TEXT DATA LABELLING USING TRANSFORMER BASED SENTENCE EMBEDDINGS AND TEXT SIMILARITY FOR TEXT CLASSIFICATION

Amiya Amitabh Chakrabarty

Data science specialist, Broadridge financial solutions, Hyderabad, India

ABSTRACT

This paper demonstrates that a lot of time, cost, and complexities can be saved and avoided that would otherwise be used to label the text data for classification purposes. The AI world realizes the importance of labelled data and its use for various NLP applications.

Here, we have labelled and categorized close to 6,000 unlabelled samples into five distinct classes. This labelled dataset was further used for multi-class text classification.

Data labelling task using transformer-based sentence embeddings and applying cosine-based text similarity threshold saved close to 20-30 days of human efforts and multiple human validations with 98.4% of classes correctly labelled as per business validation. Text classification results obtained using this AI labelled data fetched accuracy score and F1 score of 90%.

KEYWORDS

Sentence embeddings, Cosine threshold score, BERT, Pre-trained model, Tokenizer, Semantic similarity.

1. INTRODUCTION

In this document, we would explain about an NLP based use case with unlabelled text data and how AI based data labelling saved lot of time and cost yet fetched a higher text classification accuracy. The goal of this use case was to "Automate and reduce human intervention to classify emails into four categories (henceforth referred to as Cat-1, Cat-2, Cat-3, Cat-4 and Cat-5, actual name concealed due to confidentiality purpose). Based on the category selected, an auto reply format (with steps to follow to meet their requirements) would be used to reply to the customers which would be picked up based on the category predicted by AI."

Daily, there would be lot of emails from customers seeking assistance for various requirements that requires huge human effort to identify the correct category and then share a fixed template with steps to be followed.

To classify, we needed substantial historical labelled data to train our model [1]. While the historical data was available, the challenge was with the unlabelled data. The business realized it would take significant human efforts to label the data. Another piece of information that was provided to us was the fixed templates (text of 500-1000 words) that was used to reply to the emails based on their requirements/category.

DOI: 10.5121/ijnlc.2022.11201

We used these fixed templates for each category to label the data. Once the data was labelled, text classification was performed using BERT (Bi-directional Encoder Representations from Transformers) [2][6][10].

2. OUR APPROACH

The following steps were taken to address this use case.

2.1. Data

Historical data was collected by connecting to the relevant outlook folder. Total 6000 samples were collected for all the 5 categories. The collected data had no labels associated with it and hence correct categorization and labelling of the historical data was the need of the hour to go ahead with text classification approach. We applied AI based text labelling on this historical data and shared the labelled data with business for validation. Based on business validation, it was observed that 98.4% of the historical were correctly labelled by AI. We also re-labelled few of the incorrect samples correctly as per feedback from business and used this labelled data for further text classification.

2.2. Sample Auto Reply Template

We were provided with the email reply text templates for each of the five categories. These textbased templates were used to label the historical data. Below is one of the template formats for one category, Cat1. Few of the terms are hidden due to confidentiality clause.

Dear XYZ,

Thank you for contacting ABC.

Please proceed with one of the below to update your XXXX:

- Contacting our call center so a representative can perform XXXX for you. o Our dedicated call center representatives may be contacted at XXXX or the International Toll number XXXX. After dialling, you will first be connected to our XXX which allows XXXX to access their XXXX in an automated fashion, providing XXXX specific information over the phone or XXXX certain XXXX through specific options after entering your XXXX. If the XXXX you enter are not recognized by our XXXX, you will be placed in queue for the next available representative. We appreciate your patience if there is a wait to speak to our dedicated XXXX.
- Completing and returning the attached XXXX form with the XXXX signature.
- the signature of all XXXX registered to the account is required. o If the XXXX is held XXXX, the XXXX must XXXX the request.
- If submitting a request for an XXXX or on behalf of a XXXX, your XXXX must come in writing and supporting XXXX must be submitted with the XXXX.

2.3. AI Based Text Labelling

We used the concept of BERT [6][10] based sentence embeddings [2][7] and applied cosinebased text similarity [3][8] on the top of it to find all the samples which are closer to the corresponding class template. Following steps were performed:

- Taken auto reply template for Cat1 and converted it into a vector.
- Taken all the historical 6000 samples and converted them into sentence vectors [2][6].
- Find the samples that have the smallest distance in terms of cosine angle and high cosine score [3].
- Based on the analysis and cosine threshold score [3] selected for Cat1, tag all the selected samples as Cat1.
- Apply all the previous steps for all the remaining categories, Cat2, Cat3, Cat4 and Cat5.
- We now have all the samples which have semantic similarity [3] to the auto reply templates and hence all the historical email samples were labelled.

2.3.1. Sentence Embedding using Sentence BERT

BERT stands for Bidirectional Encoder Representations from Transformers [2]. As the name suggests, it's a transformer-based technique predominantly used in NLP applications. Its basically a transformer model with number of encoder layers and self-attention.

When the text to be used is large text, using tokens/words would be very long process and leading to information limitations when we extract from the word embeddings. Sentence embeddings on the other hand, would represent the sentences and its semantic information in the form of vectors. This certainly helps to understand the context and intention of the larger text better as compared to the word embeddings.

We have used ***bert-base-nli-mean-tokens*** from ***sentence-transformers [2][6]*** module that maps samples to a 768-dimensional dense vector space. Below is the dimension of auto-reply format converted to vectors:

```
d = ['Dear XYZ,Thank you for contacting ABC.Please proceed with one
embed = model.encode(d)
embed.shape
```

```
(1, 768)
```

Figure 1. Sentence embedding dimension for auto-reply email text.

Next, entire 6000 historical samples were converted to vector space of 768 dimensions as shown below:

```
sen_embeddings = model.encode(df['Clean'])
sen_embeddings.shape
```

```
(6000, 768)
```

Figure 2. Sentence embedding dimension for historical 6k email samples.

2.3.2. Text Similarity using Cosine Similarity

Cosine similarity [3][8]is a text similarity technique to evaluate closeness between two ndimensional vectors[8] and is calculated as the cosine of the angle between two vectors in an ndimensional space.

```
from sklearn.metrics.pairwise import cosine_similarity
cosine_similarity(
    [embed[0]],
    sen_embeddings[0:]
)
```

```
array([[0.77923334, 0.77923334, 0.73668224, ..., 0.8095504 , 0.714333  ,
        0.10844399]], dtype=float32)
```

Figure 3. Cosine similarity between 6k samples and auto reply text

2.3.3. Text Samples Labelling

Once, we extracted the cosine score for all the samples against the selected category reply format, we started analysis of each category against the samples and determined a threshold cosine score below which all the samples were discarded for the chosen category.

The threshold scores selected for all the 5 categories are as follows:

Cat1: 0.962
Cat2: 0.955
Cat3: 0.85
Cat4: 0.93
Cat5: 0.90

Hence, in this way, all the historical samples were labelled based on the corresponding auto-reply formats.

2.4. Multi-Class Text Classification

Text Classification is a technique to make the machine learn from the historical patterns and make predictions on text data by categorizing each text to one of the classes [9]. Here, we have 5 classes and hence our use case qualifies for multi-class classification [9]. We have used TensorFlow 2.x for this use case. We used following workflow:

- Reading the Dataset
- Text pre-processing
- Getting BERT Pre-trained model and tokenizer
- Training the model and evaluation of the model.
- Inference
- Model monitoring and retraining.

2.4.1. Reading the Dataset

The historical data of 6000 samples was fed as the input. This data is now labelled as explained above with all the 5 classes.

2.4.2. Train-Test Split

6k samples were split into train and test dataset with close to 90% data assigned for training purpose. Stratified sampling technique [4][5] was used to split the data to ensure correct distribution of the class labels.

Training classes distribution

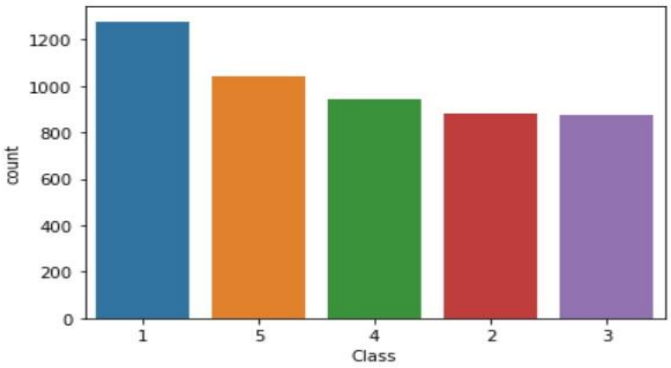

Figure 4. Train data distribution

Test classes distribution

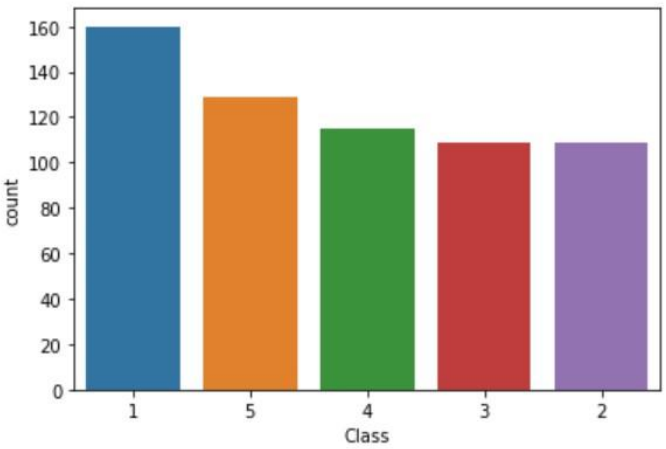

Figure 5. Test data distribution

2.4.3. Text Pre-Processing

We do less text pre-processing here since we are using transformer-based model and not count based methods. Some of the steps include:

- Removing emails, web URLs.
- Converting accented characters.
- Remove whitespaces and extra spaces.
- Removing special characters.
- Lower case.

2.4.4. Getting BERT Pre-Trained Model and Tokenizer

The BERT model [3] generally expects text data in lowercase [3]. Here we have used Hugging face transformers' *Bert Tokenizer*, that converts words into word pieces [10]. WordPiece is commonly used technique to transform words into subword-level in NLP. Here, the most frequent occurring combinations of the symbols in the vocabulary are iteratively added to the vocabulary.

2.4.5. Model Training and Evaluation

Model Network Params

N_EPOCHS = 12
BATCH_SIZE = 36
MAX_LEN = 300
LR = 2e-05

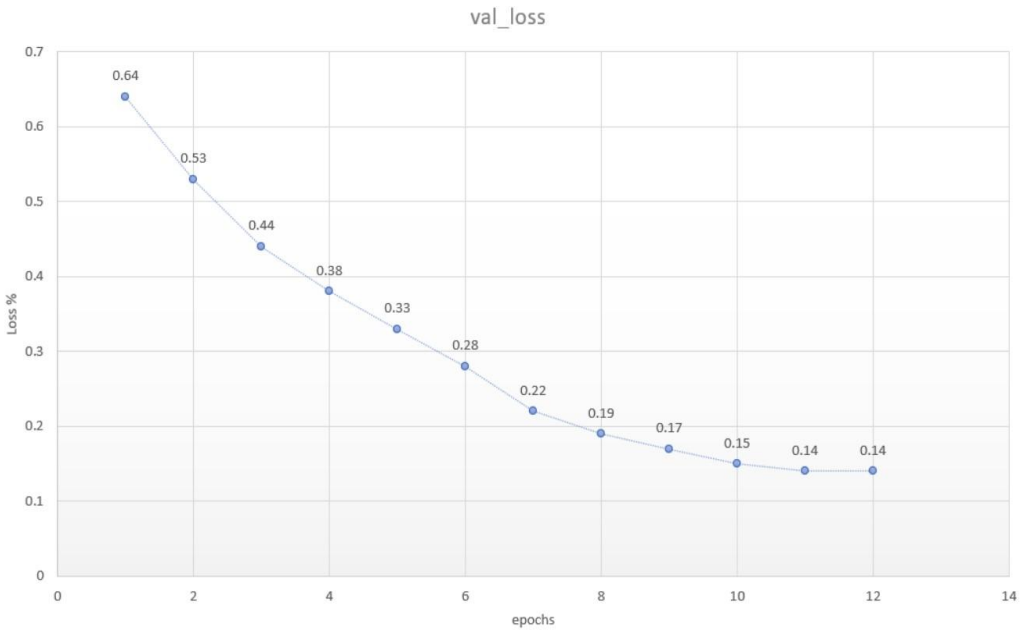

Figure 6. Epochs vs loss%

Model evaluation on test data

As shown in the below figure 7, the accuracy and F1 score [1] for this multi-class classification use case yield 90%.

	precision	recall	f1-score	support
1	0.93	0.93	0.93	160
2	0.97	0.99	0.98	109
3	0.85	0.81	0.83	109
4	0.83	0.90	0.87	115
5	0.89	0.85	0.87	129
accuracy			0.90	622
macro avg	0.90	0.90	0.90	622
weighted avg	0.90	0.90	0.90	622

Figure 7. Test data accuracy metrics

2.4.6. Inference

During real time inference, every email sample goes through the necessary pre-processing and data cleaning steps. Then the BERT [2] [10] stored model version with highest accuracy and F1 score [1] is used to predict the sample as one of the 5 categories: Cat1, Cat2, Cat3, Cat4 or Cat5. This classified email is then moved to the corresponding folder and the email is replied to the customer using Bot with auto reply format text attached for the further course of action that may be taken by the customer.

2.4.7. Model Monitoring and Retraining

Once the model is deployed in production, performance of the model is monitored periodically to ensure that model prediction power is intact and not degrading and the latest performance results is as per the expected accuracy threshold. We therefore collect results at the end of every 3 months and evaluate the deviations from actual. If there is significant deviation in the predicted output, we retrain the model by considering latest data.

3. CONCLUSION

In this paper, we demonstrated an AI based approach to label the historical text data and how to overcome the challenge of unlabelled text data required for text classification. We then performed email classification and achieved an accuracy which is above the threshold accuracy set by business and stakeholders to make a business impact. This has helped business save lot of manual efforts without making any compromise in customer satisfaction.

ACKNOWLEDGEMENTS

I thank the leadership team of Broadridge and business for giving me the opportunity to work on this project. I also thank my colleagues who helped during this journey.

REFERENCE

[1] Yang, Y. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1, 69–90 (1999).

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2019).

[3] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval2017 task 1: Semantic textual similarity multilingual and cross lingual focused evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017).

[4] Ng, A.Y. Preventing “overfitting” of cross-validation data. Machine Learning: Proceedings of the Four-teenth International Conference (1997).

[5] William Gemmell Cochran. Sampling Techniques (1953).

[6] Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERTNetworks (IJCNLP 2019).

[7] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi and Y. Goldberg, "Analysis of sentence embedding models using prediction tasks in natural language processing," in IBM Journal of Research and Development, vol. 61, no. 4/5, pp. 3:1-3:9, 1 July-Sept. 2017.

[8] A. W. Qurashi, V. Holmes and A. P. Johnson, "Document Processing: Methods for Semantic Text Similarity Analysis," 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2020.

[9] Ahmed Faraz. A COMPARISON OF TEXT CATEGORIZATION METHODS (2016).

[10] G. Soyalp, A. Alar, K. Ozkanli and B. Yildiz, "Improving Text Classification with Transformer," 2021 6th International Conference on Computer Science and Engineering (UBMK), 2021

AUTHOR

Amiya Amitabh Chakrabarty, is a proven data scientist having total industry experience of 11 years with 7 years into Data science, ML and DL. Amiya, with the help of AI based technologies, has helped automate business from multiple domains/sectors that includes Finance, automobile, Telecom, Customer analytics, banking domain to name a few.