Author Age Range Prediction using RoBERTa on the Blog Authorship Corpus

Aja Klevs, Deja Bond, Christina Dominguez
May 2020
ak7288@nyu.edu,ddb345@nyu.edu,ctd299@nyu.edu

1 Abstract

We repeat the 2004 Blog Authorship Corpus' 3-class author age range classification task using BERT, RoBERTa, Bi-LSTM, and a simple Bag of Words Neural Network. Our best RoBERTa model achieves 79.2% accuracy, an improvement of three percentage points over the original paper's 76.2% accuracy result, without the need for manual feature engineering.

2 Introduction

Authorship attribute identification is a popular topic in the world of Natural Language Processing and Understanding. It allows us to classify author attributes for different forms of texts such as books, articles, news, and of course, blogs. Author attribute detection is critical in the forensic investigation of cybercrimes, where there is significant motivation to profile anonymous authors (Yang et al., 2014).

This study employs The Blog Authorship Corpus, composed of over 71,000 blog posts made on a single day on blogger.com in August 2004, and contains self-reported author age information (Schler et al., 2006). The corpus contains an equal number of blog posts from male and female authors, and its original age range classification task which we mirror in our experiments is designed to have three distinct age buckets; 10s (ages 13-17), 20s (ages 23-27) and 30s (ages 33-47). These age categories account for 34%, 48%, and 18% of the dataset respectively. Borderline ages, ie 18-22 and 28-32, were not included in the original study, presumably to have clearer distinctions between the age range classes.

For our study, we apply modern BERT/RoBERTa models to the 2004 Blog Authorship Corpus dataset to understand if they result in improved results for the original paper's 3-class author age range classification task.

3 Related Work

Author attribute detection is a subset of text-based classification tasks that have been studied broadly. We present here a summary of advances in author attribute classification from the past thirty years and explain how our work seeks to evaluate the accuracy of more recent algorithms in terms of age range prediction for the Blog Authorship Corpus.

Until deep learning came to dominate Natural Language Processing (NLP) algorithms in the 2010s, common practice for author attribute detection was to extract word-level or character-level features from documents and utilize traditional machine learning algorithms. Schler and Koppel, authors of the Blog Authorship Corpus, used Multi-Class Real Winnow (MCRW) to predict author age range and gender (Schler et al., 2006).

While the MCRW method achieved as high as 80.1% accuracy on gender prediction and 76.2% accuracy on age range rediction, the feature engineering involved was cumbersome and difficult to replicate. Deep learning techniques address some of these problems, as the only features are the words themselves. For instance, Recurrent Neural Networks (RNNs) have since become a popular text classification tool. RNNs consists of a series of connected sequential cells, each corresponding to a different token. This allows RNNs to capture behavior over the entire sequence and take better account of the word order than a feed-forward approach (Cho et al., 2014).

The Long Short-Term Memory (LSTM) model is a type of RNN that includes a sigmoid function called the *forget gate* which controls which information is retained through each forward pass through time. The *forget gate* makes convergence much more feasible, especially for long sequences. While the basic LSTM was introduced by Sepp Hochreiter and Jurgen Schmidhube as early as 1997 (Hochreiter et al., 1997), it was not widely used in NLP until the second half of the 2010s.

In 2017, Vijay Prakash Dwivedi and others applied a LSTM and a Bi-LSTM (a variation on the traditional LSTM where information not only moves forward through the sequence of cells but also flows backward) to the Blog Authorship Corpus (Dwivedi et al., 2017). They only attempted author gender prediction as opposed to age range prediction, but ultimately achieved an accuracy of 80.3%, which was slightly higher than the original MCRW paper. Other deep learning techniques, including a recurrent convolution neural network, achieved as high as 86% accuracy on gender prediction for the Blog Authorship Corpus (Bartle et al., 2015).

However, many natural language understanding tasks were revolutionized by the discovery of ELMo in March 2018 (Peters et al., 2018) and then BERT in May 2019 (Devlin et al., 2019). BERT utilizes a series of deep learning architecture units called *transformers* to pre-train word embeddings and then use them as features in a deep learning model for a NLP task. BERT significantly outperformed all previous models in a range of text-based classification tasks.

In 2019 Youngjun Joo and Inchon Hwang used BERT to predict author gender on their dataset consisting of Twitter posts (Joo et al., 2019). They achieved an accuracy of 83% with BERT alone, which under-performed some of their experiments with feature engineering and traditional machine learning. However, when they combined their feature engineering approach with BERT, they received their highest accuracy of 88%. A model called RoBERTa discovered in July 2019 improved upon BERT for many tasks by optimizing hyperparameters and training sizes (Liu et al., 2019). As far as we know, no one has previously applied BERT or RoBERTa to the Blog Authorship Corpus.

4 Method

To test the performance of different deep learning models on author age range prediction for the Blog Authorship Corpus, we experimented with a simple three-layer bag-of-words multi-layer perceptron (MLP), a Bi-LSTM, BERT, and RoBERTa. The preprocessing we explain in section 4.1 was only applicable to the MLP and the Bi-LSTM. The BERT and RoBERTa models utilized the Transformers library (Surkov, 2020) for preprocessing.

4.1 Preproccessing

When preprocessing the blog text, we made everything lower-case, removed punctuation, and split by spaces so that each blog post was represented by a list of words. We then created a vocabulary for the data by storing the top 10,000 highest frequency words in the training set.

Each word was assigned a unique integer from 2 to 10,002, saving 0 as the *pad* token and 1 as the *unknown* token. We replaced each word with their corresponding number, truncated long blogs to 250 tokens and and padded short blogs so that each instance had the same length.

4.2 Bag-of-Words Neural Network

We decided to start with a simple feed-forward neural network to give us a baseline deep learning accuracy. The embeddings for the words in each blog were averaged together in the forward pass. The model has three layers, an embedding dimension of 300, a hidden size of 64, and it uses ReLU as its activation function. The optimizer was Adam and the learning rate was .001. As it was a baseline model, the only hyperparameter we experimented with was maximum length. However in our results we only include our best result, which was with 500 words.

4.3 Bi-LSTM

We chose to use a Bi-LSTM as an additional baseline as previous work in the author attribution space reported success in using LSTMs, and chose a Bi-LSTM over a unidirectional LSTM as they typically learn faster and may better understand context. For the Bi-LSTM we utilized the PyTorch nn.LSTM function with the bidirectional parameter set to true. Consistent with the basic neural network, we used an embedding dimension of 300, a hidden size of 64, an Adam optimizer and a learning rate of .001. As with the MLP, we experimented with maximum length thresholds but only include our best result, which was with 500 words.

4.4 BERT/RoBERTa

We were significantly limited in our ability to tune hyperparameters for our BERT/RoBERTa models due to the long training times. We therefore focused most of our tuning efforts on evaluating the impact of changing maximum blog length on our results. Otherwise, we primarily used the default parameters of 3 epochs and learning rate 2e-5. For the best model on our validation set only, we ran for an additional two epochs to see if we could further improve upon accuracy.

We saw in our initial results that RoBERTa was outperforming BERT, and thus due to long-running times, we did not run the full set of length results for BERT and subsequently focused our efforts on RoBERTa only.

5 Results and Analysis

Our RoBERTa models performed best, with our best RoBERTA model achieving 79.24% accuracy on the validation set, and 79.19% final accuracy on our test set.

	Epoch	Max Len	Accuracy
BOW	10	500	68.57
Bi-LSTM	10	500	67.21
BERT	3	128	71.32
BERT	3	250	72.48
RoBERTa	3	128	74.84
RoBERTa	3	250	77.72
RoBERTa	3	400	78.56
RoBERTa	3	500	78.84
RoBERTa	5	500	**79.24**

Table 1: Summary of validation accuracy achieved for all models for 3-class age range classification.

Surprisingly, our Bi-LSTM had a slightly worse performance for age range classification at 67.21% validation accuracy than our Bag of Words model at 68.57%. Switching to a BERT model provided approximately 5-6% additional accuracy over our Bag of Words and Bi-LSTM models, and RoBERTa provided approximately 6% additional accuracy on top of BERT.

In general, increasing the maximum text length of our blogs resulted in increased accuracy as including a higher proportion of the blog's content gave the model additional information for classification. However increasing the blog length used in the models increased our run time, and as shown in Figure 1 beyond 400 words seemed to offer increasingly diminishing returns. For example, increasing the maximum blog length for RoBERTa from 128 to 250 increased accuracy by 2.9%, but increasing from 400 to 500 increased accuracy by only .3%.

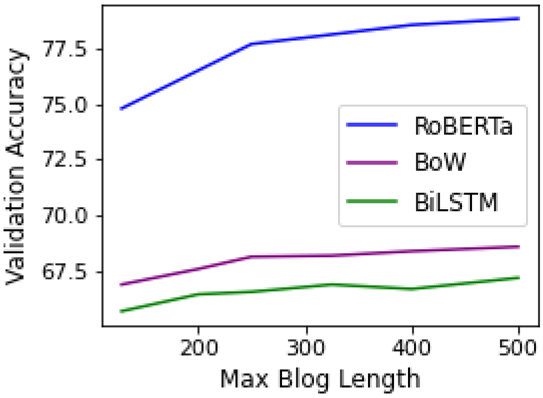

Figure 1: Impact of increased maximum blog length on validation accuracy

Looking at the distribution of blog length for our dataset, the mean length of our blogs was 220, with a median of 126. Only 15.7% of blogs had 400+ words, and 10.4% had 500+ words, which as shown in Figure 2 explains the diminishing returns of increasing length beyond 400 as this increase only impacted a small proportion of our blogs.

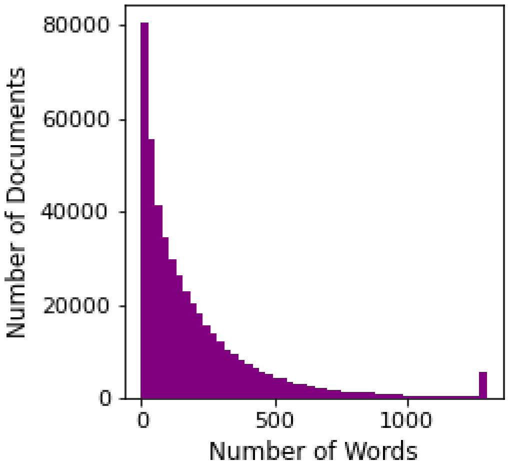

Figure 2: Blog length histogram, where blogs longer than 1,300 words were bucketed together

For our best model RoBERTa with a max length of 500, we also experimented with increasing the epochs beyond the default length of 3, running that configuration for an additional 2 epochs. This bumped our accuracy up from 78.84% to 79.24%, an additional .4%

Looking at our prediction accuracy for each class for our best and final RoBERTa model in Table 2, we saw that the 13-17 and 23-27 age range buckets both achieved high accuracy over 83%, while the model struggled to correctly classify the 33-47 age range bucket, achieving only 59% accuracy and dragging down our overall accuracy result.

Age Range	Test Accuracy
13-17	83.91
23-27	83.82
33-47	59.02

Table 2: Summary test accuracy achieved for all 3 age range classes for best RoBERTa model.

Looking at the confusion matrix in Table 3, we can see that the model frequently misclassified the 33-47 age range bucket as 23-27. This finding mirrors the result in the original paper (Schler et al., 2006), where the best model achieved using MCRW also frequently misclassified the 33-47 age range bucket as 23-27.

	Predictions		
Actual	**13-17**	**23-27**	**33-47**
13-17	29,146	5,641	433
23-27	3,937	40,614	3,906
33-47	643	6,766	10,752

Table 3: Test set confusion matrix for best RoBERTa model.

6 Conclusions and Future Work

Our final RoBERTa test result of 79.2% improves upon the best age range classification accuracy of 76.2% achieved by the original paper (Schler et al., 2006) by 3%. In the future, we believe that increasing the maximum length further, tuning the learning rate, and utilizing RoBERTa with BERT-large architecture could offer additional improvements upon our result.

Compared to both the original paper and RoBERTa, both our baselines performed relatively poorly. This is likely partially due to our lack of hyper-parameter tuning, but the gap is so stark that we suspect the MLP and BiLSTM architectures are not the optimal model architectures to handle this task. The fact that the BiLSTM performed worse than the Bag of Words MLP indicates that the order of words in the Blog Authorship Corpus is not especially relevant for age range prediction.

Furthermore, testing max length thresholds suggest that both the MLP and BiLSTM start getting diminishing returns at around 250 words whereas RoBERTa's performance plateaus at around 400 words. This suggests that RoBERTa is much better at learning useful information about the blogs, since its prediction capabilities continue to increase with more data.

Our use of RoBERTa offers a much more universally applicable framework for author age range classification than the original paper. The original paper used hand-engineered style and content features for age range classification such as parts-of-speech, blog-specific vocabulary, and blog topics, and hand-selected some features based on which had the most significant differences between age classes. Conversely, our use of RoBERTa for author age range classification does not require any manual feature engineering and the same methodology could be applied to blog sets from other time ranges, as well as other content types such as novels or articles. The primary negative of our approach compared to the original paper's use of MCRW would be RoBERTa's long training time and need for more significant computational resources.

While our final author age range classification accuracy improved upon the original paper's result, it was fine-tuned and evaluated on a dataset representing a single day's worth of blogs posted on a single website in August 2004, and utilized self-reported ages which may not have complete accuracy. If applied to corpuses from alternate sources and time ranges, our final model would not achieve comparable accuracy results to the experiments we ran on the Blog Authorship Corpus.

Although our experiments were performed on an older dataset, we suspect that more recent blogs may also contain features that can be exploited by RoBERTa to classify author age. In the future, we would like to further explore this hypothesis and repeat our experiments on more recently published blogs.

7 Collaboration Statement

Christina performed the BERT/RoBERTa experiments, Aja performed the data pre-processing prior to modeling and created a simple baseline MLP, and Deja performed the Bi-LSTM experiments. All three of us helped write the paper.

References

J. Schler, M. Koppel, S. Argamon and J. Pennebaker. 2006. Effects of Age and Gender on Blogging in Proceedings of 2006 *AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*.

Min Yang and Kam-Pui Chow. 2014. Authorship Attribution for Forensic Investigation with Thousands of Authors. *ICT Systems Security and Privacy Protection IFIP Advances in Information and Communication Technology*. pp. 339–350., doi:10.1007/978-3-642-55415-5_28.

Vijay Prakash Dwivedi and Deepak Kumar Singh. 2017. Gender Classification of Blog Authors: With Feature Engineering and Deep Learning Using LSTM Networks. *Ninth International Conference on Advanced Computing (ICoAC)*.

Cho, Kyunghyun, et al. "Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation." doi: arXiv:1406.1078.

Youngjun Joo and Inchon Hwang. 2019. Author Profiling on Social Media: An Ensemble Learning Model Using Various Features. *CLEF*.

Sepp Hochreiter and Jurgen Schmidhube.r 1997."Long Short-Term Memory. *Neural Computation*.

Yinhan Liu et al. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Facebook AI*.

N. Littlestone. 1988. Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Machine Learning 2*.

Jacob Devlin et al. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Google AI Language*.

Aric Bartle and Jim Zheng. 2015. Gender Classification with Deep Learning.

Matthew E. Peters et al. 2018. "Deep Contextualized Word Representations." *Allen Institute for Artificial Intelligence*.

Sergey Surkov. 2020. Huggingface. "Huggingface/Transformers." *GitHub*.