Natural Language Processing - Project

António Jotta - 99893, António Morais - 102643, Rúben Nobre - 99321
Group: 41
Email: antonio.jotta@tecnico.ulisboa.pt, antonio.v.morais.c@tecnico.ulisboa.pt,
rubennobre2001@tecnico.ulisboa.pt

I. Models

A. Pre-processing

The objective of this project is to develop a model capable of predicting the genre of a movie using information from a provided dataset[1]. After careful analysis, the 'Director' and 'Plot' were identified as the most relevant features for genre classification and were subsequently utilized to train models under evaluation. The dataset underwent pre-processing as follows: **1) General Pre-processing:** Removal of non-alphanumeric characters, conversion to lowercase, and stop word removal using the NLTK corpus[2]; **2) Vectorizer and Encoder:** The `TfidfVectorizer` class was used to vectorize the 'Plot' data, and `OneHotEncoder` was employed to one-hot encode the 'Director' data (for models I-B, I-C); **3) Tokenization:** The 'Plot' data was tokenized using the `DistilBertTokenizer`, with special tokens such as `[CLS]` for classification and `[SEP]` for separation. The 'Director' data was encoded into the 'Plot' data with simple string interpolation (for model I-D).

B. K-Nearest Neighbours (KNN)

The KNN model [1] was used as a weak baseline in this project, implemented with the `KNeighborsClassifier` class. It classifies a new point based on the majority class of its K nearest neighbors in the feature space, using the Euclidean distance. KNN then calculates distances to all training points, selects the K closest, and assigns the majority class to the new point.

C. Support Vector Machine (SVM)

A SVM model [2] was used as a stronger baseline classifier, using the `SVC` class. It classifies data by finding a hyperplane that best separates different classes, maximizing the margin between the nearest points of each class, called support vectors. For multi-class classification the decision rule is given by

$$f(x) = \text{sign}(\mathbf{w}_k^T \mathbf{x} + b_k),$$

where $\mathbf{w}_k$ is the weight vector for class k and b_k is the bias term. The classifier assigns the label based on which class hyperplane the data point lies on. Tuning parameters such as the kernel and the regularization parameter C will be discussed in II.

D. Bidirecional Encoder Representations from Transformers (BERT)

The BERT [3] model architecture used a single input stream, where the 'Plot' and 'Director' data were combined into a unified text representation. The 'Director' was integrated directly into the 'Plot' text before tokenization. This combined input was then processed by a pre-trained BERT encoder, which generated a pooled representation of the entire text. A dropout layer was applied to the pooled output to reduce the risk of overfitting. Finally, a dense $softmax$ classification layer predicted the probability distribution across all genres, using the output from BERT to make the final genre classification. The model's pipeline is shown in Figure 1.

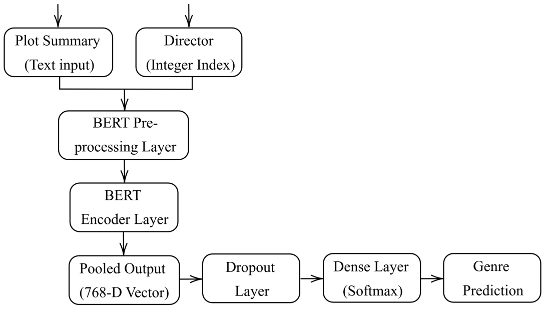


Fig. 1. Pipeline of the developed BERT model

II. Experimental Setup and Results

The labeled dataset was divided into three subsets: training (80%), validation (10%), and testing (10%). For models I-B and I-C, the test subset was used as a validation set, resulting in the use of only the training and test subsets. In contrast, model I-D employed all three subsets, with the test subset reserved for comparison with the other models. The results presented in Tables I-II correspond to the test subset, and the evaluation measures used are Accuracy, F1-Score and Recall.

For the KNN model, the tuning parameter is k. To optimize this parameter, a loop was executed for $k = 1, \ldots, 100$, selecting the optimal value of k based on the highest accuracy achieved. The optimal value identified was $k = 81$, although the metric values were low, as anticipated.

Regarding the SVM model, `GridSearchCV` was used to tune the kernel ('linear', 'poly', 'rbf', 'sigmoid'), γ ('auto', 'scale'), C (logarithmically from 0.001 to 1000), and the degree (for the 'poly' kernel) parameters of `SVC`. After tuning based on accuracy, the best parameters identified were:

[1] https://www.kaggle.com/datasets/jrobischon/wikipedia-movie-plots
[2] https://www.nltk.org/api/nltk.corpus.html

{'kernel': 'sigmoid', C: 10, γ: 'scale'}, achieving better results than KNN, as expected.

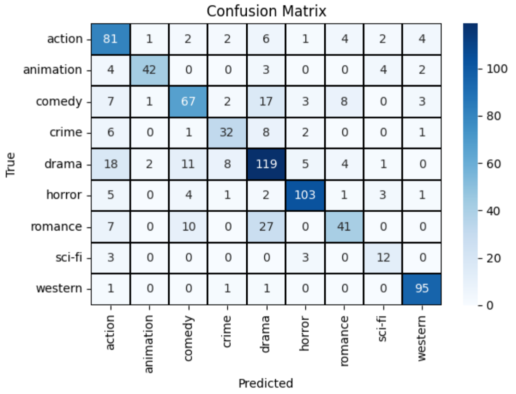
Concerning the BERT model, a pretrained model — 'distilbert-base-uncased' [4] — was used, followed by parameter tuning. The `Adam` optimizer was used with Sparse Categorical Cross-entropy as the loss function, which is appropriate for multi-class classification tasks. The learning rate was fine-tuned to a value of 2×10^{-5}. To prevent overfitting, `EarlyStopping` with a patience of 3 epochs was applied, monitoring the validation loss and restoring the best weights. The model was trained for a maximum of 10 epochs, typically stopping at 6 epochs. As expected, this was the model that achieved the best results, making it the chosen one for making predictions on `test_no_labels.txt` and submission, as BERT effectively captures complex semantic relationships in the text, outperforming SVM and KNN (as shown in Table I). Since BERT was the chosen model, the metrics presented in Table II (reflecting the dataset imbalance) and the confusion matrix in Figure 2, pertain exclusively to its performance.

TABLE I
METRIC RESULTS (GENERAL)

Metric [%]	Model		
	KNN	SVM	BERT
Accuracy	57.39	68.19	72.17
F1-Score	56.19	68.07	71.81

TABLE II
METRIC RESULTS (PER GENRE) - BERT

Genre	Metric [%]	
	Recall	F1-Score
action	78.64	70.39
animation	76.36	85.15
comedy	62.04	62.83
crime	64.00	66.67
drama	70.83	65.71
horror	85.83	86.67
romance	48.24	57.14
sci-Fi	66.67	60.87
western	96.94	93.20



	action	animation	comedy	crime	drama	horror	romance	sci-fi	western
action	81	1	2	2	6	1	4	2	4
animation	4	42	0	0	3	0	0	4	2
comedy	7	1	67	2	17	3	8	0	3
crime	6	0	1	32	8	2	0	0	1
drama	18	2	11	8	119	5	4	1	0
horror	5	0	4	1	2	103	1	3	1
romance	7	0	10	0	27	0	41	0	0
sci-fi	3	0	0	0	0	3	0	12	0
western	1	0	0	1	1	0	0	0	95

Fig. 2. Confusion matrix - BERT.

As an additional test, a small dataset - `plot_bait.txt` - was created[3] consisting of **horror** genre plots infused with **comedy**-related keywords such as "funny" and "laugh." The model correctly classified three out of five deceptive plots as **horror**, demonstrating a robust ability to interpret contextual cues rather than relying solely on specific keywords.

III. DISCUSSION

The metrics in Table I indicate that the model performed well overall. However, Table II highlights specific challenges with the **romance** genre. Further analysis[3] revealed frequent misclassification of **romance** movies (48%) as **drama** (29%). This is likely inherent of the ambiguity and thematic overlap between these genres, such as emotional narratives and interpersonal relationships, compounded by the imbalance in training data (**drama:** 1676, **romance:** 886). Notably, these genre labels are themselves subjective, even for human annotators. For instance, in the 'Plot' of the movie *Hum Ho Gaye Aapke*, the sentence *"Mohan marries someone else"* contributes to genre ambiguity, potentially leading to the misclassification as **drama**.

Regarding the other genres, misclassifications were rare, with the exception of the **comedy** genre (48%). Most of these errors were attributed to the **drama** genre (20%), again due to syntactic ambiguity and the challenges of sentiment analysis. This is particularly evident as both **comedy** and **drama** had the largest sample sizes in the training set (**comedy:** 1193). An example of this misclassification is the movie *The Boy Who Stole a Million*, where the plot describes a boy who borrows money from a bank and is pursued by the police. The ambiguity in this plot contributes to a hard annotation and subjectiveness, even for the group.

IV. FUTURE WORK

One limitation of the chosen model, BERT, is its token input limit of 512, which is easily surpassed by some plots in the dataset. Even with this restriction, the model was able to achieve good results because it is generally able to infer enough valuable information from these tokens. An interesting avenue for evolution of this model would be to discard this restriction altogether, although this is problematic for self-attention mechanisms like BERT given their quadratic complexity regarding input size.

We initially identified the 'Plot' and 'Director' columns as likely to hold the most valuable information. However, it could be beneficial to conduct a more thorough analysis of each column's significance and variability, such as using Principal Component Analysis [5]. This would help focus on linear combinations of variables that capture the most variance, retaining information from columns like "Date" or "Title" that may or may not enhance performance.

Addressing the imbalance in the distribution of genres in the dataset through techniques such as data augmentation, resampling, or applying class weights during training could mitigate overfitting and underfitting, and enhance overall model performance.

[3] Additional images/tests can be found here.

References

[1] P. Cunningham and S. Delany, “k-nearest neighbour classifiers,” *Mult Classif Syst*, vol. 54, 04 2007.

[2] T. Evgeniou and M. Pontil, “Support vector machines: Theory and applications,” vol. 2049, pp. 249–257, 09 2001.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 10 2018.

[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter,” *CoRR*, vol. abs/1910.01108, 2019. [Online]. Available: http://arxiv.org/abs/1910.01108

[5] A. Maćkiewicz and W. Ratajczak, “Principal components analysis (pca),” *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.