Identification of Quote Themes: A Multi-label Classification Problem

Abstract

Categorizing quotes is a multi-label text classification problem, that presents unique difficulties due to the brevity, ambiguity, metaphors, and nuanced language of quotes. Models tackling this problem must exhibit a deep understanding of language, context, and cultural references. This study aims to develop and evaluate a multi-label quote classification model for categorizing English quotes into their topics/themes. To identify and extract sophisticated features from quote texts, techniques such as TF-IDF, and SentenceTransformer text-embeddings, were used. Employing traditional machine learning models, specifically K-Nearest Neighbour, Random Forest, and Multi-Layer Perceptron, this study utilized two multi-label constructor strategies—Binary Relevance and Multi-Output Classifier. Through extensive evaluation using both sample-based and label-based metrics, Random Forest emerged as the most effective multi-theme quote classifier. It demonstrated higher precision, particularly with feature vectors derived from SentenceTransformers and the combination of TF-IDF and SentenceTransformers. However, it is noteworthy that the classification performance was notably hindered by the higher dimensions in the label space of the dataset. Despite this, the developed quote classification model holds promise for facilitating data-driven decision-making across diverse domains.

1. Introduction and Mathematical Formulation

Multi-label text classification is a critical task in Natural Language Processing as the text passages are not confined to a single category, but may belong to multiple, distinct labels or themes simultaneously [1]. On the other hand, quote classification poses challenges compared to other multi-label text classification problems, as quotes are brief, often lacking the contextual information essential for a full understanding. Ambiguity is another hurdle, as quotes can be open to multiple interpretations, adding complexity to the categorization task. Moreover, the source of a quote and its author's style can also heavily influence the categorization.

Training models to classify quotes into topics/themes has both practical and research-oriented advantages. In addition to enhancing user experiences, and content recommendations on social platforms, this supports academic research and decision-making in various domains, such as psychology, sociology, and market research. By categorizing quotes, researchers can track and analyze emerging trends, cultural shifts, and evolving public opinions over time. While there are few studies in the literature, the authors of [2] have developed a Naive Bayes model for classifying Indonesian Twitter quotes into 6 categories; Love, Life, Motivation, Education, Religion, and Others.

Inspired by this context, this study aims to develop a multi-label quote classification model for identifying the related topics/themes of a given quote. The dataset[1] that we are using for this study is made up of 3000 English quotes that have been sourced from Goodreads[2]. This has three data fields: Quote, Author, and Tags, however, we will only use the quotes and tags for our problem. An example record of the dataset is shown below.

- Quote: Live as if you were to die tomorrow. Learn as if you were to live forever.
- Labels: ['carpe-diem', 'education', 'inspirational', 'learning']

The evaluation metrics of multi-label classification are mainly categorized into two as, sample-based and label-based [3][4][5]. Here we will use sample-based metrics, such as precision, recall, f1-score, and Hamming loss. Equation (1) shows the Hamming-loss, which refers to an average binary classification error [4]. Assume that the multi-label evaluation dataset D contains multi-label examples $(x_i, Y_i,), i=1, 2... N, Y_i \subseteq L$ is a set of true labels, $L= \{l_j: j=1...M\}$ is the set of all labels, and x_i is a new instance. Hence, multi-label Hamming-loss can be calculated as:

$$Hamming\text{-}loss = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \Delta Z_i|}{M} \quad (1)$$

where M refers to the maximum number of labels, N refers to the maximum number of examples, and $Zi = h(x_i)$ is a set of labels predicted by a multi-label classifier h for example x_i.

For label-based evaluations, the macro/micro precision, recall, and f1-score will be used. Equation (2) and (3) shows the calculation of the f1-score and its micro-average respectively [4].

$$F1\text{-}measure = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i|+|Y_i|} \quad (2)$$

$$M_{micro} = M\left(\sum_{l=1}^{M} tp_l, \sum_{l=1}^{M} fp_l, \sum_{l=1}^{M} tn_l, \sum_{l=1}^{M} fn_l\right) \quad (3)$$

where tp_l, fp_l, tn_l, and fn_l denote the number of true positives, false positives, true negatives and false negatives for l labels.

2. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is imperative for comprehending the dataset's characteristics before constructing a multi-label text classification model. In this study, various EDA techniques were employed for the dataset with two columns: text quotes and multiple labels. Initially, basic statistics were computed and are presented in Table 1. These statistics were derived from the final dataset used for model development, following several pre-processing tasks. Notably, the

[1] https://www.kaggle.com/datasets/abireltaief/english-quotes
[2] https://www.goodreads.com/quotes

dataset exhibits a relatively sparse nature, with a higher ratio of the total number of labels to the total number of records. Additionally, it is interesting to observe that the average number of labels per quote hovers around 2, while the maximum number of labels per quote is 36.

Table 1: Basic statistics of the complete dataset

Number of records in the dataset	2231
Number of unique labels in the dataset	1592
Maximum number of labels per quote	36
Minimum number of labels per quote	1
Average number of labels per quote	2.36
Average length of quotes	14.34 words

For the text quotes, as illustrated in Fig. 1, a word cloud was generated to visualize the most frequent words associated with the overall dataset. This visualization provided insights into the prominent themes. The most commonly occurring words in the dataset included: love, life, time, book, people, and world.

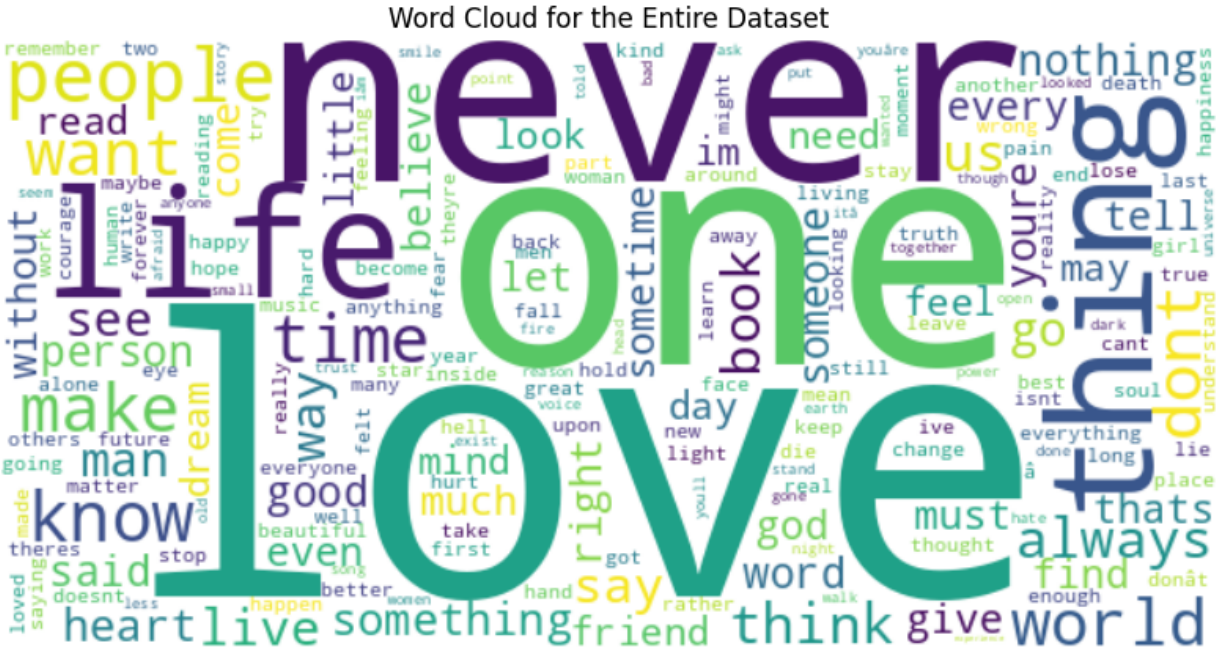

Figure 1: The word cloud for the complete dataset

Since the number of distinct labels in the dataset is relatively higher, this EDA only focused on the top N most frequent labels. Fig 2. visualizes the frequency of the top 20 labels. Notably, the most common word in the dataset is "love," reflecting the idea of the prominent label in the dataset. Also, the labels connected to humor, inspiration, life, and books have a much larger number of quotes.

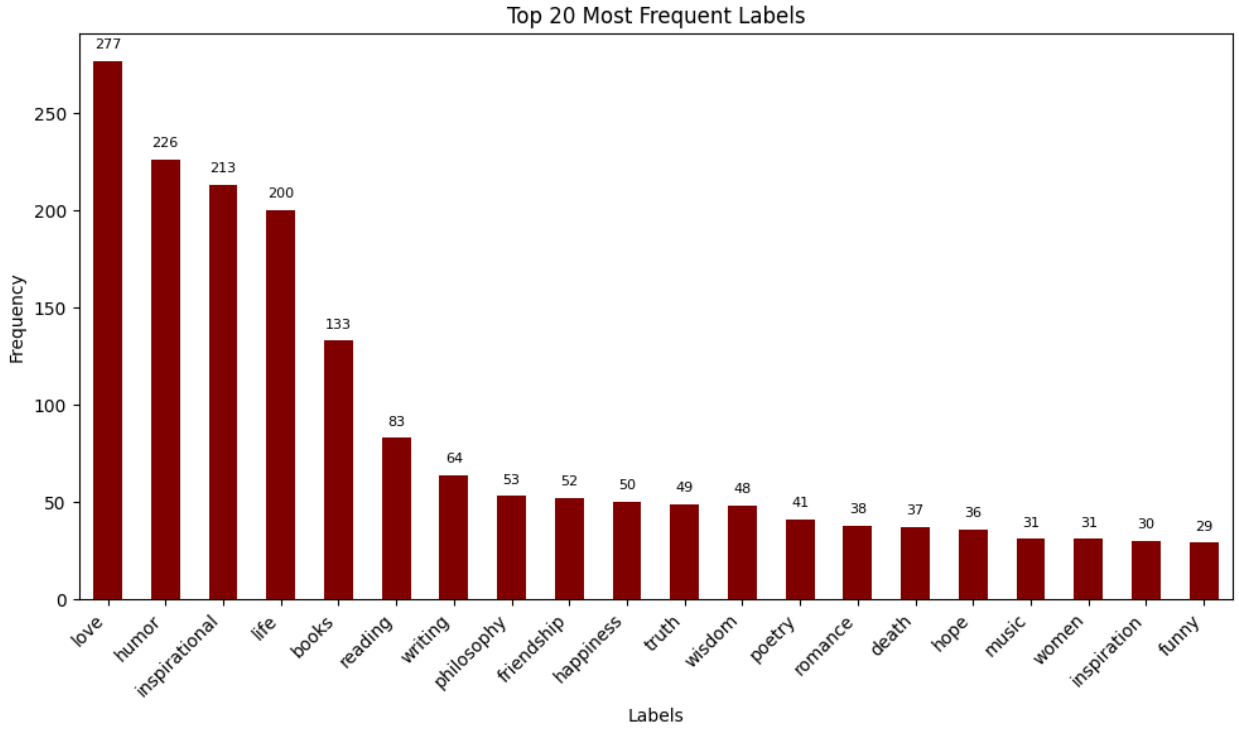

Figure 2: Frequency of the top 20 labels in the dataset.

Subsequently, the co-occurrence of the top 30 labels was analyzed by generating a heatmap to visualize which labels tend to appear together (refer to Fig. 3). This analysis sheds light on the relationships between different popular labels, explaining how often label 'i' and label 'j' appear together. Darker cells in the heatmap indicate a higher frequency of co-occurrence between the corresponding labels. Noteworthy correlations in the label space include reading-books, love-inspirational, love-life, life-inspirational, life-humor, and love-romance.

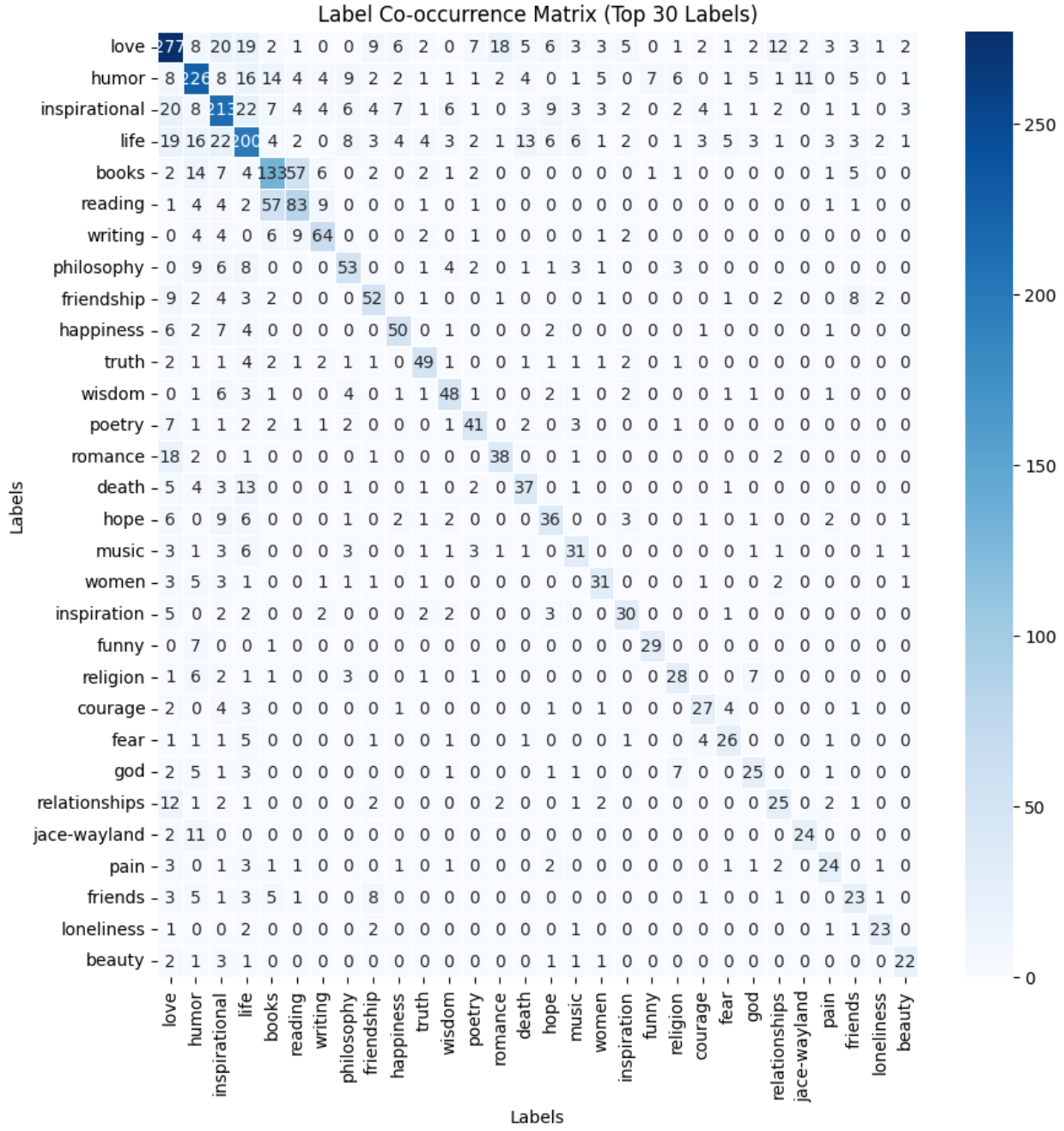

Figure 3: The relationships between the most popular 30 labels.

Finally, the EDA explored the distribution of text lengths for quotes with the top 5 labels using a boxplot. As depicted in Fig. 4, the top label (love) exhibits a higher distribution of text lengths, while label 4 (life) shows the minimum distribution.

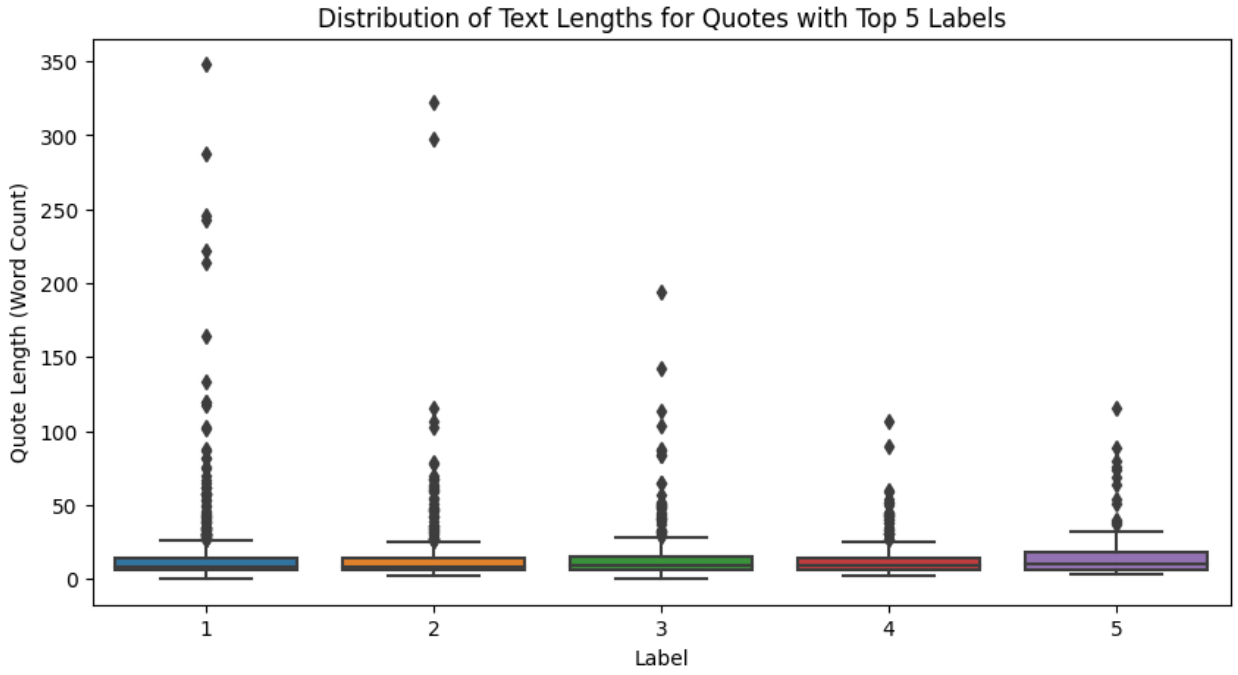

Figure 4: Distribution of text length in the top 5 labels.

3. Model Development

In this section, the techniques used to clean the dataset, extract the features, and implement classifiers will be discussed in detail.

3.1 Data Pre-processing

This study implemented several data-cleaning steps before proceeding with feature extraction. Initial tasks included the removal of rows containing non-alphanumeric characters, the conversion of text to lowercase, elimination of extra spaces, punctuation, and stopwords. Subsequently, the dataset was prepared for multi-label classification through one-hot encoding, converting labels into separate columns. Labels represented solely by numbers and special characters were discarded as they lacked meaningful themes. Following this, rows without at least one label were identified and removed from the dataset, resulting in a dataset size of (2494, 2152).

Given the relatively high number of total labels, quotes with an excess of five labels were further eliminated to reduce dataset size. This was crucial to prevent RAM space issues during y-space feature engineering. After the removal of rows with no labels, the final dataset size was (2231, 1592). Lastly, the dataset was split into training and testing sets using a 7:3 ratio, resulting in 1561 samples for training and 670 samples for testing.

3.2 Feature Extraction and Engineering

Feature engineering is a critical step in building effective models for text classification. This process involves the identification and extraction of the most relevant aspects of textual data that can be used to distinguish and categorize text into predefined classes. This study focused on identifying features, including words, phrases, or even more complex linguistic patterns, which have been extracted through techniques, such as TF-IDF (Term Frequency-Inverse Document Frequency), and text embedding methods, including Word2Vec[6], and SentenceTransformers[3].

After obtaining these feature vectors, the Principal Component Analysis (PCA) technique was applied to reduce the higher dimensional space. Notably, PCA was exclusively performed on the TF-IDF vectors, given that text-embedding vectors are typically dense. Additionally, the TF-IDF vectors underwent scaling using the MinMaxScaler function, while the embedding vectors were exempted as they are generally normalized. This study utilizes four types of feature vectors, including the combination of TF-IDF and SentenceTransformers vectors as a feature to train and test the classifiers. Importantly, the feature extraction and engineering techniques were separately applied to the training and testing datasets.

3.3 Implementing Classifiers

This study developed two traditional machine learning models, namely K-Nearest Neighbour (KNN), Random Forest (RF), and Multi-Layer Perceptron (MLP) as classifiers using two multi-label classification strategies: MultiOutputClassifier (MOC) and BinaryRelevance (BR) [7]. The MOC involves training a separate classifier for each target, while the BR constructor converts a multi-label classification task with L labels into L individual single-label binary classification tasks, utilizing the base classifier specified in the constructor. These strategies serve to adapt classifiers that do not inherently support multi-label classification.

Therefore, considering three classifiers, two multi-label constructors, and four types of feature vectors, a combination of 24 result sets was expected. However, due to limitations in RAM and GPU on the execution environment (Google Colab with V100 GPU), only 17 result sets were obtained. The performance measures, including accuracy, Hamming loss, micro Precision, micro Recall, and micro F1-Score, were generated.

4. Results and Discussion

The summary of the classification results are shown in Table 2.

[3] https://www.sbert.net/examples/applications/computing-embeddings/README.html

Table 2: Multi-label classification results

Model	Feature vector	Multi-label constructor	Classifier	Accuracy	Hamming loss	Precision	Recall	F1-score
M1	TF-IDF	MOC	KNN	0.0000	0.0026	0.5039	0.0759	0.1320
M2			RF	0.0000	0.0025	0.7915	0.0947	0.1692
M3			MLP	0.0000	0.0027	0.3622	0.0667	0.1126
M4		BR	KNN	0.0000	0.0026	0.5039	0.0759	0.1320
M5			RF	0.0000	0.0026	0.8286	0.0832	0.1513
M6			MLP	0.0000	0.0027	0.3856	0.0694	0.1176
M7	SentenceTransformers	MOC	KNN	0.0045	0.0026	0.5489	0.1660	0.2549
M8			RF	0.0000	0.0025	0.9795	0.1075	0.1938
M9			MLP	0.0030	0.0026	0.5492	0.1488	0.2341
M10		BR	KNN	0.0045	0.0026	0.5489	0.1660	0.2549
M11			RF	0.0030	0.0025	0.9841	0.1048	0.1895
M12			MLP	0.0030	0.0025	0.5646	0.1530	0.2408
M13	Combined	MOC	KNN	0.0000	0.0026	0.5368	0.0888	0.1524
M14			RF	0.0000	0.0025	0.9941	0.0952	0.1738
M15			MLP	0.0045	0.0026	0.5547	0.0917	0.1574
M16		BR	KNN	0.0000	0.0026	0.5368	0.0888	0.1524
M17			RF	0.0000	0.0025	0.9947	0.1044	0.1889
M18			MLP					

In comparing the performances of various feature vectors, it is evident that the overall performance is consistently higher across all metrics when utilizing SentenceTransformers for feature generation. In contrast, TF-IDF yields the least favorable outcomes. An interesting observation is that, when combined with TF-IDF, the classification performance using SentenceTransformer features decreases in terms of recall and F1-score, while it increases in terms of precision. Moreover, RF demonstrates notable performance superiority over both KNN and MLP across all metrics, while KNN outperforms MLP. Remarkably, there is no significant difference observed in the results of the two multi-label classification constructor strategies.

Reviewing the overall performance of multi-label classification models reveals an interesting pattern. Although the model demonstrates relatively higher precision, indicating accurate predictions of positive labels, the overall results, including accuracy and recall, are lower. This suggests that while the model excels at accurately predicting positives, it struggles to identify all relevant positive instances. A possible reason for this challenge is the higher label space, which could lead to increased model complexity, potentially causing overfitting on the training data and

difficulties in generalization to unseen examples [8]. Moreover, this higher feature space posed challenges related to computational efficiency and increased resource requirements during training and inference.

Considering the reduction of the label space, this study implemented a label space reduction technique: Binary Relevance with Feature Selection using the Chi-squared test approach. However, due to the higher y-space dimensional space and limited RAM and GPU resources available, further experiments were unable to be performed.

To further enhance the study, experiments on class distributions could be beneficial. Moreover, exploring label space feature selection techniques could be beneficial to identify and retain the most relevant features, reducing dimensionality and potentially improving the model's ability to generalize. For this, dimensionality reduction in multi-label feature space techniques, such as Principle Label Space Transformation (PLST) [9], could be conducted. Additionally, employing sentiment/emotion scores using lexicon tools, such as VADER and EmoTFIDF, could be considered as features, especially given that these quotes are rich in sentiments and emotions."

Conclusion

In this study, we applied various techniques for feature extraction and classification to conduct multi-label classification, aiming to identify pertinent themes or tags within human-generated quotes. The outcomes revealed that employing Random Forest with Binary Relevance and Multi-Output Classifier strategies for SentenceTransformer and TF-IDF combined feature vectors yielded the highest precision, specifically 99.47% and 99.41% in label prediction. However, the challenges associated with a higher number of labels became apparent. This complexity heightened the demands on both the classification task and the computational resources needed for training and inference. As the number of labels grows, the dataset has become sparse, leading to imbalances in label occurrences and potentially hindering the model's ability to generalize effectively. Moreover, a larger label set often implies a more diverse range of concepts, making it harder to interpret and understand the relationships between labels. Consequently, the accuracy, recall, and f1-scores exhibited lower values. In light of these findings, it is essential to recognize that while multi-label classification enables a nuanced data representation, thoughtful consideration of the trade-offs related to managing a larger number of labels is crucial.

References

[1] Liu, W., Pang, J., Li, N., Zhou, X., & Yue, F. (2021). Research on multi-label text classification method based on tALBERT-CNN. *International Journal of Computational Intelligence Systems*, *14*(1), 201.

[2] Rachmadany, A., Pranoto, Y. M., Multazam, M. T., Nandiyanto, A. B. D., Abdullah, A. G., & Widiaty, I. (2018). Classification of Indonesian quote on twitter using naïve bayes. In *IOP Conference Series: Materials Science and Engineering* (Vol. 288, No. 1, p. 012162). IOP Publishing.

[3] Zhang, M. L., & Zhou, Z. H. (2013). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, *26*(8), 1819-1837.

[4] Nasierding, G., & Kouzani, A. Z. (2012, May). Comparative evaluation of multi-label classification methods. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery* (pp. 679-683). IEEE.

[5] Wu, X. Z., & Zhou, Z. H. (2017, July). A unified view of multi-label performance measures. In *international conference on machine learning* (pp. 3780-3788). PMLR.

[6] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[7] Luaces, O., Díez, J., Barranquero, J., del Coz, J. J., & Bahamonde, A. (2012). Binary relevance efficacy for multilabel classification. Progress in Artificial Intelligence, 1, 303-313.

[8] Siblini, W., Kuntz, P., & Meyer, F. (2019). A review on dimensionality reduction for multi-label classification. IEEE Transactions on Knowledge and Data Engineering, 33(3), 839-857.

[9] Tai, F., & Lin, H. T. (2012). Multilabel classification with principal label space transformation. Neural Computation, 24(9), 2508-2542.

Group Members (Group No 20)

Chandreen Ravihari Liyanage - 1158931
Sara Nazar - 1223124