Deanonymizing Quora Answers

Pranav Jindal
pranavj@stanford.edu

Paranjape, Ashwin
ashwinpp@stanford.edu

1 Introduction

Quora is a knowledge sharing website where users can ask/answer questions with the option of anonymity .We investigate the problem of **Author-identification** for Quora answers using deep learning techniques in Natural Language Processing.

1.1 Problem Statement

We hope to achieve significant precision on the task of identifying users from their writings with the end-goal of recognizing the authors of anonymous answers on Quora. Previous work indicates that writing style harbors essential cues about authors and we believe that deep learning is a powerful tool to extract such features to distinguish between the various writing styles that people have. The work finds applications to several other tasks like Forensic Linguistics, email spam detection, identity tracing in cyber forensics etc.

1.2 Background Reading

There has been fair amount of interest in author-identification in previous NLP works with most of the work focussing on manually engineered features:-

1. Comparing Frequency and Style-Based Features for Twitter Author Identification:

   Examines author identification in short texts focussing on messages retrieved from Twitter to determine the most effective feature set for recognizing authors look at Bag-of-Words and style-marker features and use SVMs for the classification task

2. A Comparative Study of Language Models for Book and Author Recognition:

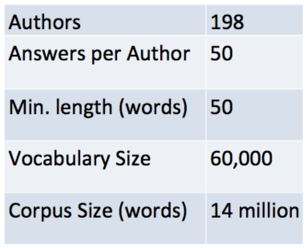
   Evaluates similarity between documents and authors showed that syntactic features are less successful than function words for author attribution

3. A Survey of Modern Authorship Attribution Methods

   Discusses how this scientific field has been developed substantially taking advantage of research advances in areas such as machine learning, information retrieval, and natural language processing over the past few decades

1.3 Dataset

To generate a small-version of the problem we deal with , we select 200 writers from the list of top Quora writers and use RSS feeds to generate a dataset containing exactly 50 answers per user.

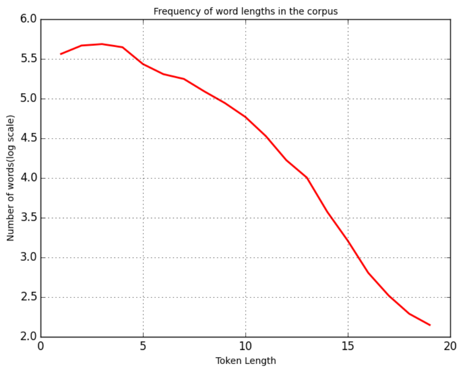


Authors	198
Answers per Author	50
Min. length (words)	50
Vocabulary Size	60,000
Corpus Size (words)	14 million

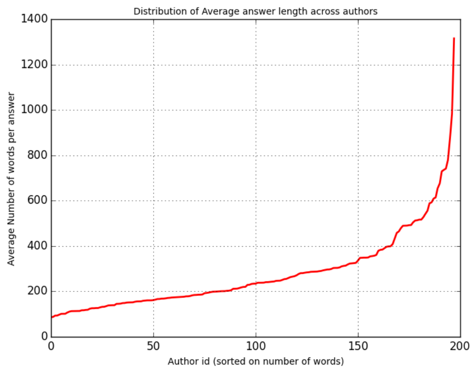
Figure 1: Dataset details

Quora Top writers: they are a group of people with recognized expertise, knowledge and authenticity. Some people in the group are experts in specific fields, while others are simply great writers with a talent for describing the human condition and the world around us.

Selecting these people ensures high quality content in the form of a large number of long answers per author.



(a) Word length vs. Frequency



(b) Average answer length distribution

Figure 2: Data statistics

We observe that most of the authors in the dataset have an average of less than 150 words per answer, though there are a few authors in the dataset which write very long answers. Also, not surprisingly, the word length vs frequency curve follows a power law.

2 Technical Approach and Models

The labels are hidden for a fraction of answers for every author to test our final model and the remaining dataset is used to train on the task of author attribution. We used machine learning models on engineered features as well as deep learning models for the task.

2.1 Model 1: Style marker features

The following commonly used style marker features from previous works in author-identification were used for the classification task

1. Number of words in the answer
2. Fraction of words that were punctuations
3. Average word length
4. Standard deviation of word length

5. Number of sentences in the answer
6. Average Sentence length
7. Number of digits in the answer

2.2 Model 2: Word Frequency model

Each answer is modeled by a feature vector of the length of the vocabulary set and contains the counts for each word in the vocabulary set for that answer
The vocabulary set is varied by incrementally adding more tokens in the order of decreasing frequency, using the complete vocabulary works best.

2.3 Model 3: LSTM with mean-pooling

The LSTM model is a recurrent neural network with memory units, allowing the cells to remember or forget its previous state, as needed.

Notation:
x_t is the input to the memory cell layer at time t
W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o and V_o are weight matrices
b_i, b_f, b_c and b_o are bias vectors

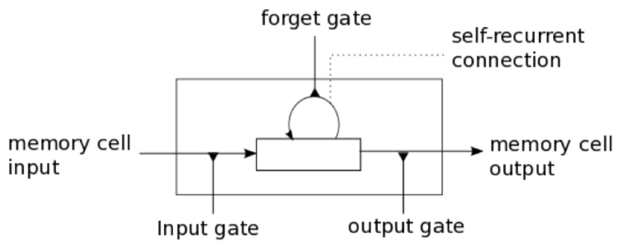


Figure 3: LSTM unit

Memory Unit Update:

First, we compute the values for i_t, the input gate, and $\widetilde{C_t}$ the candidate value for the states of the memory cells' at time t :

(1) $i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$
(2) $\widetilde{C_t} = tanh(W_c x_t + U_c h_{t-1} + b_c)$

Second, we compute the value for f_t, the activation of the memory cells forget gates at time t

(3) $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$

Given the value of the input gate activation i_t, the forget gate activation f_t and the candidate state value $\widetilde{C_t}$, we can compute C_t the memory cells' new state at time t :

(4) $C_t = i_t * \widetilde{C_t} + f_t * C_{t-1}$

With the new state of the memory cells, we can compute the value of their output gates and, subsequently, their outputs :

(5) $o_t = \sigma(W_o x_t + U_o h_{t-1} + b_1)$
(6) $h_t = o_t * tanh(C_t)$

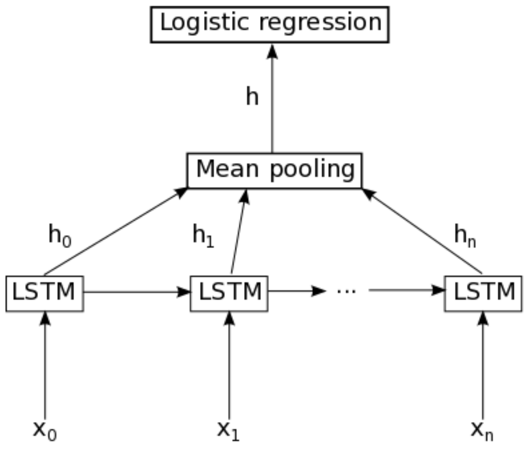


Figure 4: Mean Pooling with LSTMs

Mean pooling:
The model is composed of a single LSTM layer followed by an average pooling and a logistic regression layer as illustrated in Figure 4. Thus, from an input sequence $x_0, x_1, x_2, ..., x_n$, the memory cells in the LSTM layer will produce a representation sequence $h_0, h_1, h_2, ...h_n$.

This representation sequence is then averaged over all time-steps resulting in representation h. Finally, this representation is fed to a logistic regression layer whose target is the class label associated with the input sequence.

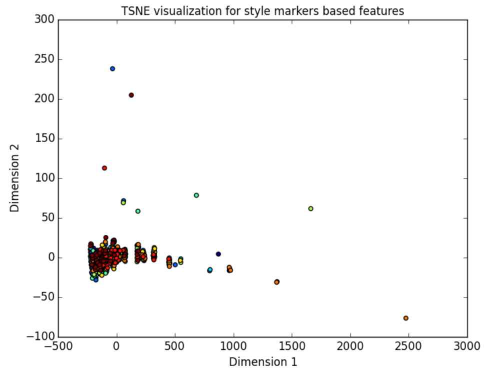
3 Results

The dataset for 200 authors with 50 answers per author was split into 80:10:10 for train, validation and test datasets respectively.

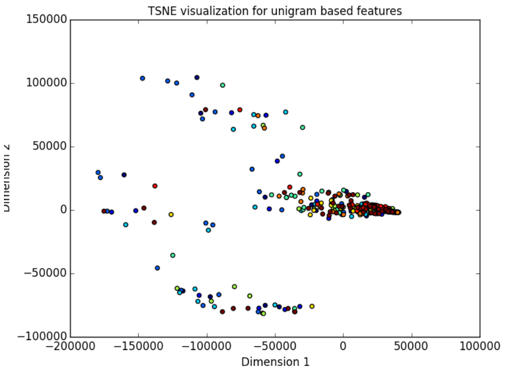
3.1 Visualization (TSNE)

TSNE is a tool to visualize high-dimensional data, converts similarities between data points to joint probabilities and tries to minimize the KL divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data

We visualize the 2 sets of baseline features using TSNE, displaying how well the features separate the various classes. Each scatter point represents an answer and the color represents the author for that answer



(a) TSNE Visualization - Style markers



(b) TSNE Visualization - Unigram Features

The Unigram features clearly do a much better job at separating the data as compared to the style markers, confirming the findings in [, 2]

3.2 Evaluation Metrics

We use top-k accuracy to evaluate the performance of our models, i.e. the author prediction is considered as correct if the true author belongs to the top-k predictions of the model for a given answer

Random Guessing Top-1 Accuracy = 0.505% (198 classes)

3.2.1 Traditional Machine Learning Methods

Various classifiers including random forests, Multinomial Naive Bayes, Adaboost, Gradient Boosting were tested on the feature sets , the best performance was achieved by using Random Forests, Multinomial Naive Bayes also achieved reasonable accuracy.

Dataset	Top-1 Accuracy	Top-5 Accuracy	Top-10 Accuracy
Training	45.51	74.14	84.70
Validation	6.71	17.8	27.12
Test	6.63	17.91	27.22

Table 1: Results for model 1 features for best model (Random Forests)

Dataset	Top-1 Accuracy	Top-5 Accuracy	Top-10 Accuracy
Training	97.44	98.44	98.72
Validation	33.83	55.85	65.16
Test	32.34	55.15	66.27

Table 2: Results for model 2 features for best model (Random Forests)

Random forests capture co-occurrence of words, thus giving significant improvement over random guessing and style-based features using the word frequency feature vector.

3.2.2 Deep Learning Model - LSTM with mean pooling

Training LSTMs on the entire answers is a computationally expensive task, thus each answer was split into smaller chunks and the model was trained to predict on each chunk rather than every answer.

Dataset	Top-1 Accuracy	Top-5 Accuracy	Top-10 Accuracy
Training	51.32	75.6	83.98
Validation	20.26	37.78	53.51
Test	20.12	36.96	53.42

Table 3: Results for LSTM model on chunk size: 50 words

Comments:

1. Using random word vector initialization performed better than using pre-trained word vectors from the wikipedia dataset. The reason behind this may be that, for this specific task, words which are semantically very similar (eg. synonyms) might still be required to be separated out in the word vector space defined by the way different people use those words.
2. A direct relation is observed between average answer length and accuracy, indicating that author attribution is easier for longer answers. Authors with an average answer length > 400 have at least 70% accuracy. (See figure 6)

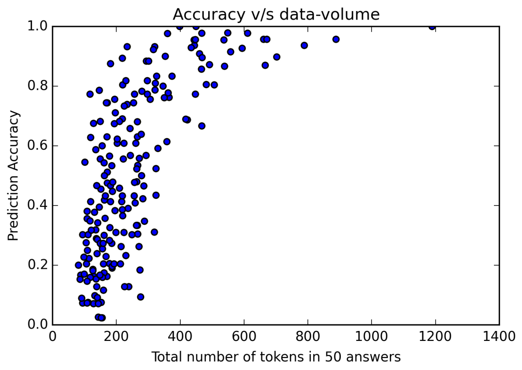


Figure 6: Correlation between accuracy and answer length

3. The performance is much better than random guessing It is lower than the random forest model however which was trained on full answers as compared to chunked answers for the LSTM model training

4 Future Work

Although computationally expensive, training on entire answers should improve results as compared to training on individual chunks.

The model was prone to overfitting to specific words in the training data which could disambiguate between the authors and thus lead to poor generalization. Using a mean pooling layer after a softmax layer at each neuron rather than directly on the hidden layer output might help overcome this, since softmax will normalize all the hidden layer outputs to sum to 1.

Using pre trained word vectors from the wikipedia dataset lead to worse performance than random initialization. However, our dataset was relatively small and results should improve if the word vectors are trained using the author-attribution task on a much larger dataset and initializing the word vectors to these instead.

References

[1] Green, R. M., & Sheppard, J. W. (2013, May). Comparing frequency-and style-based features for twitter author identification. In The Twenty-Sixth International FLAIRS Conference.

[2] Uzuner, zlem, and Boris Katz. ”A comparative study of language models for book and author recognition.” Natural Language ProcessingIJCNLP 2005. Springer Berlin Heidelberg, 2005. 969-980.

[3] Stamatatos, Efstathios. ”A survey of modern authorship attribution methods.” Journal of the American Society for information Science and Technology 60.3 (2009): 538-556.

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780

[5] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation

[6] Graves, Alex. Supervised sequence labeling with recurrent neural networks. Vol. 385. Springer, 2012.