Authorship Attribution Based on Pre-Trained Language Model and Capsule Network

Zeping HUANG† and Mizuho IWAIHARA‡

Graduate School of Information, Production and Systems, Waseda University

2-7 Hibikino, Wakamatsu ku, Kitakyushu-shi, Fukuoka, 808-0135 Japan

E-mail: † huangzeping@ruri.waseda.jp, ‡ iwaihara@waseda.jp

Abstract Authorship Attribution (AA) is a sub-field of Authorship Analysis and text classification, attributing a text to the correct author among a closed set of potential authors. Since short texts usually contain less information about the author, authorship attribution on short texts is often more challenging than authorship attribution on long texts. Recently, the widespread use of pre-trained language models has greatly improved the accuracy of text classification tasks. In this paper, we propose a model which uses the pre-trained language model BERTweet with the capsule network, to solve the authorship attribution on tweets. BERTweet is the first large-scale domain-specific pre-trained language model for English tweets, which can generate high-quality sentence representations of tweets. We combine BERTweet with the capsule network which is particularly powerful at capturing deep features of sentence representations. Thus, both BERTweet and capsule help us achieve remarkable improvements on AA tasks. We also incorporate user writing styles into our model. Our experimental results show that our model shows the state-of-the-art result on the known tweet AA dataset.

Keyword Authorship attribution, Short texts, Social network, Pre-trained language model, BERTweet, User writing styles, Capsule network

1. Introduction

In recent years, with the rapid popularization of the Internet, especially the mobile Internet, social media become more and more important in people's daily life. In addition to interacting with other people, people are increasingly turning to social media. On social media platforms such as Twitter, a large number of tweets are posted all the time. Authorship Attribution (AA) is a task to attribute a text to the correct author among a closed set of potential writers. AA on short texts such as tweets can be applied in a variety of scenarios, such as filtering spams [11], detecting multiple IDs of a particular user, and avoiding identity frauds [14]. AA on short texts is often more challenging than AA on long texts, because short texts usually contain less information about the author.

The most significant difference between AA and other text classification tasks is that the specific style of each author needs to be captured. Existing AA methods focus on capturing the writing style of each author through models such as SVM [14], CNN [15] and RNN [2]. Stylistic features are captured through word and character n-grams, as well as syntactic and semantic information [4][14] have been used in these models. These stylistic features, especially word and character n-grams, show remarkable effects on the performance of the models.

In addition to learning features directly from the original texts (such as word and character n-grams), certain models utilize stylometric features [5] of tweets, such as text length, and the number of URLs in tweets. Huang et al. [7] introduce latent posting style features such as per-user sentiment orientation of tweets, and obtained good results.

Deep learning models have achieved remarkable results in various text classification tasks. Pre-trained language models can be finetuned toward a target task with relatively less amount of labeled training samples. Wang et al. [19] integrate RoBERTa finetuning and user writing styles for tweet AA, achieving the state-of-the-art results so far.

In this paper, we propose a novel method for AA, which exploits BERTweet [10] and capsule network [12]. BERTweet is the first large-scale pre-trained language model trained on large English tweet corpora, which has produced better performance results than the previous state-of-the-art models on three Tweet NLP tasks: Part-of-speech (POS) tagging, Named-entity recognition (NER) and text classification [12]. Since BERTweet is pre-trained on a large number of tweets, tweet-specific contextualities can be exploited for AA on tweets. Secondly, we integrate part of the capsule network into our model. Capsule network is a type of neural network, which can be used to better model hierarchical relationships. The main idea of capsule network is to add capsules, which is a group of neurons, to a convolutional neural network (CNN). The capsule network has shown excellent performance on MNIST [12]. The work [18] applies capsule network on AA, showing effectiveness. Finally, we also incorporate user writing styles [19] into our model.

For evaluation of our model for tweet AA, we adopt baselines from traditional methods and the state-of-the-art model. One baseline is logistic-regression over TF-IDF scores. Since the RoBERTa_CNN model [19] achieves the state-of-the-art on the tweet AA dataset, we also compare the results with the models in [19]. Our experimental results show that our model outperforms these models and achieving the state-of-the-art result.

The rest of the paper is organized as follows. We first give an overview of related work on AA in Section 2. The basic concepts used in our work are discussed in Section 3. In Section 4, we propose our method which is based on BERTweet and capsule network. Section 5 presents our experimental details and evaluation results. Finally, in Section 6, we summarize our work and discuss the future work.

2. Related Work

Authorship analysis is an ensemble of methods that aims to extract useful authorship information of a text by analyzing writing style, which mainly includes three sub-fields, Author Profiling, Authorship Attribution and Authorship Verification. In this paper, we mainly discuss Authorship Attribution (AA). AA has been used on a variety of texts, especially on web data such as blogs [8], forums [16], and emails [1]. AA on tweets is especially important due to the vast amount of tweets and anonymity of users.

Previous work on AA produces a variety of models, such as CNN, RNN, SVM and LSTM [3][14][15][17]. As for modelling of textual features, word and character n-grams [11][14] are often used in these models. Schwartz [14] et al. propose a concept called k-signature that uses flexible patterns to attribute authors' texts, and word and character n-grams are used in their model. Shrestha [15] et al. use the character n-grams as input to CNN to perform AA. Huang [7] et al. propose character embeddings with mixed word and character n-grams, to be used as the input of CNN and LSTM, showing remarkable performance. With the development of deep learning methods, pre-trained language models have also been adopted to generate deep contextualized word representations of texts, and achieved state-of-the-art results. Fabien [5] et al. propose BertAA, which is a BERT model finetuned for authorship attribution. Wang [19] et al. use RoBERTa finetuning to produce post representations and classify the post representations using CNN, showing the state-of-the-art result on the tweet AA dataset.

Capsule network was initially applied on MNIST (Mixed National Institute of Standards and Technology database) and showed excellent performance [12]. However, recent work shows that capsule network can also be used on NLP tasks. Saha [13] et al. introduce BERT-Caps, which is a transformer-based capsule network for tweet act classification. Liu [9] et al. propose a BERT-Cap hybrid neural network model with focal loss for user intent classification to capture user intents in dialogue. Suman [18] et al. present a capsule network-based model through character level n-grams feature for Twitter authorship attribution.

Using stylometric features to characterize writing style can be another aspect. Instead of extracting features directly from the original text, stylometric features focus on the statistical and semantic features of the text. Fabien [5] et al. incorporate stylometric and hybrid features to their BertAA model. Huang [7] et al. introduce an additional feature set with 10 elements and apply these features to the CNN model to improve the AA accuracy. Wang [19] et al. integrate RoBERTa fine-tuning and user writing styles, which learns post representations using a triplet loss function.

In this paper, we propose a model which integrates BERTweet with the capsule network. We further incorporate user writing styles into our model.

3. Background

In this section, the concepts used in this work are discussed.

3.1. BERTweet

Bidirectional Encoder Representations from Transformers (BERT), and its variants have successfully helped produce new state-of-the-art performance results for various NLP tasks. For specific domains, it is possible to retrain a domain-specific model using the BERTology architecture. BERTweet is a large scale pre-trained language model for English tweets, proposed by Nguyen [10] et al. BERTweet uses the same architecture as the BERT base model, which is trained with a masked language modeling objective. The pre-training procedure of BERTweet is based on RoBERTa which optimizes the BERT pre-training approach for more robust performance. When fine-tuning BERTweet for text classification, a linear prediction layer is appended on top of the pooled output. BERTweet performs better than two large models RoBERTa and XLM-R using the same fine-tuning approach on text classification.

What makes BERTweet so effective for Tweet NLP is that 850M English tweets are used for training. The characteristics of tweets are generally different from those of traditional written text, due to the typical short length of tweets and frequent use of informal grammar as well as irregular vocabulary e.g. abbreviations, typographical errors and hashtags. Most of the existing language models are pretrained on large-scale conventional text corpora with formal grammar and regular vocabulary, which make it challenging to apply those language models on tweet data. Also, "soft" normalization strategy [10] is used to the tweets, which is translating word tokens of user mentions and web/url links into special tokens @USER and HTTPURL, respectively, and converting emotion icon tokens into corresponding strings. These processes are based on the characteristics of tweets, which makes BERTweet effective for Tweet NLP.

3.2. Capsule

A capsule is a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part [12]. Capsules capture the spatial relationships between entities and learn these relationships via dynamic routing. This motivates the work using capsule networks for the attribution task [18].

In CNN, the convolution operator is represented by a weighted sum of lower layers, which makes it difficult to carry out these features into upper layers in the case of complex objects [18]. It can be said that CNNs do not capture hierarchical relationships. To overcome these shortcomings, pooling layers are introduced. Pooling can reduce the computational complexity of convolution operations and capture the invariance of local features. However, pooling operations lose information regarding spatial relationships and are likely to misclassify objects based on their orientation or proportion. The capsule network is a structured model, which solves the problems of CNNs. To learn the existence of visual entities and encoding them into vectors, there are locally invariant groups that are known as capsules. Capsule networks transfer iteratively important information to the higher-level capsule through the dynamic routing mechanism, which uses a nonlinear function called squashing for grouping of neurons:

$$S_j = \sum_i W_{ij} \times h_i \tag{1}$$

$$v_j = \frac{\|S_j\|^2}{1 + \|S_j\|^2} \frac{S_j}{\|S_j\|^2} \tag{2}$$

where W_{ij} is the weight matrix multiplied with the capsule layer h_i to represent the upper layer feature S_j. v_j is the output vector obtained after applying the squashing function over the upper layer features (S_j). In this way, capsules help in finding the vector representations of the features. Thus, it differs from the fully connected scalar operations [18].

The output of the dynamic routing mechanism is a higher-level capsule, which retains the important features of the sentence sequence during the iteration process and uses the weight matrix to continuously adjust the acquired features. Finally, a vector output is used to represent the sentence sequence as the input of the classifier [9]. See [12] for more details

4. Methodology

In this section, we first describe our proposed BERTweet_Capsule model for AA. Then we discuss integration of user writing styles with BERTweet_Capsule, using Multilayer Perceptron (MLP).

4.1. Proposed BERTweet_Capsule

We use contextualized word representations from the pre-trained language model BERTweet for tweet AA. The obtained representation is passed through a series of operations and, in turn, hierarchical layers to obtain an optimal vector to classify the tweets to the authors that are described as follows.

1) **Convolutional Layer:** Tweets are fed to the pre-trained language model BERTweet to obtain their sentence embeddings, which are the vector outputs from the last hidden layer of BERTweet. The embeddings are then passed through the convolution layer to produce several feature maps. Then the feature maps are fed to the max pooling layer for the pooling operation:

$$V_{Conv}^i = Conv_i(V_{BERTweet}) \tag{3}$$

$$V_{Pool}^i = Pool_i(V_{Conv}^i) \tag{4}$$

where $V_{BERTweet}$ is a sentence embedding, $Conv_i$ and $Pool_i$ are convolution and pooling operations. Index $i \in [1, n]$, corresponds to the convolution and pooling of different. The output of this layer can be represented as:

$$V_{CNN} = [V_{Pool}^1, V_{Pool}^2, \cdots, V_{Pool}^n] \tag{5}$$

2) **Capsule Layer:** The output vectors from the convolutional layer is first stacked into a matrix, and the matrix is reshaped to fit the input size of the capsule layer, as shown in Figure 1.

The matrix is then fed to a capsule layer with dynamic routing, to capture deep features of sentence representation. The key features of the sentences are transferred to the high-level capsule as the output:

$$V_{Capsule} = Capsule(M) \tag{6}$$

where M is the input matrix, and $Capsule$ is the capsule layer.

3) **Classifier:** The vector output from the capsule layer is flattened and then fed to the fully connected layer, which maps the sentence sequence from the high-dimensional feature space to the low-dimensional category label space.

$$logit = FNN\left(Flatten(V_{Capsule})\right) \tag{7}$$

$$label = argmax(logit) \tag{8}$$

The softmax activation function is used to convert the logit into the class probability. Below, $probs$ is a vector where each element represents the class probability of an author.

$$probs = Softmax(logit) \tag{9}$$

The architectural representation of the proposed model is shown in Figure 2.

4.2. Integrating BERTweet_Capsule with UWS

Previous work has shown that style features have a positive effect on model performance [5][7][19]. Thus, we incorporate user writing styles (UWS module in [19]) into our model, as an improvement.

In the UWS module, a two-step approach is applied for learning user writing styles. First is to utilize multi-view representations of a user's post and train these features by neural networks. The second step is to combine the cross entropy and triplet loss [6] objective functions to learn user writing styles for AA. There are five kinds of embeddings of multi-view representations, which are word, character, bigram, POS and feature embeddings. The input of the POS and feature embeddings are sequences of POS tags and lexical stylometric feature tags. A CNN is used as the final classifier for AA [19]. Similarly to the classifier part of BERTweet_Capsule module, we use the softmax activation function to obtain the class probability for each author.

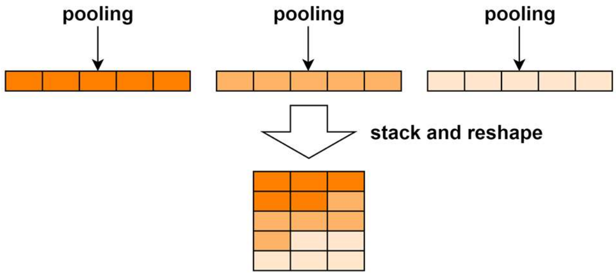

Figure 1: The input matrix of the capsule layer.

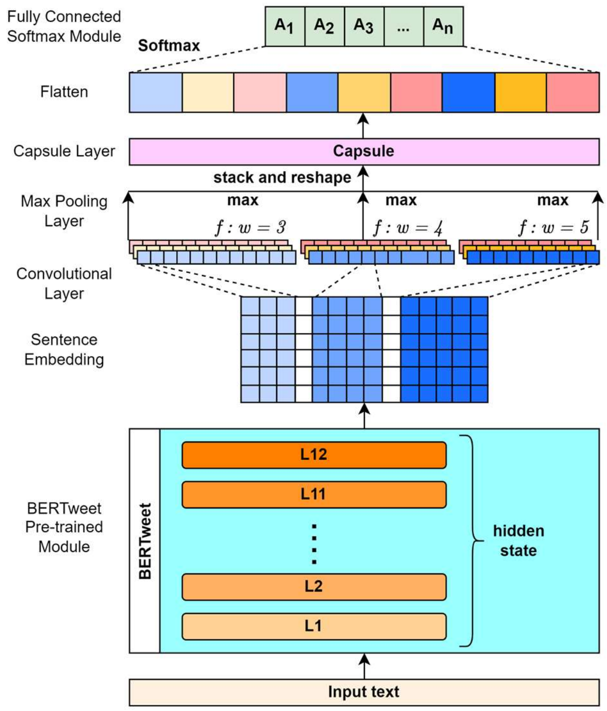

Figure 2: Proposed BERTweet_Capsule architecture.

To combine the BERTweet_Capsule module and the UWS model module, the output probabilities of two modules are concatenated into a vector. Then the vectors are fed to an MLP which contains only one hidden layer, for classification. By integrating the UWS module, our model can extract features of tweets from stylometric perspective and thus achieve better performance. Figure 3 shows the architecture of the overall system.

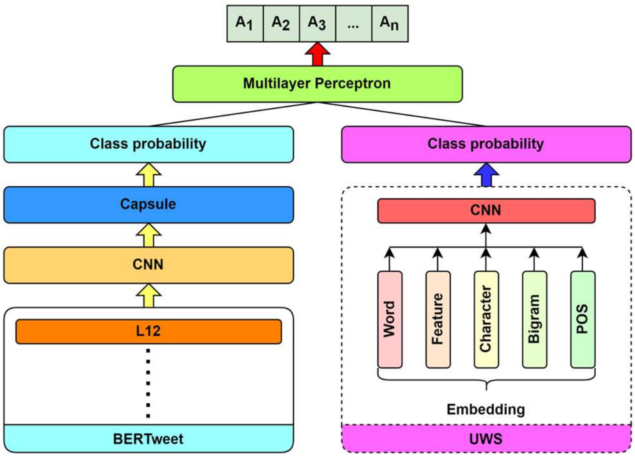

Figure 3: Proposed model with user writing styles.

5. Experiments

In this section, we perform experiments on several models in order to evaluate the performance of our proposed model. The benchmark dataset and baselines used in experiments, experimental details, the results of experiments, and ablation studies are described.

5.1. Dataset

We use the dataset released by Schwartz et al. [14] in all experiments, which is used by other previous approaches [7][11][18][19] too. The dataset contains groups of about 9,000 Twitter users, and each user has 1000 tweets, thus approximately 9 million tweets in total.

The proposed BERTweet_Capsule model uses the original text which is not preprocessed, while the UWS model uses the preprocessed tweets. Non-English tweets, tweets with less than three words and retweets have already been removed from the dataset. Mentions, URLs, numbers, dates and time in the tweets are replaced by tags 'M', 'U', 'N', 'D' and 'T' respectively and all letters in tweets are converted to lowercase for further preprocessing. The tweets with and without preprocessing are shown in Table 1.

user_id	original tweets	preprocessed tweets
746323	@MHDMunim Good afternoon to you.	Mmhdmunim good afternoon to you.
746323	Heading to SXSW 2010? Please vote for my "Get Yourself Lucky" talk - http://bit.ly/SXSWLucky	heading to sxsw D? please vote for my "get yourself lucky" talk - U
746323	@podcaststeve I6 was a local channel for me. 19 was for truckers.	Mpodcaststeve iN was a local channel for me. N was for truckers.

Table 1: Examples of original tweets and preprocessed tweets

5.2. Methods Used in Experiments

All the methods and descriptions used for our experimental evaluations are listed in Table 2.

Baselines. We use a logistic regression classifier over TF-IDF scores of words as a baseline. We also refer to the experimental results of [19], which combines RoBERTa finetuning and CNN models with user writing styles (UWS), for comparisons.

Our Models. We use the pre-trained language model BERTweet with the capsule network. We further incorporate UWS as an improvement and evaluate its effectiveness over our models.

To show the contribution of BERTweet and Capsule specifically, we remove some parts of the model and observe changes in accuracy. Since the contribution of CNN and the pre-trained language model has been demonstrated by a number of previous works, we do not remove CNN and pre-trained language model parts in the experiment. When removing BERTweet to observe its contribution, we use RoBERTa as the pre-trained language model instead.

Methods		Descriptions
Baselines	TF-IDF+LR	Traditional machine learning-based, calculating TF-IDF scores, then train a logistic regression (LR) classifier.
	RoBERTa+CNN (RoBERTa_CNN)	Extracting multiple hidden layers from RoBERTa and use different vector combinations as input features into CNN for classification task.
	User Writing Style (UWS)	An embedding-based framework that learns the representations of users' writing styles.
	RoBERTa_CNN+UWS	Combinations of user writing styles (UWS) and RoBERTa_CNN.
Proposed Models	RoBERTa_CNN+Capsule	Add Capsule Layer into RoBERTa_CNN.
	BERTweet+CNN	Remove Capsule Layer from BERTweet_Capsule.
	BERTweet+CNN+Capsule (BERTweet_Capsule)	Using last hidden state from BERTweet as input of capsule-based CNN.
	BERTweet_Capsule+UWS	Combinations of user writing styles (UWS) and BERTweet_Capsule.

Table 2: Overall methods and their descriptions.

5.3. Experimental Details

In the stage of model construction, we experimentally choose the best combination of the hyperparameter values for each part of BERTweet_Capsule model, on the train set. The BERTweet_Capsule model adopts the BERTweet base model, which has the same architecture as the BERT base model. A dropout layer is added before the fully connected layer of BERTweet_Capsule model to prevent overfitting. The details of the hyperparameters combinations for each part of BERTweet_Capsule are shown in Table 3.

Layer	Hyperparameters	
BERTweet	Epoch	45
	Batch size	64
	Max token size	64
	Embedding size	768
Embeddings	Length	128
	Dimension	768
CNN	Filter size	[3,4,5]
	Filter number	[128,128,128]
	Pooling	Global max
Capsule	Primary units number	32
	Primary unit size	12
	Output unit size	3
	Output units number	128
	Iterations	3
Dropout	Dropout rate	0.1

Table 3: Hyperparameter details on BERTweet_Capsule.

We find in experiments that when the capsule layer is added to the model, the convergence is much slower than that without the capsule layer. Figure 4 shows the validation losses in both cases for the training of 10 epochs. Further experiments show that the validation loss of the BERTweet_Capsule model becomes stable when training around 45 epochs, as shown in Figure 5. Thus, we set the number of epochs to 45.

The model which integrates user writing styles learns the post embeddings using Multi-Style Post Representation by CNN. We use the same hyperparameter settings as [19] for training user writing styles, see [19] for more details. An MLP model with one hidden layer is used to combine two modules. We set batch size to 64 to train the MLP model, and early stopping is applied in the training process to prevent the model from overfitting.

5.4. Basic Results

In experiments, we choose accuracy as the evaluation metric, which is widely used in classification tasks to evaluate the classification performance. The accuracy can be defined as the frequency to which predictions equal labels.

50 authors having 1000 tweets are selected randomly from the dataset which is already discussed in the previous section, for a total of $50 \times 1000 = 50000$ tweets. The 50000 tweets are divided into train, test and validation sets by 8 : 1 : 1, and the models are evaluated on the test set with 5000 tweets. The basic experimental results are shown in Table 4.

$$Accuracy = \frac{\#\ of\ samples\ predicted\ correctly}{\#\ of\ samples} \tag{10}$$

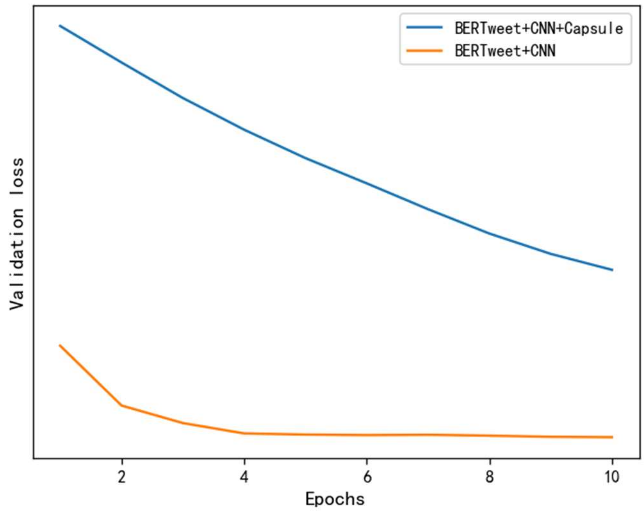

Figure 4: Influence of capsule layer on the model.

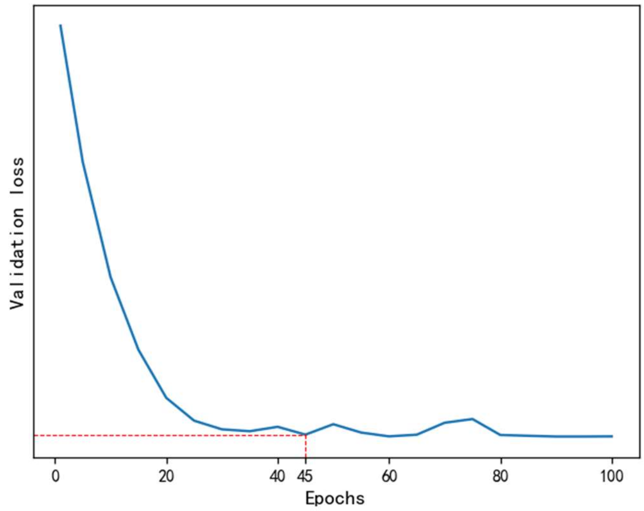

Figure 5: Best number of epochs for BERTweet_Capsule.

We can see from the experimental results that the traditional machine learning method (TF-IDF+LR) can achieve a good accuracy of 0.674. However, the deep learning methods are significantly more powerful than the machine learning method, the CNN-based architecture with pre-trained language model greatly improves the performance. Our proposed model has two core improvements: The first is adopting the domain-specific pre-trained language model to obtain the sentence representations. The second is using the capsule layer with dynamic routing to capture deep features of sentence representations. The proposed BERTweet_Capsule model shows significant improvement over the baselines, with an accuracy of 0.930, which is 4.5% higher than the existing best record. The UWS model is also having a positive impact on BERTweet_Capsule model, BERTweet_Capsule+UWS model achieves the state-of-the-art performance with an accuracy of 0.936, which increases by 0.6% compared to using BERTweet_Capsule model only.

Methods		Accuracy
Baselines	TF-IDF+LR	0.674
	RoBERTa+CNN (RoBERTa_CNN)	0.879
	User Writing Style (UWS)	0.848
	RoBERTa_CNN+UWS	0.885
Proposed Models	RoBERTa_CNN+Capsule	0.904
	BERTweet+CNN	0.913
	BERTweet+CNN+Capsule (BERTweet_Capsule)	0.930
	BERTweet_Capsule+UWS	**0.936**

Table 4: Accuracy for 50 authors with 1000 tweets each.

The experimental results also show that both BERTweet (compared to RoBERTa) and capsule layer positively contribute to our proposed model, as all the accuracies after removing one factor are below the accuracy of BERTweet_Capsule. Since replacing BERTweet by RoBERTa shows the largest impact on the accuracy, which decrease by 2.6%, we can conclude that BERTweet has greater contribution than the capsule layer to our proposed model.

6. Conclusion and Future Work

In this paper, we develop a capsule-based architecture over pre-trained language model BERTweet for authorship attribution on tweets. We further incorporate UWS as an improvement. The concatenations of class probabilities from both modules are used as the input of an MLP for authorship attribution. Our best method achieves an accuracy of 0.936 on a publicly available AA dataset, which is 5.1% improvement over the state-of-the-art result.

For future work, we will evaluate the models in case of the increasing difficulty level of the task, i.e., with decreasing number of tweets per author and increasing number of candidate authors. We would also like to explore more sophisticated architecture of the Capsule Layer, to make our proposed model more efficient and more robust.

References

[1] A. Abbasi and H. Chen, “Writeprints: A stylometric approach to identity level identification and similarity detection in cyberspace,” ACM Transactions on Information Systems (TOIS), 26(2):7, pp. 17 29, 2008.

[2] D. Bagnall, “Author identification using multi--headed recurrent neural network,” In Working Notes Papers of the CLEF 2015 Evaluation Labs, Vol. 1391.

[3] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, “Person Re-identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1335-1344, 2016.

[4] S. H. H. Ding, B. C. M. Fung, F. Iqbal, and W. K. Cheung, "Learning Stylometric Representations for Authorship Analysis," in IEEE Transactions on Cybernetics, vol. 49, no. 1, pp. 107-121, Jan. 2019.

[5] M. Fabien, E. ú Villatoro-Tello, P. Motlicek, and S. Parida, “BertAA: Bert fine-tuning for authorship attribution,” in Proc. ICON, pp. 1–11, Dec, 2020.

[6] Z. Hu, R. K. W. Lee, L. Wang, E. P. Lim, and B. Dai, “DeepStyle: User Style Embedding for Authorship Attribution of Short Texts,” in Proc. Asia-Pacific Web (APWeb) Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data, Springer, pp. 221-229, Aug. 2020.

[7] W. Huang, R. Su, and M. Iwaihara, “Contribution of improved character embedding and latent posting styles to authorship attribution of short texts,” in Proc. Asia-Pacific Web (APWeb) Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data, Springer, pp. 261–269, Aug. 2020.

[8] M. Koppel, J. Schler, and S. Argamon, “Authorship attribution in the wild,” Language Resources and Evaluation, 45(1), pp. 83 94, 2011.

[9] H. Liu, Y. Liu, L. Wong, L. Lee, and T. Hao, “A Hybrid Neural Network BERT-Cap Based on Pre-Trained Language Model and Capsule Network for User Intent Classification,” Complexity, 2020.

[10] D. Q. Nguyen, T. Vu, and A. T. Nguyen, “BERTweet: A pre-trained language model for English Tweets,” Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 9-14, Oct, 2020.

[11] L. Patamawadee and M. Iwaihara, “Utilizing Latent Posting Style for Authorship Attribution on Short

Texts," IEEE CBDCom 2019, Fukuoka, pp.1015-1022, Aug. 2019.

[12] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 3859-3869, Dec, 2017.

[13] T. Saha, S. Ramesh Jayashree, S. Saha, and P. Bhattacharyya, "BERT-Caps: A Transformer-Based Capsule Network for Tweet Act Classification," in IEEE Transactions on Computational Social Systems, vol. 7, no. 5, pp. 1168-1179, Oct. 2020.

[14] R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel, "Authorship Attribution of Micro Messages," Proc. 2013 Conf. Empirical Methods in Natural Language Processing, Seattle, pp. 1880-1891, Oct. 2013.

[15] P. Shrestha, S. Sierra, F. A. Gonzalez, P. Posso, M. Montes y Gomex, and T. Solorio, "Convolutional Neural Networks for Authorship Attribution of Short Texts," Proc. 15th Conf. European Chapter of the Assoc. Computational Linguistics, Vol. 2, Valencia, pp. 669-674, Apr. 2017.

[16] T. Solorio, S. Pillay, S. Raghavan, and M. Montes Gomez, "Modality specific meta features for authorship attribution in web forum posts," Proc. 5th Conf. International Joint on Natural Language Processing, pp. 156 164, Nov. 2011.

[17] E. Stamatatos, "A survey of modern authorship attribution methods," Journal of the American Society for information Science and Technology, 60.3, pp. 538-556, 2009.

[18] C. Suman, A. Raj, S. Saha and P. Bhattacharyya, "Authorship Attribution of Microtext Using Capsule Networks," in IEEE Transactions on Computational Social Systems, 2021.

[19] X. Wang, and M. Iwaihara, "Integrating RoBERTa Fine-Tuning and User Writing Styles for Authorship Attribution of Short Texts," in Proc. Asia-Pacific Web (APWeb) Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data, Springer, pp. 413-421, Aug. 2021.