Authorship Classification With BERT

Chloe Sheen **Tina Huang** **Joseph Liu** **Worthan Kwan** **Mia Mansour**

Department of Computer and Information Science

Abstract

Authorship classification is an essential topic in Natural Language Processing, and it can be used in tasks such as identifying most likely authors of documents, plagiarism checking, and as a new way for recommending authors to readers based on the reader's preferred style of writing. In this project, we explored this problem at different levels, with different deep learning models and with different implementations of combining BERT embeddings with bag of words in a neural network on the Reuters_50_50 dataset. Out of all of the models we tested, we achieved the best result of 92.9% with sentence embeddings output by BERT as features as well as a bag-of-words that were fed into a simple forward-feed neural network, using an end-to-end embedding and classification method.

1 Introduction

Authors have unique writing styles to their works and are often consistent across a range of different topics and document styles. There have been numerous approaches to handle this task of characterization and classification of authorship, including using a bag-of-words model or Word2Vec. In particular, we aim to experiment with a language representation model called BERT for this classification task. The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications (Devlin et al., 2018).

Our contributions are as follows:

- We walk through our choice of data and the data pre-processing step, explore the different models that we used for authorship identification, and present the experimental results on tuning different hyperparameters.
- Further, we show that using both the sentence embeddings outputted by BERT as features and bag of words to feed into a simple forward-feed neural net, then combining the embedding and classification within a single neural net to form an end-to-end differentiable classifier (Figure 1, Right) provides the most accurate results.

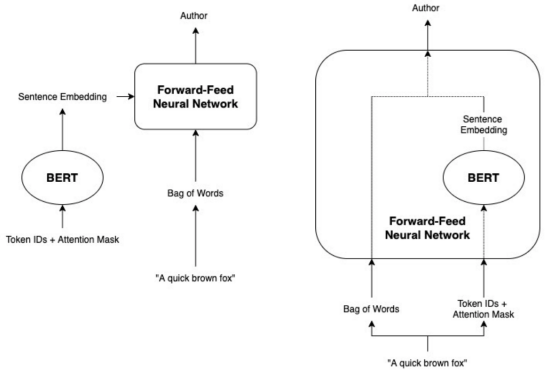

Figure 1: Two implementations of combining BERT embeddings with bag of words in a forward-feed neural net.

2 Literature Review

As mentioned, there has been a long history of handling the task of authorship classification. Here, we briefly describe the approaches that are used and how well the approaches worked.

2.1 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2018)

This paper introduced the new language representation model Bidirectional Encoder Representations from Transformers (BERT). BERT was designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. The pre-trained BERT model can then be fine-tuned

with just another output layer to create models for a wide range of tasks, including but not limited to answering questions, language inference, and classification.

The breakthrough of BERT was expanding the fine-tuning approaches of the pre-trained representations to be bidirectional. The authors proposed BERT to alleviate the unidirectionality constraint by using a masked language model pre-training objective that randomly masks some of the tokens from the input with the objective to predict the original vocabulary id of the basked word based on its context. Additionally, they used a next sentence prediction task that jointly pre-trains text-pair representations.

With BERT, the authors were able to demonstrate the merits of bidirectional pre-training of language representations, which was in contrast to the then state-of-the-art unidirectional language models and shallow concatenation of independently trained left-to-right and right-to-left LMs. They also showed that pre-trained representations reduce the need for several heavily-engineered task specific architectures by demonstrating that BERT as the first fine-tuning based model that achieved state-of-the-art performance on a large suite of sentence-level and token-level tasks. Finally, BERT advanced the state-of-the-art for eleven NLP tasks.

2.2 Tweet Classification with BERT in the Field of Disaster Management (Ma, 2019)

This paper applies deep learning techniques to address Tweets classification problems in the disaster management field. The goal is to use disaster-related information for emergency response and better disaster management in the fields of outbreak detection, evacuation study, hazard and damage assessments. Accurate message classification is a necessary requirement to make decisions in the field of disaster management. Effective filtering techniques are needed to filter out noisy information in the user-generated data.

The baseline is the bidirectional LSTM with GloVe Twitter embeddings. The authors tested the default BERT, BERT + nonlinear layers, BERT + convolution, and the BERT-based LSTM which attained the best results. The BERT models are built based on the pytorch-pretrained-BERT repository by huggingface. Five metrics were used to evaluate these models: accuracy, Matthews correlation coefficient, precision, recall, and F-1 score.

That are multiple factors that have lead to misclassifications: (1) ambiguity and subjectivity, (2) lack of context, (3) non-ascii words emojis, (4) semantic misconstrue (sarcasm, metaphors..), (5) keyword influence or misleading hashtag, (6) events that might not happen, (7) short messages, (8) debatable source. If the quality of the data were to be improved, that would boost up the performance of the classifier.

2.3 Deep Learning based Authorship Identification (Qian et al., 2017)

This paper applies deep learning to author classification of a Reuters 50-50 (news) and Gutenberg (story) dataset. The three major models were a GRU (Gated Recurrent Unit) network , LSTM (Long Short Term Memory) network, and a Siamese network. The most impressive models are almost universally deep learning based neural networks. Variation between models comes from the vector representations of the words.

The first dataset is the Reuters 50-50 (called C50) subset of RCV1. RCV1 Is an archive of over 800,000 manually categorized newswire stories. The top 50 authors by total article size were used. Each selected text had at least one subtopic of the class CCAT (corporate/industrial) in order to minimize the odds of classifying by topic accidentally.

The second dataset is the Gutenberg dataset consisting of over 53,000 books. 50 of the most popular 100 authors were chosen. The books cover a wide range of topics and styles in order to minimize the odds of classifying by topic accidentally. Books were edited to remove noise (page numbers, table of contents, information on contributors).

For modeling, GloVe[10] word vectors of size 50 were used on pre trained word embeddings with a total vocabulary of 400,000 tokens. Numbers and special characters were eliminated to allow matching of word representations during parsing.

After optimization, the Article-level GRU performed best on authorship identification with an accuracy of 69.1% on C50 and 89.2% on Gutenberg. The highest testing accuracy of LSTM was 62.7%. The siamese model was used for verification and achieved an accuracy of 99.8% on both datasets.

We will be using this paper to implement as our published baseline. The paper explores the two

datasets we are most interested in: the Reuters 50-50 dataset, along with different deep learning models. Since the publication of this paper was in 2017, and given the introduction of the BERT language representation model in October 2018, we believe that the neural networks implemented in this paper would serve as a good baseline to test the performance of a newer model on the same dataset.

3 Experimental Design

3.1 Dataset

Reuters_50_50 is a subset of Reuters Corpus Volume 1 (RCV1), which is a collection of over 800,000 English language news stories dating from August 20, 1996 to August 19, 1997 that have been made available by Reuters, Ltd. We chose this corpus because it has already been used in author identification experiments (Qian et al., 2017). The dataset consists of 5000 stories by the top 50 authors (according to the number of stories written) with at least one subtopic of the class CCAT (corporate/industrial) selected to minimize the topic factor in distinguishing among texts. The average number of words per document is approximately 500. The train/test split is 50:50 with each set consisting of 2,500 texts (50 per author). However, in a similar experiment by (Qian et al., 2017), the dataset was reorganized into a 9:1 train/test split, so in order to benchmark our results accordingly, we went with a 8:1:1 train/val/test split.

3.2 Evaluation Metric

The chosen evaluation metric is F1 score. However, Precision, Recall, Matthews Correlation Coefficient, and Accuracy were also calculated for BERT models in order to get a more complete view of model performance. Matthews Correlation Coefficient (originally Pearson's phi coefficient) is a common machine learning metric which uses observed and predicted binary classifications to score a value $[-1, 1]$. -1 represents complete incorrect prediction, 1 represents completely correct prediction, and 0 represents the equivalent of random prediction.

The formula for MCC is as follows.

$$|\text{MCC}| = \sqrt{\frac{X^2}{n}}$$

where n is the total number of observations.

3.3 Data Pre-Processing

Pre-processing is split into two parts: pre-processing required for BERT models and pre-processing required for non-BERT models.

3.3.1 BERT

Data preprocessing for BERT is dependent on which pre-trained BERT model is used. In this section, we describe the process for the base BERT model.

Several steps are required for input into BERT:

- Tokenization
- Addition of special tokens
- Creation of attention mask

BERT uses WordPiece tokenization, which means a word may be broken down into several pieces. As a result, many articles become over a thousand tokens long, which is well over the 512 maximum length limit, so the number of tokens are truncated in order to fit the data. After tokenization, BERT requires the addition of special tokens, namely the classification token, [CLS], which is added to the start of every sentence, and the separation token, [SEP], which is added to the end of every sentence. Finally, an attention mask is required as additional input that signals to the model which tokens to focus its attention on.

3.3.2 Non-BERT

Data preprocessing for non-BERT models focused on using techniques such as Bag of Words and TF-IDF. While BERT and its respective encodings are involved in most if not all of the leading author classification models, these other techniques were used in conjunction and independently of BERT.

Bag of Words was developed in several steps:

- Punctuation Removal
- Lemmatization
- Stopword Removal
- Vectorization (transformation to Bag of Words)

Bag of Words performs best when words are reduced to base form and special characters are removed. Processing data in the way allows models to more easily differentiate between authors without requiring as much data. TF-IDF was developed using a vectorizer with removal of stopwords and a specified maximum vector size. This was utilized to remove infrequent words which would not be informative to our model.

3.4 Models

In this section, we elaborate on the different models that we used in our experiment.

3.5 Baseline

We establish two baselines. The first is a naive guess baseline and the second is a Bag Of Words Baseline based on the common Bag of Words implementation in Author Classification.

3.5.1 Random Forest

Random Forest models consist of a multitude of decision trees. Their primary benefits for our problem include feature engineering and a strong aversion to over-fitting.

3.6 Neural Net

Forward Feed Neural Networks or Multilayer Perceptrons are relatively simple neural nets where inputs are mapped to outputs through an approximated model. This was chosen for its simplicity allowing us to utilize more complex features while still retaining a relatively simple model which performs strongly on classification tasks. Another option would have been a Recurrent Neural Network or RNN. RNN's are popular in NLP for their ability to process sequenced words.

3.6.1 XGBoost

XGBoost is a boosted gradient tree algorithm. Each new iteration of a tree is made to correct the residual errors of the last iteration. This model was chosen for its strong performance in modern data science and classification tasks.

3.6.2 BERT for Sequence Classification

BERT is a bi-directional transformer model released by Google that has broken records for several language-based tasks. Since then, numerous other variations have been made, which are available via *Huggingface*'s transformers library. We selected those that were adapted for sequence classification in English. Table 1 provides a summary of the selected pre-trained models. In spite of the different architectures, the general premise remains the same: after pre-processing, each token input outputs a vector of the same size as the number of hidden units (in the original BERT model, this size is 768). The first token of every input is the [CLS] token, and it's output vector contains all the information that is representative of the sequence. It is this vector that is used in classification and the rest is discarded.

4 Experimental Results

4.1 Simple Baseline

The simple baseline is an implementation of a completely naive classification model. One would expect an individual with absolutely no knowledge of authors to predict 1 in 50 correct due to the equal distribution of texts per author. The baseline predicted the same author for every text and obtained an F1 and Accuracy of .02.

4.1.1 Bag of Words Baseline

The Bag of Words baseline is an implementation of a bag of words feature classified using a random forest. The data is cleaned for punctuation and special characters, and stop words, and is lemmatized. 10000 estimators results in an F1 score .47 and Accuracy of .49.

We based our baseline off of the ideas of *Text Classification by Augmenting Bag of Words Representation with Co-occurence Feature* (K and Joseph, 2014). The use of bag of words is a very common baseline for NLP models. Our Bag of Words model fell far short of the published baseline which achieved an F1 score of ~.80 at minimum. One reason for this was the large number of classes we had compared to the published baseline (50 vs 20).The published baseline used wikipedia data to classify newsgroups while our data was the classification of authors writing about similar subjects. Thus genre specific words were much less applicable or helpful in our model. Additionally as we wanted to use Bag of Words as a sub feature to BERT we chose not to modify it or support it with other features such as TF-IDF or LDA in our base model.

4.2 BERT

In our search for the best BERT model for the task, we tuned the following hyperparameters:

Architecture	Model Name	Description
BERT	bert-base-cased	12 layer, 768 hidden, 12 heads, 110M parameters. Trained on lower-cased English text.
	bert-base-uncased	12 layer, 768 hidden, 12 heads, 110M parameters. Trained on cased English text.
	bert-large-cased	24 layer, 1024 hidden, 16 heads, 340M parameters. Trained on cased English text.
XLNet	xlnet-base-cased	12 layer, 768 hidden, 12-heads, 110M parameters. XLNet English model.
XLM	xlm-mlm-en-2048	12 layer, 2048 hidden, 16 heads. XLM English model.
RoBERTa	roberta-base	12 layer, 768 hidden, 12 heads, 125M parameters. RoBERTa using the BERT-base architecture.
Distilbert	distilbert-base-cased	6 layer, 768 hidden, 12 heads, 65M parameters. The DistilBERT model distilled from the BERT model bert-base-cased checkpoint.
ALBERT	albert-base-v2	12 repeating layers, 128 embedding, 768-hidden, 12-heads, 11M parameters. ALBERT base model with no dropout, additional training data and longer training.
XLM-RoBERTa	xlm-roberta-base	125M parameters with 12 layers, 768 hidden-state, 3072 feed-forward hidden-state, 8 heads, Trained on on 2.5 TB of newly created clean CommonCrawl data in 100 languages.

Table 1: Summary of different variations of BERT models used

- Max sequence length: 128 or 512.
- Cased or uncased: whether the model was pre-trained or cased or lower-cased English text.
- Variations of BERT architecture.

The results be can found in Table 2. These results were obtained using the AdamW optimizer (Adam with weight decay) and an initial learning rate of 4e-5, which was recommended in the original BERT paper. The learning rate follows a schedule that linearly decreases after linearly increasing during an initial warmup period.

Models that have been pre-trained on cased text performed better than models that have been pre-trained on lower-cased text. A higher maximum sequence length confers better performance, although not as much as we'd expect. This seems to indicate that a maximum length of 128 tokens is enough to capture the essence of a writer's style. Out of all the different architectures, XLM performed the best, which is surprising because XLM was adapted for cross-lingual tasks. However, it also varies the way it does preprocessing (Lample and Conneau, 2019), which may explain the difference in results.

4.3 BERT Embeddings + Bag of Words Random Forest

Next, we combined the BERT embeddings with the bag of words representations as features for a simple random forest model. The BERT embeddings significantly improved our model from the previously recorded F1 of 0.49 baseline. We heuristically performed hyperparameter tuning, yielding the best results of 0.68. Results are illustrated in the table below.

4.4 BERT Embeddings + Bag of Words XGBoost

Similarly, we used the BERT embeddings and bag of words as features for the XGBoost model. We performed a random search for the hyperparameter tuning of learning rate, maximum depth, minimum child weight, gamma, and the subsample ratio of columns when constructing each tree. The results of the random search yielded the following best parameter values:

- Learning rate = 0.1
- Maximum depth = 8

Model Type	Model Name	Max Length	Precision	Recall	F1 Score	MCC	Accuracy
BERT	bert-base-cased	128	0.721	0.730	0.712	0.713	0.721
BERT	bert-base-uncased	128	0.631	0.626	0.616	0.615	0.625
BERT	bert-large-cased	128	0.742	0.734	0.727	0.727	0.730
BERT	bert-base-cased	512	0.747	0.746	0.734	0.735	0.742
Distilbert	distilbert-base-cased	128	0.703	0.706	0.688	0.690	0.706
ALBERT	albert-base-v2	128	0.602	0.604	0.586	0.587	0.599
XLM-RoBERTa	xlm-roberta-base	128	0.000	0.020	0.001	0.000	0.020
XLNet	xlnet-base-cased	128	0.759	0.730	0.723	0.728	0.726
XLM	xlm-mlm-en-2048	128	0.773	0.764	0.753	0.756	0.762
XLM	xlm-mlm-en-2048	256	0.756	0.758	0.747	0.747	0.756

Table 2: Performance comparison of different BERT models

Model	F1 Score
Bag of words Baseline	0.49
Bag of Words + BERT Embeddings + Bag of Words (estimators = 100)	0.67
Bag of Words + BERT Embeddings + Bag of Words (estimators = 1000)	0.68

Table 3: Random Forest Result

- Minimum child weight = 7
- Gamma = 0.1
- Colsample by tree = 0.4

The best model yielded a F1 score of 0.69.

Overall, using XGBoost was marginally better than the random forest model.

4.5 BERT Embeddings + Bag of Words Neural Net

In our final experiment we used the sentence embeddings outputted by BERT as features in combination with a bag of words and fed them into a simple forward-feed neural net. We tested two ways of implementing this. The first method maintains a degree of separation between BERT and the neural net. The second method combines embedding and classification within a single neural net to form an end-to-end differentiable classifier.

4.5.1 Separate Embedding and Classification

Figure 2 pictorially depicts our first implementation. We fine-tuned a standard BERT model pre-trained on cased English text with a max sequence length of 128 on our classification task.

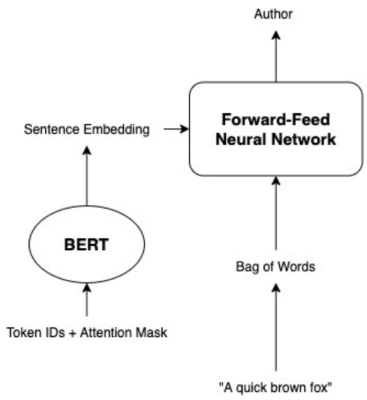

Figure 2: Separate embedding and classification

Method	Precision	Recall	F1 Score	MCC	Accuracy
Separate embedding & classification	0.841	0.828	0.826	0.827	0.827
End-to-end embedding & classification	0.901	0.886	0.886	0.888	0.885

Table 4: Comparison of two different methods of combining embedding and classification

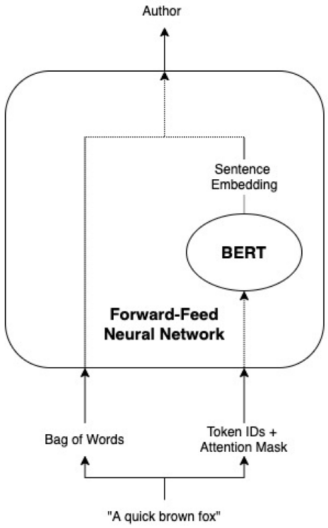

Figure 3: End-to-end embedding and classification

The output embeddings were then averaged and combined with bag of words to feed into a neural net. The neural net consisted of a single hidden layer with a ReLU activation unit and a dropout layer ($p = 0.5$), which was optimized with cross entropy loss as its loss function. We tested various hyperparameters and found the following settings gave the most stable results:

- Optimizer: Adam
- Learning rate: 0.001
- Epochs: 100

The best performance metrics can be seen in Table 4.

4.5.2 End-to-end Embedding and Classification

For the second implementation we decided to incorporate the BERT model as an embedding layer in the neural net so that the neural net can optimize for both embedding and classification. This is visualized in figure 3. The idea stems from the fact that that the first implementation does not optimize the embedding part in combination

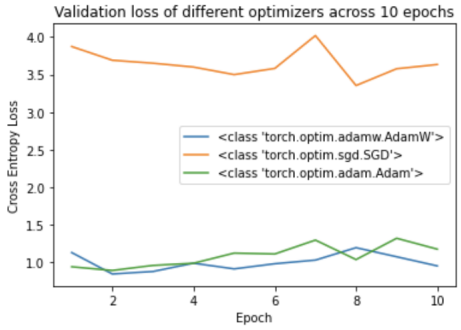

Figure 4: Validation losses of different optimizers

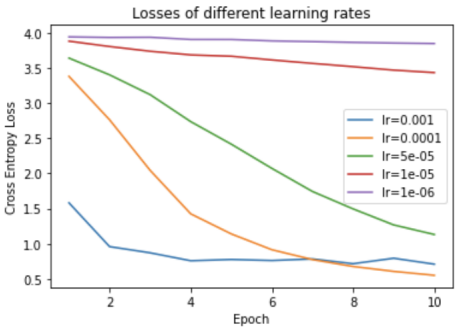

Figure 5: Validation losses using AdamW with different learning rates

with the bag of words. Our hope was that by optimizing for both tasks together, we would be able to improve performance.

Again, we took a standard BERT model pretrained on cased English text with a maximum sequence of 128 tokens, but this time the model was not fine-tuned. The BERT model was incorporated as a layer in the neural net, which again, consisted of a single hidden layer with a ReLU activation unit and a dropout layer ($p = 0.5$). We tested the following hyperparamters:

- Optimizer: Adam, AdamW, SGD (figure 4)
- Learning rates: 1e-3, 1e-4, 5e-5, 1e-5, 1e-6 (figure 5)
- Batch sizes: 2, 4, 8, 16, 32 (figure 6)

We found the following hyperparameters gave the most stable results:

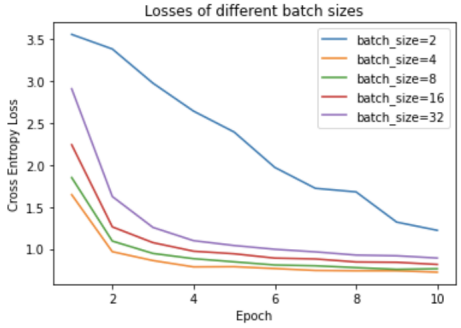

Figure 6: Validation losses using AdamW with lr=1e-4 and different batch sizes

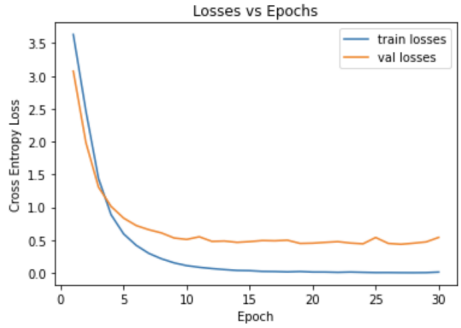

Figure 7: Train vs Val loss using best hyperparameters on max sequence length of 128

- Optimizer: AdamW
- Learning rate: 0.0001
- Epochs: 20

Despite our best efforts by adding a dropout layer and tuning however, it's clear that our model still overfits (figure 7).

A comparison with the separate embedding and classification method in table 4. shows that with the end-to-end implementation, we see a marked improvement in all metrics. After finding the right hyperparameters for a max sequence length of 128, we increased the max sequence length to 512 to deliver our best performance so far with an accuracy and F1 score of 92.9% (table 5).

4.5.3 Error Analysis

In this section, we compare correctly classified texts with incorrectly classified texts from the same author to see what conclusions we could draw. For brevity, we only include snippets of the entire text that we found relevant.

Marcel Michelson:

1. *(CORRECT) The French government on Wednesday called off a controversial sale of the state electronics group Thomson SA to Lagardere Groupe but President Jacques Chirac vowed to bring the sell-off to fruition.*

2. *(CORRECT) The French government called off the controversial sale of electronics group Thomson SA Wednesday in a major policy reversal after weeks of staunchly defending its controversial choice of a buyer.*

3. *(CORRECT) Shareholders of electronics group Thomson SA rubber-stamped an 11 billion franc ($2.12 billion) state capital injection of Friday as the government reviewed how to relaunch the privatisation process. "I believe we need a few days to think things over.*

4. *(CORRECT) The French government Wednesday called off its controversial sale of state electronics group Thomson SA to Lagardere Groupe but President Jacques Chirac vowed to complete the privatisation.*

5. *(CORRECT) Shareholders of electronics group Thomson SA rubber-stamped an 11 billion franc ($2.12 billion) state capital injection of Friday as the government reviewed how to relaunch the privatisation process.*

6. *(CORRECT) The French government on Wednesday halted the sale of electronics group Thomson SA to Lagardere Groupe but said it still wanted to privatise the group to build a large French defence company.*

7. *(CORRECT) The French government on Wednesday called off the controversial sale of electronics group Thomson SA in a major policy U-turn after weeks of staunchly defending its choice of buyer.*

8. *(CORRECT) The French government on Wednesday called off the controversial sale of electronics group Thomson SA in a major policy U-turn after weeks of staunchly defending its choice of buyer.*

9. *(INCORRECT) Within a few years, some two million severely handicapped people in Europe could communicate with each other over the*

Architecture	Model Name	Max Sequence Length	Precision	Recall	F1 Score	MCC	Accuracy
BERT	bert-base-cased	128	0.901	0.886	0.886	0.888	0.885
BERT	bert-base-cased	512	0.940	0.930	0.929	0.931	0.929

Table 5: Comparison of different BERT architectures used in end-to-end embedding and classifcation neural net

global Internet using a special eye-movement control to run a personal computer. A small French team of researchers on Thursday showed the latest prototype of an Apple Macintosh computer that reacts to the subtle movements of the eye.

We examined the beginning of 9 texts by Marcel Michelson above. A brief glance shows that variations of *"The French government on Wednesday called off the controversial sale..."* is used in 6 out of 9 texts and variations of the sentence "Shareholders of electronics group Thomson SA..." is used in 2 out of 9 texts. In fact, a closer read reveals that all 8 of those texts are talking the same issue: the sale of an electronics company called Thomson SA. Unsurprisingly, all 8 of those texts have been classified correctly, while the only incorrectly classified text had nothing to do with Thomson SA nor did it open with a similar sentence as another text.

In general, most authors seem to have one topic they are specialists in. Location seems to be a big factor here and for a news agency like Reuters, this would make sense; correspondents tend to be designated regions of the world to report on. For Marcel Michelson, it seems to be France. In the following example, we examine another author, William Kazer, who reports mostly on China. In his incorrectly classified texts, the model usually predicted another author who also wrote texts about China.

William Kazer:

1. *(CORRECT) China gave new details on Wednesday of the failed launch of a satellite aboard its Long March 3B rocket in February, confirming that the cause lay with the new generation launcher.*

2. *(CORRECT) China has begun shipping corn from key growing areas in the north to other parts of the country, cutting northern stockpiles in a bid to keep weak prices there from falling further. However, domestic corn prices, already under pressure, were likely to extend their slide, industry officials said on Tuesday.*

3. *(CORRECT) China is on target with plans to to promote 100 large chemical groups by 2000 by tapping a $1.6 billion war chest, Minister of Chemical Industry Gu Xiulian said on Friday. The industry hoped that 10 of the groups, which are being groomed to compete in the export market, could eventually list their shares either on domestic or foreign stock exchanges, she said.*

4. *(CORRECT) China vowed on Friday to get tough in its drive to keep banks out of the stock market and accused big financial institutions of fuelling a speculative bubble on the nation's bourses. State media quoted China's top economic policy maker, Vice Premier Zhu Rongji, as warning bankers that they could go to jail for diverting funds into the stock market.*

5. *(CORRECT) A Chinese ideologue known for his strictly orthodox Marxist views denied on Tuesday that he was the author of two essays critical of China's reforms.*

6. *(INCORRECT) China on Tuesday rejected Hong Kong Governor Chris Patten's criticism of its proposal to abolish a series of laws in the territory after it reverts to Beijing's control, saying it was an internal affair. Beijing also angrily reminded Britain that China was no longer a weak, prerevolutionary government, and said it would not tolerate other countries trying to impose their will on it.*

In the following texts, it's clear William Kazer reports mainly on China. In the one text that was classified incorrectly, the model predicted the author as Benjamin Kang Lim. A quick glance of his texts reveal that he too, is also a correspondent for China.

5 Conclusions

In this project, we experimented with various approaches to authorship identification using different models. As shown from section 4.5.2, we were able to achieve an accuracy of 92.9% with sentence embeddings output by BERT as features as well as a bag-of-words that were fed into a simple forward-feed neural network, using an end-to-end embedding and classification method. The state-of-the-art performance on a previously published paper in 2017 (Qian et al., 2017) using article-level GRU achieves a result of 69.1% accuracy on the same dataset. Our implementation was able to reach beyond this performance on the authorship classification task.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Soumya K and Shibily Joseph. 2014. Text classification by augmenting bag of words (bow) representation with co-occurrence feature. *IOSR Journal of Computer Engineering*, 16:34–38.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *arXiv e-prints*, page arXiv:1901.07291.

Guoqin Ma. 2019. Tweets classification with bert in the field of disaster management.

Chen Qian, Ting He, and Rao Zhang. 2017. Deep learning based authorship identification.