Region-Based SMS Stylometry Using Recurrent Neural Networks

Arlie Coles and **Lino Toran Jenner**
McGill University
arlie.coles@mail.mcgill.ca and lino.toranjenner@mail.mcgill.ca

Abstract

Stylometry, or authorship detection, has been approached in several ways, including the use of n-gram representations and recurrent neural networks (RNNs) to predict the author of a document, particularly for documents of large size. In this paper we attempt to broaden the stylometry task and detect the dialectical region of the English used by the author of short SMS messages. We design and implement two RNNs for this task which integrate regional word usage information from the Google Trends database in two different ways. Comparing their performances against a number of baselines, we find that RNN approaches outperform other methods, although the usefulness of the inclusion of Google Trends data is uncertain outside one particular case.

1 Introduction

Stylometry is the task of learning particularities and patterns in the use of produced language that reflect some identifying characteristic of the author. Often this learned information is used for explicit identification of an individual author, but sometimes it is used to predict a category or broader feature of the author, such as age, gender, or native language. In this paper, we propose two Recurrent Neural Network (RNN) methods for predicting the dialectical region of the English used by the author of SMS messages. Since the short length of SMS data can pose difficulty, our methods attempt to make use of RNNs' natural ability to learn a wide variety of features (including parts of speech and syntactical features as well as n-grams) as well as integrate in an outside source of potentially useful data to serve as a "prior": region-based search information from Google Trends.

2 Related Work

Traditionally, stylometry research has focused on identifying specific authors of longer texts, such as works of fiction or political documents (Matthews and Merriam, 1993; Tweedie et al., 1996). However, given the revolutionary nature of modern social media, there is today both the data and the motive, including forensic applications and demographic research, to perform stylometry on smaller forms of communication. In recent years stylometry techniques have been applied to Twitter posts, emails, and text (SMS) messages (Layton et al., 2010; Brocardo et al., 2013; Ragel et al., 2014). Because this kind of data's short length and other characteristics including use of internet slang and other typing idiosyncrasies (e.g. is use of *lol* vs. *lool* a style choice indicative of an individual author, or just a typo?), traditional stylometric approaches often do not lead to satisfying results in this domain. Instead, novel feature selection tools, (e.g. only looking at character frequencies instead of words) or more advanced neural networks are used.

Brocardo et al. use a character n-gram model to identify the senders of e-mails, an approach that was imitated on the word level by Ragel et al. to identify individual senders of SMS messages, these studies achieving a 14.35% Equal Error Rate and an accuracy rate of above 80% respectively (2013; 2014). While encouraged by their results, these authors note that n-grams are but one of the many features potentially useful for stylometry. Deitrick et al. used a variety of features including combinations of word- and character-level n-grams as inputs to a Modified Balance Winnow neural network, achieving 98.51% accuracy on the broader task of gender detection on Twitter messages (2012). Still others have attempted to leverage the sequence-modelling ability of Recurrent Neural Networks (RNNs) for stylometry of longer documents with varying degrees of success (Yao and Liu, 2015; Bagnall, 2015).

3 Proposed RNNs

While RNNs have been used for stylometry of long documents (Yao and Liu, 2015), we seek to apply them to short SMS messages and to compare their performance against previous non-RNN accuracy rates.

Additionally, we examine whether the inclusion of additional regional word usage information scraped from Google Trends is useful in the proposed classification task. Google Trends (GT) is a tool which takes an input query and returns relative Google search information for that query by location. We make use of this tool in order to bolster our network with information about what may be regional shibboleths: GT returns the United Kingdom as the top searcher for *pub*, and shows that a search for *y'all* is more than 7 times more likely to have come from the United States than from Singapore. As such, the information it returns may be helpful for the task of automatic regional detection.

3.1 The GT-Embedding RNN

Our first proposed model, the *GT-Embedding RNN*, takes as input the sequence of words of an SMS message and feeds them into an LSTM including an embedding layer (as described above). Instead of learning word embeddings from the ground up, as is typical, we use GT's smoothed probabilities as priors. This results in the learning of something like a "location embedding" for each word, hopefully boosted in salience by the Google Trends data. Figure 1 shows a diagram of this model.

3.2 The GT-Filter RNN

Our second proposed model, the *GT-Filter RNN*, comprises of an RNN and a *Filter*. The RNN takes as input the sequence of words of an SMS text message and feeds them into an LSTM including an embedding layer, where embeddings are initialized uniformly randomly.

The Filter makes use of GT regional probabilities on the message level (see Section 4.2 for calculation details) and is then applied to the output probability distribution of the network in the following way. The region with the lowest RNN probability is dropped from consideration. Then, if the remaining two RNN probabilities both surpass a threshold of X, here 30% (meaning they are very close to one another), and if the two corresponding GT message probabilities are not within Y, here 10% of each other (meaning GT is clearly favoring one of the two locations), then the location corresponding to the higher of the two GT message probabilities is chosen. Otherwise, the highest of the RNN probabilities is chosen. This way, the Filter gives a "tie-breaking" boost to regions which may be more likely for a given message according to GT when the RNN itself is having trouble deciding. These threshold values were tuned using the development set. Figure 2 shows a diagram of this model.

3.3 Network architecture and hyperparameters

We use the Keras library (Chollet and others, 2015) for the implementation of our models. Our proposed RNNs feature an embedding layer with varying embedding vector lengths: the *GT-Embedding RNN* uses vector length 3 (one dimension for each location class). Meanwhile, the *GT-Filter RNN* uses vector length 32, as we noticed during its development that results improved with a more complex representation of word embeddings.

This embedding layer is connected to a LSTM layer (tanh activation) with 100 units. We applied dropout regularization with 65% dropout to this layer. This layer is then connected to the output layer which consists of 3 units. The output of the network is a softmax-generated probability distribution over the three potential regions of origin of the SMS message. We use cross-entropy as a loss function and Adam as optimizer (Kingma and Ba, 2014).

We trained the models with vector embedding length 3 for 10 epochs and the model with length 32 for 3 epochs. These values were determined by

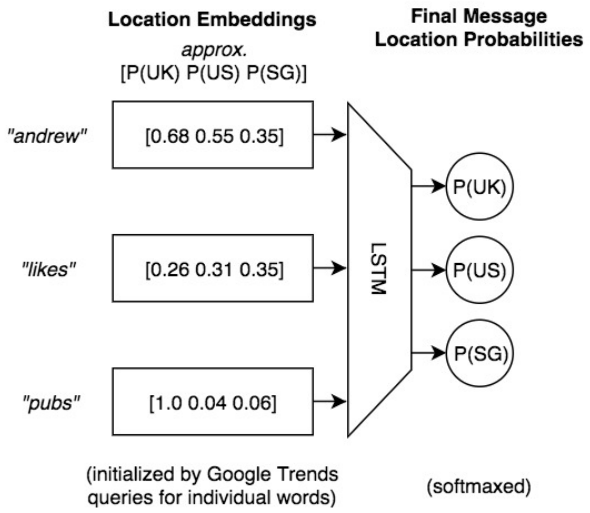

Figure 1: The GT-Embedding RNN.

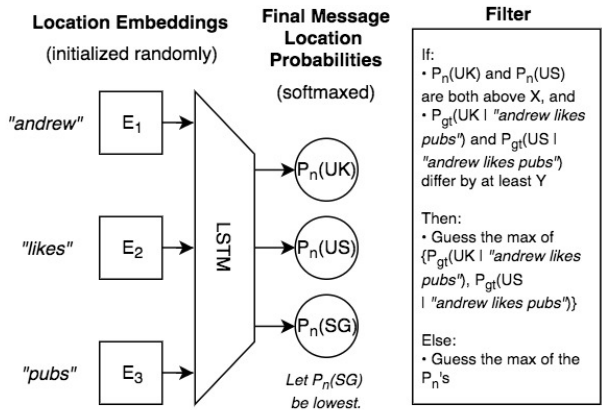

Figure 2: The GT-Filter RNN.

monitoring the development set accuracy.

4 Method

4.1 Datasets

In order to amass enough English-language SMS data for a meaningful experiment, we assemble what we have named the English SMS Corpus. This assembled corpus is a combination of English-language SMS messages from two sources: the NUS SMS Corpus and what we call the Tagg Corpus. The NUS SMS Corpus is a corpus of English-language SMS messages tagged with information (as available) including author name, age, gender, and location (Chen and Kan, 2013). The Tagg Corpus, scraped from the doctoral dissertation by the author of the same name, comprises a curated set of SMS messages from the United Kingdom (Tagg, 2009). After combining these two sources and grouping their messages by the three most common locations, we arrive at our English SMS Corpus which includes 38,098 messages in total (31,860 from Singapore, 5,748 from the United States, and 490 from the United Kingdom).

4.2 Experiments

To evaluate the performance of our RNNs, we compare their overall accuracies (the percent correct classifications of messages as originating from Singapore, the United States, or the United Kingdom) against the accuracy of a number of baselines. In cases where training and testing are both needed, we shuffle and split the English SMS Corpus into 80% training, 10% development, and 10% testing sets.

First, we consider a standard multinomial Naive Bayes three-way classifier (with alpha parameter of 1.3, chosen following testing on a development set) trained on word-level n-grams. This classifier allows us to compare our more sophisticated RNN approach with a machine-learning baseline. Next, we examine the performance of two RNNs that do not incorporate any GT information (the *Non-GT RNNs*). In one case, we use a embedding vector length of 3 and in the other 32. These Non-GT RNNs are implemented to check whether using the GT-Filter RNN leads to better results than an RNN without GT information. The RNN with higher embedding vector length is used to check whether more complex word embeddings lead to better classification results in general.

We also compare the performance of our RNNs to two static baselines that do not require training: the baseline of always guessing ”Singapore” (AGS), and a *Google Trends baseline*. Since the location classes in our dataset are not evenly distributed, guessing Singapore leads to results that are significantly better than random guessing, but still fulfills the function of an unsophisticated baseline here. The Google Trends baseline's prediction for a message is calculated by summing the GT-returned probabilities by location of every (non stop-) word in the message, then dividing by the length of the message,

Model	Percent Accuracy	
AGS baseline	83.56%	
Google Trends baseline	42.64%	
	Train	**Test**
Naive Bayes	88.73%	86.64%
Non-GT RNN (3 dim.)	94.47%	88.71%
Non-GT RNN (32 dim.)	92.55%	**90.16%**
GT-Embedding RNN	94.34%	89.87%
GT-Filter RNN	—	90.08%

Table 1: Accuracies for each baseline and model by testing condition (on training and testing sets, where applicable).

creating three "average" probabilities: the probability that the message originated in Singapore, in the United States, and in the United Kingdom. This baseline then selects the highest of these three "average" probabilities. The Google Trends baseline is used to determine the basic usefulness of the GT data.

Finally, we compare these baselines against the performances of our two RNNs themselves, the GT-Embedding RNN and the GT-Filter RNN, on the test set.

5 Results

Accuracies for each baseline and testing condition can be found in Table 1.

As can be seen, the AGS baseline already predicts 83.56% of the classes correctly due to to the uneven distribution of classes in our data. Therefore, any model yielding accuracies below that threshold is not useful. The results of the Google Trends baseline show that GT alone does not seem to be very accurate for classification for most of the text messages, and thus provides ample room for improvement. Since this baseline takes an average over all words in a message, the results become noisy, leading to ineffective classification alone. The Naive Bayes baseline achieves better accuracies on the test set than these two non-trained baselines, as do all of our RNN models. While the Google Trends baseline might be unimpressive, its component information might still be relevant in some cases. For instance, the GT-Embedding RNN which initialized its weights with the GT data performed better than the Non-GT RNN (with same embedding length of 3) that did not.

The strategy of using the GT data as a filter leads to improvements in accuracy on the development set. However, when applying this approach on the test set, this improvement was not repeated, instead yielding a slightly worse accuracy than the Non-GT RNN (of 32 dimensions) without filtering (90.08% vs. 90.16%).

6 Discussion and Conclusion

We described two approaches of including GT location-based search data into an RNN to provide prior information for a dialect classification task. While the first implementation that used this data as initial word embeddings, the GT-Embedding RNN, did lead to better classification accuracy compared to a model with the same word embedding length but without this data, the second implementation that used the GT information only in cases where the RNN was unsure, the GT-Filter RNN, did not. However, the GT-Embedding RNN's performance was outshone by the Non-GT RNN (32 dim.), which makes no use of GT data, but does use a larger embedding vector length. This suggests that while GT can be a helpful supplement under certain circumstances, the information they represent that is pertinent to this task can be learned by a complex enough RNN from the data alone.

Our approaches have several limitations. Most notably, there is a dearth of available SMS data in general, and our English SMS Corpus suffers from being heavily skewed toward Singaporean data. As such, any RNN that we train may very well be able to identify a Singaporean SMS, but may have difficulty classifying one as from the UK, the region with the fewest messages in our corpus. Since the GT-Filter RNN, uses GT data only when the RNN alone assigns similar probabilities to two regions (and is "unsure"), and since given the uneven distribution of the data it is unlikely the RNN will ever be "sure" about assigning "UK" to a message, the only-modest improvement of the GT-Filter RNN over the GT-Embedding RNN and its lack of ability to surpass the 32-dimensional Non-GT RNN is understandable.

Another limiting factor is the nature of the SMS data itself. As before mentioned, SMS-related idiosyncrasies can be hard to account for in preprocessing. A persistent habit we noticed was omis-

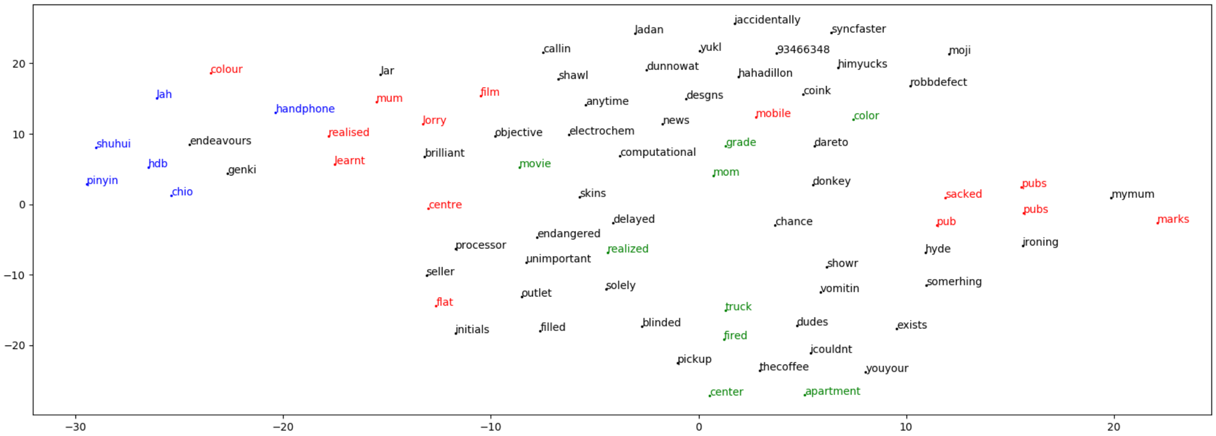

Figure 3: Learned embeddings from the GT-Embeddings RNN. Typically Singaporean words are marked in blue, American ones in green, and British ones in red.

sion of spaces between two or more words, which we kept as-is when querying GT under the impression that perhaps these conglomerates could be revelatory of location; however, they were not, as often nobody had searched for them from any location. It is thus more likely that inserting a space and querying on the component words could provide more informative results, which could be accomplished by employing a word-boundary parser on the SMS input.

Furthermore, the GT data itself is noisy in nature. Some words, while arguably used similarly often in all three locations, show a higher probability in a certain region (e.g. the word *food* achieves GT scores of 1.0 for Singapore, .67 for the US and .49 for the UK. Note also that these results do not form a probability distribution *per se*, adding to the difficulty.) This is one of the reasons why the Google Trends baseline does not perform competitively with the other models. Since we used the GT scores for all non-stopwords, this noisiness might have overshadowed potential usefulness. An attempt to counter this and improve results could be to only use the GT data for a subset of words for which this data is very accurate, if such a method to find this subset could be developed. We remark in addition that in the potential context of a real-time classification, GT may be able to provide an edge in that it has access to more up-to-date usage information than past corpus data; an appropriate subsetting method should therefore account for message age.

Finally, we note that while the GT-Embedding RNN did not provide top results, its learned embeddings, a subset of which are shown in Figure 3, do seem to cluster somewhat by location. This is a promising indicator that this task is an appropriate one for learning "location embeddings", and presumably these clusters would become denser given more data. Such embeddings might be useful for more robust dialect-identification tasks, especially if combined with data sources other than SMS messages.

In sum, a general RNN approach yields better results on our location identification task than non-neural network methods, but the efficacy of integrating GT data into these networks to this end requires further research.

7 Statement of Contributions

Generally, Coles was responsible for the assembly and parsing of the English SMS Corpus, the scraping of GT location data, and the conducting of the Google Trends and Naive Bayes baselines. Toran Jenner was responsible for the needed development research into Keras and the implementation and conducting of the Non-GT RNNs and AGS baselines. Coles and Toran Jenner together were responsible for the design and implementation of the GT RNNs including tuning of hyperparameters, for experimental design, for needed background research, and for the writing of this report. We hereby state that all the work presented in this report is that of the authors.

References

Douglas Bagnall. 2015. Author identification using multi-headed recurrent neural networks. *arXiv preprint arXiv:1506.04891*.

Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, and Isaac Woungang. 2013. Authorship verification for short messages using stylometry. In *Computer, Information and Telecommunication Systems (CITS), 2013 International Conference on*, pages 1–6. IEEE.

Tao Chen and Min-Yen Kan. 2013. Creating a live, public short message service corpus: the NUS SMS corpus. *Language Resources and Evaluation*, 47(2):299–335.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

William Deitrick, Zachary Miller, Benjamin Valyou, Brian Dickinson, Timothy Munson, and Wei Hu. 2012. Gender identification on Twitter using the modified balanced winnow. *Communications and network*, 4(03):189.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Robert Layton, Paul Watters, and Richard Dazeley. 2010. Authorship attribution for Twitter in 140 characters or less. 0:1–8, 07.

Robert AJ Matthews and Thomas VN Merriam. 1993. Neural computation in stylometry I: An application to the works of Shakespeare and Fletcher. *Literary and Linguistic Computing*, 8(4):203–209.

Roshan G. Ragel, P. Herath, and Upul Senanayake. 2014. Authorship detection of SMS messages using unigrams. *CoRR*, abs/1403.1314.

Caroline Tagg. 2009. *A corpus linguistics study of SMS text messaging*. Ph.D. thesis, University of Birmingham.

Fiona J Tweedie, Sameer Singh, and David I Holmes. 1996. Neural network applications in stylometry: The Federalist Papers. *Computers and the Humanities*, 30(1):1–10.

Leon Yao and Derrick Liu. 2015. Wallace: Author detection via recurrent neural networks.