Bilingual Tweets Authorship Attribution

Anonymous TACL submission

Abstract

This document contains the formatting requirements for TACL submissions. These formatting rules take effect for all submissions received from September 2, 2018 onwards.

1 Introduction

In recent years, political propaganda has moved a significant amount of resources onto the social media in additional to the tranditional mass media, which has created both positive and negative effects in the crowd. Since then, there has been many researches on applying NLP techniques to counter-attack the manufactured information published on the social media. Most of them analyzed stylometric features on mono-language data and showed promising potential in this field such as Rocha et al. (2016).

In June 2019, the proposed anti-extradition law in Hong Kong had attracted great controversy on the social media. Political propoganda also had played a big role in this movement. In fact, it was so severe that Twitter had to suspend about 1000 twitter accounts violating their platform manipulation policies[1]. A brief study by Wood et al. (2019) had revealed that the languages used in these tweets contents spaned across several languages (but mostly in Chinese and English), which proposed challenges for cross-langauge authorship attribution on short social media texts.

Unlike the authorship attribution in English social network texts, cross-language authorship attribution has not been discovered extensively yet. However as propoganda now reaching out more people around the world by using multiple language, there is the need to develop authorship attribution techniques for cross-language social media texts. In this project, I have decided to try the possibilities for bilingual authorship attribution — focuing on English and Chinese, by applying both machine translation and aligned word embeddings.

[1] https://blog.twitter.com/en_us/topics/company/2019/information_operations_directed_at_Hong_Kong.html

Chinese	English
是	am, is, are
的	of
有	have, has
在	at, in, on

Table 1: Function words used in the query string

2 Related Works

3 Methodology

3.1 Dataset

To my best knowledge there is currently no publicly available corpus focused in social media texts in both English and Chinese, hence I have decided to build my own using Twitter Public API.

The first step is to build a query string to collect as many tweets in either English or Chinese as possible. I selected several frequent function words in Chinese and their English counter parts, as shown in Table 1. Let's say $C = \{c_1, c_2, c_3, \cdots, c_i\}$ is a group of Chinese function words and $E = \{e_1, e_2, e_3, \cdots, e_j\}$ is their English siblings, the query string Q would be union set over the Cartesian product $C \times E$.

$$Q = \bigcup_{k=1} s_k, s_k \in C \times E \quad (1)$$

For example if we select the first row in the table, then our Chinese cadidate function word

# of tweets in total	69710
ZH tweets	27476
EN tweets	37604
Other lang. tweets	4630

Table 2: Overall look at the Dataset

is "是" while the English candidates are "am", "is" and "are". Our query string would be `(是 am) OR (是 is) OR (是 are)`.

The next step is to fine grain all of the possible twitter users from the previous step. Here I define two bilingual ratio values — let L_1 and L_2 denote two sets of the tweets in any of the two language written by a user, T as the set of all of his/her tweets, then we have R_{inner} as the ratio between these two tweets languages.

$$R_{inner} = \frac{\min(|L_1|, |L_2|)}{\max(|L_1|, |L_2|)} \quad (2)$$

And $R_{overall}$ as the ratio between the bilingual content and the total tweets.

$$R_{overall} = \frac{|L_1| + |L_2|}{|\mathrm{T}|} \quad (3)$$

For each user, I crawled his or her first 200 tweets from his timeline (exclude retweets) and set the threshold at $R_{inner} \geq 0.5$ and $R_{overall} \geq 0.8$. In this way I could filter out two kinds of users — someone who occasionally tweets in another language than his/her main language, and someone who tweets in all kinds of languages.

After these two steps I have selected 52 valid bilingual twitter accounts. Two of them are obviously non-personnal users so I have removed them from the list, which made the length of the final list of bilingual twitter users to 50. From them I crawled all of their tweets and applied cleaning to these data (including removing URLs, hashtags, mentions, reserved words, emojis and smileys). The CNN classifier cannot take tweets that are shorter then 5 words so I have also segmented both English and Chinese tweets and removed those that are not longer than 5 segments. In the end, I have obtained 69710 tweets that are detailed in Table 2 and 3. Also, each model is evaluated by 10-fold cross validation, except for the aligned word embeddings model which was trained and validated in fixed training and validataion dataset derived from the full dataset by a ratio of 0.6, 0.1 and 0.3.

3.2 Vanilla Model

As Rocha et al. (2016) had showned in their work, the best stylometric features for authorship attribution on social media text are word-level and char-level n-grams. I have designed a vanilla attribution model with word-level 1, 2, 3-grams and char-level 1, 2, 3-grams, which is then feeded into a logistic regression classifier and a LIBLINEAR(Fan et al., 2008) SVM classifier implemented by Scikit-learn (Pedregosa et al., 2011). The word-level and char-level features are calculated from all available tweets without distinguishing the language. I have also transformed all of the appearance counts to TF-IDF values to deminish the effect of trending words.

We have seen from Table 3 that the number of tweets from each user is highly unbalanced. The most frequent user tweets 40 times more than the most quiet user. So I have set both of the logistic regression and LIBLINEAR SVM to balance out the dataset automatically by assigning more weight to minor classes (users).

3.3 Machine Translation Model

Inspired by Bogdanova and Lazaridou (2014) I have adapted to tackle the cross-langauge authorship attribution problem is by using machine translation. I have used the Translator Text API from Microsoft Azure Cognitive Services, manully specifying the source language and the target langauge. The state-of-art machine translation service is far from perfect and underperforms on social media text than formal writings. In order words it will inevitably introduce "distortion" to the raw tweet and worsen the result in theory, I still argue that machine translation is one of the cheapest and the most intuitive solution to multi-language tasks in NLP.

In this second authorship attribution model, I have extracted and divided the raw tweets into the English group and the Chinese group. For tweets that mix both languages I seperated them into these two monolingual groups. Then tweets in each language group will be machine translated into the other language before being feeded into the aforementioned logistic regression and SVM classifier, together with other translated tweets within the same group.

	# of tweets/user	Length of raw tweets	Length of ZH tweets	length of EN tweets
mean	1383.4	100.309	37.664	72.59
std	1211.126	38.24	29.142	31.04
min	102	5	5	6
25%	334.5	68	16	47
50%	856	109	27	74
74%	2306.75	140	51	100
max	4195	159	140	146

Table 3: Distribution of the Dataset

3.4 Aligned Word Embeddings Model

The last model I have applied is a Convolutional Neural Network classifier inspired by Shrestha et al. (2017). They have proposed an architecture using char n-gram models as the single embedding layer to deal with tweets classification. But since my task is to explore the classification problem between two languages, I have switched to the aligned fastText word embeddings (Bojanowski et al., 2017) (Joulin et al., 2018) as my embedding layer.

In the aligned fastText word embeddings, each word is represented by a 300 dimension vector. I concatenated two embeddings to form a bunch of 600 dimension word embeddings for the possible bilingual vocabularies, padded them and send them to the next layer. Each OOV words are marked as `<UNK>` and are giving an embedding of zeros.

The CNN network also has four convolutional layers, each of them has 100 kernels sizing from 2, 3, 4 or 5. They are designed to catch the information hidden inside the word bigram. trigram and quadgrams before being max-pooled. I have adapted Adam optimizer (Kingma and Ba, 2014) and all of the hyperparams are shown in Table 4.

Hyperparameters	Value
# of embedding layers	1
dimension	600
# of convolutional layers	4
kernel size	[2, 3, 4, 5]
# of kernels	100/layer
pooling	max
Dropout	0.5
Learning rate	0.001
Max epochs	10
Batch size	32

Table 4: Hyperparameter settings in the Aligned Word Embeddings model

4 Results

4.1 Best Features

For the first two models I have applied grid search to find the best stylometric feature for biligual authorship attribution. As shown in Table 5, the best feature are usually the shorter word unigrams or bigrams for each classifier. Only in the translated Chinese group the best results appeared in the char-level bigrams and trigrams. But this can be explained by the fact that in Chinese the average word length is about one to two characters while in English it is about four to five letters. (Chen et al., 2015) (Bochkarev et al., 2015). In other words, Chinese characters them alone can carry as much information as English words. The result that word-level n-grams are more effective than char-level is inline with results from other previous work in many other authorship attribution tasks (Kestemont et al., 2018) (Rangel et al., 2019).

Also the SVM classifier outperformed the logistic regression classifier in nearly all comparisions, except in the char-level unigram one. This can be seen in Rocha et al. (2016)'s work as well as they attributed it to the application of Maxiumn Margin Princple in SVM classifiers. However for authorship attributuon task on long articles, logistic regression could be better than SVM (Bogdanova and Lazaridou, 2014). Thus I think SVM is better suited for short social media texts than logistic regression.

4.2 Distortion from Machine Translation

As mentioned previously, machine translation will inevitably introduction noise into the text and will bring down the accuracy. In Table 5 I have also calculated the impact of machine translation com-

	LR	SVM	LR+MT(EN)	SVM+MT(EN)	LR+MT(ZH)	SVM+MT(ZH)
Word 1-gram	0.659	0.714	0.533 (-19.1%)	0.553 (-22.5%)	0.614 (-6.8%)	0.642 (-10.1%)
2-gram	0.648	0.744	0.543 (-16.2%)	0.602 (-19.1%)	0.612 (-5.6%)	0.68 (-8.6%)
3-gram	0.62	0.733	0.523 (-15.6%)	0.599 (-18.3%)	0.595 (-4%)	0.673 (-8.2%)
Char 1-gram	0.452	0.45	0.216 (-52.2%)	0.197 (-56.2%)	0.575 (+27.2%)	0.583 (+29.6%)
2-gram	0.592	0.655	0.415 (-29.9%)	0.433 (-33.9%)	0.622 (+5.1%)	0.68 (+3.8%)
3-gram	0.637	0.723	0.5 (-21.5%)	0.555 (-23.2%)	0.615 (-3.5%)	0.682 (-5.7%)

Table 5: Results for the Vanilla Model and the Machine Translation Model

Word Embeddings	Accuracy	Loss
Aligned Bilingual	70.06%	1.197
ZH Only	68%	1.242
EN Only	68.99%	1.195
Unaligned Bilingual	64.76%	1.816

Table 6: Results for the Aligned Word Embeddings Model

pared to the untranslated original text. Word-level bigrams has topped the chart in almost every group, followed closely by word-level trigrams and unigrams, which once again showed that word-level n-grams are better features than char-level n-grams in our bilingual tweet dataset. Further more, LIBLINEAR SVM classifier still achieved higher accuracy than logistic regression classifier after machine translation, showed that it is more suitable for short social media text no matter what language the text is in.

Move on to the performance difference between translated Chinese and English texts, we can see that Chinese suffered more than English if it been translated. Especially in the case of char 1-gram, when all Chinese text are translated into English the performance dropped more than 50%. In contrast while we turn all English content into Chinese, we have manageed to improve the performance by nearly 30%. Since many Chinese characters can serve as a word alone, short char-level n-grams in Chinese can be viewed as word-level n-grams in English. Thus explains why we had a large gain on performance when we translate everything into Chinese.

4.3 Aligned Word Embeddings

Finally the results for the aligned word embeddings model are shown in Table 6. Because fastText only provides word embeddings for Tranditional Chinese I have used OpenCC[2] to convert all the Simplified Chinese content to Tranditional Chinese. Furthermore I have also added a reference group using the unaligned common fastText embeddings. Both the aligned and the unaligned word embeddings for Chinese and English were pre-trained on Wikipedia text (Bojanowski et al., 2017).

The aligned word embeddings model didn't surpass my vanilla model, however it is much closer to it than the machine translation model. Also the performance gain by using bilingual embeddings was subtle compared to monolingual embeddings, only around 2%. The unaligned model performed worst among these three kinds of embeddings, which is expected.

The result for the aligned word embeddings model might suggest a better combination for bilingual word embeddings rather than concatenation. But there are other facts that could also affect the final result. First, the fastText word embeddings were trained on a domian that is far away from social media text. Wikipedia is more formal, serious and comprehensive place than Twitter, and the topics it includes had little intersection with the topic from Tiwtter. The results could be improved by training a dedicated word embeddings from Twitter corpus. Another things to note is the imbalance between English and Chinese word embedding sizes. The English embeddings has 2519370 items and is 7.5 times larger than the Chinese embeddings, just as the English Wikipedia articles are around 5 times more than articles in Chinese [3] [4]. Smaller embeddings size will introduce more OOV words and will lower the overall accuracy.

[2] https://github.com/BYVoid/OpenCC

[3] As Dec 2018, https://stats.wikimedia.org/EN/TablesWikipediaEN.htm

[4] As Dec 2018, https://stats.wikimedia.org/EN/TablesWikipediaZH.htm

5 Conclusion

References

Vladimir V Bochkarev, Anna V Shevlyakova, and Valery D Solovyev. 2015. The average word length dynamics as an indicator of cultural changes in society. *Social Evolution & History*, 14(2):153–175.

Dasha Bogdanova and Angeliki Lazaridou. 2014. Cross-language authorship attribution. In *LREC*, pages 2015–2020. Citeseer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Heng Chen, Junying Liang, and Haitao Liu. 2015. How does word length evolve in written chinese? *PloS one*, 10(9):e0138567.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Mike Kestemont, Michael Tschuggnall, Efstathios Stamatatos, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. 2018. Overview of the author identification task at pan-2018: cross-domain authorship attribution and style change detection. In *Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al.*, pages 1–25.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Francisco Rangel, Paolo Rosso, L Cappellato, N Ferro, H Müller, and D Losada. 2019. Overview of the 7th author profiling task at pan 2019: Bots and gender profiling. In *CLEF*.

Anderson Rocha, Walter J Scheirer, Christopher W Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne RB Carvalho, and Efstathios Stamatatos. 2016. Authorship attribution for social media forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):5–33.

Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674.

Daniel Wood, Sean McMinn, and Emily Feng. 2019. China used twitter to disrupt hong kong protests, but efforts began years earlier.