News Categorisation: Application of Text Classification of Natural Language Processing

Kai Sun Anson Lam

[redacted]

1 Problem Statement and Motivation

In this coursework, I intend to categorise news in the selected dataset with text classification techniques of Natural Language Processing (NLP). On account of the increasing amount of data generated, while digital news have become more easily accessible, this has created the challenge of identifying and finding contents of ones' interest (Kaur and Bajaj, 2016). News categorisation, hence, has gained significant importance to assist internet users to have efficient access to news of their interest in real time (Kaur and Bajaj, 2016). In an attempt to categorise news in the dataset, it would be useful to apply NLP text classification models to news titles.

2 Research Hypothesis

How effective can news categorisation be performed with NLP text classification techniques? It is assumed that the effectiveness is high. In a bid to assess such effectiveness, performance metrics of built machine learning models are compared with each other, as well as the baseline. Although text classification is one of the most widely used NLP techniques, it is unknown and worth to test whether it is suitable for news categorisation.

3 Related work and background

Kowsari et al. (2019) illustrated an overview of text classification pipeline, including feature extraction, dimensionality reduction (optional), classification (i.e. learning model), and evaluation (i.e. prediction of test data and evaluation of model). Classification, obviously, forms the most crucial part of the whole pipeline. Provided with multiple news categories as classes in the dataset, SVM, particularly multi-class SVM, is the first candidate for classification technique, ranging from One-vs-One to All-vs-One variant to be possibly implemented (Kowsari et al., 2019). Another classification approach is Neural Network (NN), including Deep Neural Network (DNN), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) to be considered (Kowsari et al., 2019). Based on NN, Bidirectional Encoder Representations from Transformers (BERT) is a more advanced approach. Finally, in order to evaluate the models, a number of evaluation metrics, for instance accuracy, sensitivity, specificity, precision, and recall, can be used (Kowsari et al., 2019).

Kaur and Bajaj (2016) specifically illustrated the workflow of news categorisation, reviewed a multitude of algorithms particular for news categorisation, and analysed the pros and cons of different approaches of relevant algorithms. The workflow includes 4 main stages: news collection, news pre-processing, feature selection, and news classification (Kaur and Bajaj, 2016), which is in line with the more general text classification pipeline of Kowsari et al. (2019).

3.1 Accomplishments

A. Majority Class Baseline (MCB) and Support Vector Machine (SVM)

i. Data Pre-processing

- Remove links – Completed
- Clean digits with regular expressions – Completed
- Tokenize and Lemmatize – Completed
- Remove stop words – Completed
- Append texts – Completed

ii. Feature Extraction

- Create bag of words – Completed

iii. Modelling

- Build and train MCB and SVM – Completed

- Test / validate MCB and SVM – Completed
- Perform error analysis on MCB and SVM – Completed

B. Bidirectional Encoder Representations from Transformers (BERT)

i. Data Pre-processing

- Convert labels from strings to integers, then to lists – Completed
- Drop unnecessary dataframe columns – Completed
- Tokenise data with BERT tokenizer – Completed
- Create customised training and testing datasets and dataloaders – Completed

ii. Modelling

- Define hyperparameters – Completed
- Build and train (fine tune) BERT – Completed
- Test / validate BERT – Completed
- Perform error analysis on BERT – Completed

4 Approach and Methodology

A. Core Idea

In an attempt to prove the high effectiveness of NLP text classification techniques to categorise news, SVM and BERT are the 2 built machine learning models as main approaches, alongside MCB. Through comparing performance metrics of all the 3 main models can the overall effectiveness and the most effective model be found.

B. Limitation

Capturing pattern in data, SVM and BERT are expected to outperform the MCB. While the performance of the 2 main models might be affected by the length of string of titles, the capability MCB completely depends on the distribution of data.

C. Working Implementation

i. Majority Class Baseline (MCB)

After loading the dataset, dataframes of training and testing sets are created and modified, specifically filling n/a values with empty strings, transforming string labels to numerical labels, and dropping unnecessary dataframe columns. Data pre-processing detailed in the Dataset section is then performed, followed by creating count vectorizer and bag of words. If training is in progress, the MCB model is created, fitted with training data, and saved; otherwise, if testing is in progress, the model is loaded. The model, finally, performs validation / testing with accuracy and other metrics by predicting the most frequent class.

ii. Support Vector Machine (SVM)

SVM has an implementation similar to MCB with bag of words as feature extractor. The main difference is the actual model created, fitted, and saved / loaded is SVM. With respect to hyperparameters, the kernel type and C value can be tuned. Finally, the model performs validation / testing with accuracy and other metrics by predicting the category of news.

iii. Bidirectional Encoder Representations from Transformers (BERT)

Compared to the previous 2 models, BERT is implemented in a more different way. When dataframes of training and testing sets are created and modified, numerical labels are transformed from string labels further form numerical lists. As for data pre-processing, pretrained BERT tokenizer is used. While hyperparameters, from learning rate to epoch, from training batch size to testing batch size, are defined, training and testing dataloaders are created. BERT model, next, is created with a drop out and linear layer. If training is in progress, the model is trained with a loss function and optimiser, and then saved for inference, otherwise, if testing is in progress, the model is loaded with 'state_dict' for inference. The model, lastly, performs validation / testing with accuracy and other metrics by predicting the category of news.

D. Main Libraries Used

- Numpy: Numerical operations
- Pandas: Dataframe operations
- Joblib: Loading and saving models
- Sklearn: Machine learning models, performance metrics etc.
- Nltk: Text processing
- Torch: Neural network implementation
- Transformers: BERT model

E. Reference to Existing Implementations

In order to build more efficient models, some implementation approaches are with reference to online tutorials with relevant model types being applied to completely different datasets. Links to them are in the References section.

F. Implementation and Relevant Files

Model Type	**Notebook File to Be Run**	**Pre-trained Model File Loaded in Notebook**
MCB	majority_class_baseline.ipynb	majority_class_baseline_model.joblib
SVM	svm.ipynb	svm_model.joblib
BERT	bert.ipynb	bert.pth

G. Issues and Challenges

One of the main challenges is dataset handling. For each model, data has to be transformed to specific formats of data types to facilitate corresponding implementation.

5 Dataset

A. Introduction to Dataset

This is a news dataset containing unique news articles in English language. Data fields include 'Title' (string of news title), 'Excerpt' (string of short extract from news body), and 'Category' (string of category as label). Provided that 'Category' has already been assigned values of either 'business', 'sports', 'politics', 'health', 'entertainment', or 'tech', annotation is not required.

B. Examples of Dataset

For instance, a news article has the title 'Putin Signs Law Paving Way to Rule Until 2036', the excerpt 'Russian President Vladimir Putin has signed a law allowing him to run for two more terms in the Kremlin once', and the category 'politics'. Another example is combined with title 'Oil Prices Near $80 as Omicron Concerns Ease', excerpt 'Oil prices rose over three per cent on Monday on hopes that the Omicron COVID-19 variant would have limited', and category 'business'.

C. Challenging Properties of Dataset

Chief among the challenging factors is the short length of title and excerpt strings in the dataset. In both training and testing sets, the mean length of title and except strings are approximately 70 and 126 words respectively. On account of their short lengths, it might be difficult for the built models to capture similarities among strings of various categories. In order to keep the approach clear while increasing the level of challenge, only title strings (with shorter mean length compared to excerpt strings) are used.

D. Source of Dataset and Basic Statistics

This publicly accessible dataset is available on Hugging Face at https://huggingface.co/datasets/okite97/news-data. It is consisted of 5,514 rows, of which 4,686 and 828 rows form the training and testing sets respectively.

E. Other Relevant Statistics

Data distributions of training and testing sets are highly similar. The weightings of 'business', 'sports', 'politics', 'health', 'entertainment', and 'tech' news are approximately 0.271, 0.240, 0.229, 0.117, 0.081, and 0.062 respectively in both sets. This, nevertheless, indicates that the data in them is quite imbalanced and might not be a good representation of real-world data.

5.1 Dataset pre-processing

A. Majority Class Baseline (MCB) and Support Vector Machine (SVM)

With similar implementation, the text pre-processing techniques used are common for MCB and SVM.

- Remove links using regular expression
- Keep Alphabetical and Numerical Characters and remove other digits using regular expression
- Tokenise and lemmatise to convert word to basic form
- Remove stop words without useful meaning and with high number of appearances

B. Bidirectional Encoder Representations from Transformers (BERT)

Pre-trained BERT tokenizer 'bert-base-uncased', specifically, is used for text processing.

6 Baselines

As mentioned in previous section, Majority Class Baseline (MCB) is the baseline used, which predict the most frequent label with a bag of word approach. This is selected for the following reasons. For one, the high consistency of accuracy of MCB enables it to be a reliable benchmark for other main models. For another, through predicting the most frequent class can MCB provide initial insights of the dataset, in addition to its basic statistics.

7 Results and Error Analysis

A. Results Overview

	MCB	SVM	BERT
Accuracy	0.29 (low)	0.89 (very high)	0.86 (very high)
Precision (Weighted Average)	0.08 (very low)	0.90 (very high)	0.91 (very high)
Recall (Weighted Average)	0.29 (low)	0.89 (very high)	0.88 (very high)
F1 Score (Weighted Average)	0.13 (very low)	0.89 (very high)	0.87 (very high)

i. Performance Metric Value Classification

Very high:	0.85 <= value <= 1
High:	0.60 <= value < 0.85
Medium:	0.40 <= value < 0.60
Low:	0.15 <= value < 0.40
Very low:	0 <= value < 0.15

ii. Weight Average

Weighted average gives more weight to classes with more samples.

B. Detailed Results

i. Accuracy

Accuracy = (TP + TN) / (TP + TN + FP + FN)

While SVM and BERT have very high accuracy, MCB has low accuracy. SVM is the model with highest accuracy.

ii. Precision (Weighted Average)

Precision = TP / (TP + FP)

While SVM and BERT have very high precision, MCB has very low precision. BERT is the model with highest precision.

iii. Recall (Weighted Average)

Recall = TP / (TP + FN)

While SVM and BERT have very high recall, MCB has low recall. SVM is the model with highest recall.

iv. F1 Score (Weighted Average)

*F1 Score = 2 * Precision * Recall / (Precision + Recall)*

F1 score is the harmonic mean of precision and recall. While SVM and BERT have very high F1 score, MCB has very low F1 score. SVM is the model with highest F1 score.

C. Results Discussion

All in all, the effectiveness of news categorisation using NLP text classification is high, provide that all the performance metrics of both SVM and BERT are very high. It is, nonetheless, difficult to say either of them has an obvious advantage over the other.

D. Error Analysis

i. Majority Class Baseline (MCB)

MCB has performance ranging from very low to low because of multiple classes. Since it only takes the distribution of class labels into account, its performance is inversely proportional to the number of classes.

ii. Support Vector Machine (SVM) and Bidirectional Encoder Representations from Transformers (BERT)

While performances of SVM and BERT are very high and close to each other, both models may incorrectly categorise news when the input title strings are very short.

8 Lessons Learned and Conclusions

A. Lessons Learned

In this coursework, I have gained hand-on experience of applying text classification techniques of NLP to news categorisation. I have especially found text processing techniques useful because they can also be used to pre-process datasets for machine learning tasks other than NLP. Eventually, I have achieved the goal of proving the high effectiveness of NLP text classification to categorise news with the built models, but there are also challenges encountered. With regards to model inputs, I have to additionally transform data, specifically news titles, to particular formats or data types, such as bags of words and lists, to cater the need of different models.

B. Conclusions

It can be concluded that the built SVM and BERT are very effective models for news categorisation, given their performance metrics ranging from very high to high. Capturing useful patterns in the dataset, these 2 main models outperform MCB with respect to all performance metrics. It, however, is difficult to say that SVM has better performance compared to BERT, and vice versa. There is, nevertheless, room for improvement. For BERT, more different combinations of hyperparameters can be tried produce better results. In the future, more sophisticated models, such as DistilBERT, can be implemented to achieve higher performance.

References

Gurmeet Kaur and Karan Bajaj. 2016. News classification and its techniques: a review. IOSR Journal of Computer Engineering, 18(1):22–26.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. Information, 10(4):150.

Prakash, P. (2023, May 17). Understanding Baseline Models in Machine Learning. Medium. https://medium.com/@preethi_prakash/understanding-baseline-models-in-machine-learning-3ed94f03d645#:~:text=The%20baseline%20classifier%2C%20such%20as

Text Classification: SVM Explained. (n.d.). Kaggle.com. Retrieved May 15, 2024, from https://www.kaggle.com/code/mehmetlaudatekman/text-classification-svm-explained/notebook

Google Colab. (n.d.). Colab.research.google.com. Retrieved May 15, 2024, from https://colab.research.google.com/github/abhimishra91/transformers-tutorials/blob/master/transformers_multi_label_classification.ipynb