Analysis: Calculate classification metrics such as accuracy, precision, recall, F1-score, and confusion matrix to analyze the model's performance on the test data and adjust hyperparameters and experiment with different configurations to optimize the model's performance. 14. Deployment: The model is deployed on the web using Flask by creating the pickle file of the model. Web application to be hosted on a cloud platform for easy access. Fig 1.1: Flowchart of Model IV.

RESULT Our journey to improve our classification model is one of hard work and innovation. On this journey, we realized the importance of using a wide range of classification methods to get the most out of our model. With careful experimentation, we selected an ensemble of algorithms such as multinomial naive, logistic regression and cutting edge deep learning neural networks, each with its own strengths. Together, these methods yielded an accuracy score of 74% as a stepping stone for continuous improvement. Driven by a spirit of constant iteration, we set out to continually improve our model.

performance, tuning parameters and refining architectures to get the best out of the model. At the heart of our model enhancement strategy is augmenting our dataset and optimizing our training protocols. Understanding the importance of data in model t hesitate to expand our corpus. By combining a wide range of research papers from different disciplines, we increased our dataset and enabled our model to learn from a wider range of textual nuances. We also increased the training time by increasing the number of iterations of our neural network. Not only did this increase the model's resilience to overfitting, but it also gave it a better understanding of the structure of the research paper text.

At the end of the day, our machine-learning model is a testament to the power of innovation and persistence. Inside an easy-to-use interface, we have a powerful tool that can analyze research papers with remarkable accuracy. By classifying texts into five different domains — Methods, results, conclusion, background, and objective — our model makes it easy for users to navigate the maze of academic literature. As the future beckons, we continue to push the limits of text classification. Not only is our model a sign of technological progress, but it is also a sign of hope for a future where the pursuit of knowledge knows no bounds. V.

CONCLUSION We implemented a Natural Language Processing (NLP) model capable of segmenting text lines in abstracts from medical research papers. The model was capable of learning the task of the said segmentation with minimal dependencies as few features were used to train the model. The model was also capable of generalizing to unseen data according to the performance metrics used, notably achieving a decent F1-score & Mathews correlation coefficient on the training, validation, and testing data.

With ongoing advancements, text classification models hold the promise of revolutionizing various industries and improving the quality of life for people around the

world. REFERENCES [1] Han Lin, H. Y., & Murli, N. (2022). BIM Sign Language Translator Using Machine Learning (TensorFlow). Journal of Soft Computing and Data Mining, 3(1), 68 77. Retrieved from https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/1166 [2] Aggarwal, A., Bhutani, N., Kapur, R. et al. Real-time hand gesture recognition using multiple deep learning architectures. SIViP 17, 3963 3971 (2023). https://doi.org/10.1007/s11760-023-02626-8 [6] Suharjito, S., Anderson, R., Wiryana, F., Ariesta, M.C., Kusuma, G.P.:

Sign language recognition application systems for deaf-mute people: a review based on input-process-output. Procedia Comput. Sci. 116, 441 448 (2017). https://doi.org/10.1016/J.PROCS.2017.10.028 [7] Konstantinidis, D., Dimitropoulos, K., Daras, P.: Sign language recognition based on hand and body skeletal data. In: 3DTV-Conference, pp. 1 4 (2018). https://doi.org/10.1109/3DTV.2018.8478467 [8] Dutta, K.K., Bellary, S.A.S.: Machine Learning techniques for Indian sign language recognition. In: International Conference Current Trends Computing Electric Electronics Communication CTCEEC 2017, pp. 333 336 (2018). https://doi.org/10.1109/CTCEEC.2017.8454988 [9] Bragg, D., et al.:

Sign language recognition, generation, and translation: an interdisciplinary perspective. In: 21st International ACM SIGACCESS Conference on Computer Accessibility,12, pp.16 31 (2019). https://doi.org/10.1145/3308561 [10] Rosero-Montalvo, P.D., et al.: Sign language recognition based on intelligent glove using machine learning techniques. In: 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM), pp. 1 5 (2018). https://doi.org/10.1109/ETCM.2018.8580268 [11] Enikeev, D.G., Mustafina, S.A.: Sign language recognition through leap motion controller and input prediction algorithm. J. Phys. Conf. Ser. 1715, 012008 (2021). https://doi.org/10.1088/1742-6596/1715/1/012008 [12] Cui, R., Liu, H., Zhang, C.:

A deep neural framework for continuous sign language recognition by iterative training. IEEE Trans. Multimed. 21, 1880 – 1891 (2019). https://doi.org/10.1109/TMM.2018.2889563 [13] Bantupalli, K., Xie, Y.: American sign language recognition using deep learning and 13 computer vision. In: Proceedings of - 2018 IEEE International Conference on Big Data, Big Data 2018, pp. 4896 – 4899 (2019). https://doi.org/10.1109/BIGDATA.2018.8622141 [14] Sharma, A., Sharma, N., Saxena, Y., Singh, A., Sadhya, D.: Benchmarking deep neural network approaches for Indian sign language recognition. Neural Comput. Appl. 33(12), 6685 – 6696 (2020). https://doi.org/10.1007/s00521-020-05448-8 [15] Pu, J., Zhou, W., Li, H.:

Iterative alignment network for continuous sign language recognition. In: IEEE Computer

Social Conference Computing Vision Pattern Recognition, pp. 4160 – 4169 (2019). https://doi.org/10.1109/CVPR.2019.00429 [16] Liang, Z., Liao, S., Hu, B.: 3D Convolutional neural networks for dynamic sign language recognition. Comput. J. 61, 1724 1736 (2018). https://doi.org/10.1093/COMJNL/BXY049

INTERNET SOURCES:

<1% - https://www.careers360.com/colleges/apex-institute-of-technology-chandigarh
<1% - https://www.researchgate.net/profile/Aditya-Singh-222
<1% - https://ar5iv.labs.arxiv.org/html/1710.06071
<1% - https://link.springer.com/article/10.1007/s42486-019-00007-0
<1% - https://arxiv.org/abs/1910.03771v1
3% - https://www.researchgate.net/publication/320464805_PubMed_200k_RCT_a_Dataset_for_Sequential_Sentence_Classification_in_Medical_Abstracts
<1% - https://www.sciencedirect.com/science/article/pii/S2949719122000012
<1% - https://paperswithcode.com/dataset/pubmed-rct
<1% - https://github.com/JamesMTucker/pubmed-rct-expanded
<1% - https://www.v7labs.com/blog/train-validation-test-set
<1% - https://www.ritchieng.com/machine-learning-multinomial-naive-bayes-vectorization/
<1% - https://towardsdatascience.com/text-classification-using-naive-bayes-theory-a-working-example-2ef4b7eb7d5a
<1% - https://medium.com/@ethanwchen/understanding-support-vector-machines-svm-c929d79d3b0a
<1% - https://stats.stackexchange.com/questions/577327/kfold-cross-validation-with-train-test-validation-set
<1% - https://pg-p.ctme.caltech.edu/blog/ai-ml/what-is-machine-learning
<1% - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8223521/
<1% - https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/download/11662/4993
<1% - https://dl.acm.org/doi/10.1145/3394171.3413528
<1% - https://www.researchgate.net/publication/348273892_Sign_language_recognition_through_Leap_Motion_controller_and_input_prediction_algorithm
<1% - https://www.semanticscholar.org/paper/A-Deep-Neural-Framework-for-Continuous-Sig

n-by-Prakash-Moni/2a7817bbaf95e6256a7fc1f6757354292767995a