

Project Report - IT Support Email Classification

Overview

The purpose of the project is to build a machine learning model to classify IT support emails into different categories based on the content of the email. Once ML model is trained, the task would be to predict the label for each corresponding email. The goal is to automate email classification to streamline the support process.

Data Collection and Data Preprocessing

The IT support email dataset used for this project consists of 5,000 support emails. Each data is labelled with a category. The data was loaded and pre-processing was performed:

- a) **Text Preprocessing:** The text cleaning was performed using Spacy. The raw text was first cleaned by converting it to lowercase, removing stop words, punctuation, and digits. Also, lemmatization was performed to convert the words to their base form. Lastly, the tokens were extracted from the text.
- b) **Label Encoding:** The label column named 'Topic_group' was encoded into numerical labels using LabelEncoder. After that each distinct label in the dataset was converted into a unique integer.
- c) **Data Splitting:** The dataset was splitted into training, validation, and test sets. 80% of the data was used for training and validation while 20% was used for testing purpose.
- d) **Creation of Vocabulary:** A vocabulary was created from the training data. Each word in the vocabulary was assigned a unique index.

Model Architecture (V1)

The model is based on a convolutional neural network (CNN) architecture.

- a) **Embedding Layer:** An embedding layer was used to convert the words into dense vectors of an embedding dimension i.e. 100.
- b) **Convolutional Layers:** Used 3 convolutional layers with different kernel sizes i.e. 3, 5, and 7 were used to capture different n-gram features in the email text. Each layer used 128 filters, and ReLU as activation function.
- c) **Fully Connected Layer:** Once features were extract from the convolutional layers, the results were concatenated and passed through a fully connected layer to produce the output scores.

- d) **Dropout:** Dropout was used with a probability of 50% was applied to prevent overfitting of training data.

Model Architecture (V2)

To improve performance and regularization, several modifications were made to the initial model:

- a) **Convolutional Layers:** The number of filters was reduced to 64 in each convolutional layer.
- b) **Batch Normalization:** Batch normalization was added after each convolutional layer to stabilize the learning process.
- c) **Dropout:** Dropout was increased to 0.8 to enhance regularization.
- d) **Learning Rate and Regularization:** The learning rate was increased to 0.002, and L2 regularization (weight decay) was applied with a value of $1e-6$.

Model Training

The models were trained using the following setup:

- a) **Loss Function:** Cross-entropy loss was used to measure the discrepancy between predicted and true labels.
- b) **Optimizer:** The Adam optimizer was used with a learning rate of 0.001 for the initial model and 0.002 for the version 2.
- c) **Training Duration:** The model was trained for 10 epochs, with the loss monitored on both the training and validation sets.

Model Evaluation

After training, the models were evaluated on the test set. Below are the performance metrics for both versions of the model.

Model V1

- a) **Accuracy:** 80%
- b) **Precision, Recall, and F1-score:** The model performed well on categories like "Purchase" and "Access" with high precision and recall. It performed less well on categories like "Administrative rights" and "Hardware" which may require further tuning or additional data to improve performance.

Model V2

- a) **Accuracy:** 81%
- b) **Precision, Recall, and F1-score:** The V2 model showed similar overall accuracy, but the precision and recall values for certain categories improved, such as "HR Support" and "Purchase." The model's regularization improvements helped maintain its performance across various classes, particularly the smaller ones like "Administrative rights."

Key Insights

- **Model Performance:** Both versions of the model achieved an accuracy of approximately 81%. While the second version showed slightly improved performance due to regularization and tuning, the accuracy remained relatively consistent.
- **Category Performance:** Categories like "Access," "Purchase," and "HR Support" performed well, but more complex or smaller categories, such as "Administrative rights" showed room for improvement.
- **Overfitting and Regularization:** The second version with higher dropout and regularization performed more stably on the validation set, preventing overfitting.

Conclusion

This project successfully demonstrated how CNNs, can be applied to text classification tasks in the IT support domain. The model can classify emails into various categories, streamlining the support process. Despite some areas of improvement, especially for smaller classes, the project has solid foundation for building a robust email classification system for handling IT support tickets.