Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Authorship Attribution Using a Neural Network Language Model

Zhenhao Ge, Yufang Sun and Mark J.T. Smith
School of Electrical and Computer Engineering, Purdue University
465 Northwestern Ave, West Lafayette, Indiana, USA, 47907-2035
Emails: {zge, sun361, mjts}@purdue.edu, Phone: +1 (317) 457-9348

Abstract

In practice, training language models for individual authors is often expensive because of limited data resources. In such cases, Neural Network Language Models (NNLMs), generally outperform the traditional non-parametric N-gram models. Here we investigate the performance of a feed-forward NNLM on an authorship attribution problem, with moderate author set size and relatively limited data. We also consider how the text topics impact performance. Compared with a well-constructed N-gram baseline method with Kneser-Ney smoothing, the proposed method achieves nearly 2.5% reduction in perplexity and increases author classification accuracy by 3.43% on average, given as few as 5 test sentences. The performance is very competitive with the state of the art in terms of accuracy and demand on test data. The source code, preprocessed datasets, a detailed description of the methodology and results are available at https://github.com/zge/authorship-attribution.

Introduction

Authorship attribution refers to identifying authors from texts by their unique textual features. It is challenging since the author's style may vary by topics, mood and environment. Many methods have been explored to address this problem, such as Latent Dirichlet Allocation for topic modeling (Seroussi, Zukerman, and Bohnert 2011) and Naive Bayes for text classification (Coyotl-Morales et al. 2006). Regarding language modeling methods, there is mixed advocacy for the conventional N-gram methods (Kešelj et al. 2003) and methods using more compact and distributed representations, like Neural Network Language Models (NNLMs), which was claimed to capture semantics better with limited training data (Bengio et al. 2003).

Most NNLM toolkits (Mikolov et al. 2010) are designed for recurrent NNLMs which are better for capturing complex and longer text patterns and require more training data. In contrast, the feed-forward NNLM framework we proposed is less computationally expensive and more suitable for language modeling with limited data. The database is composed of transcripts of 16 courses from Coursera, collected one sentence per line into a text file for each course. To reduce the influence from text topics, courses were all selected from science and engineering fields, such as Algorithm, DSP, Data Mining, Machine Learning, etc. There are 8000+ sentences/course and about 20 words/sentence on average. The author vocabulary sizes varies from 3000 to 9000. After stemming with Porter's algorithm and pruning words with frequency less than $1/10^5$, they are reduced to a range from 1800 to 2700, with average size around 2000. Fig. 1 shows the vocabulary size for each course, under various conditions and the database coverage with the most frequent $500, 1000, 2000$ words after stemming and pruning.

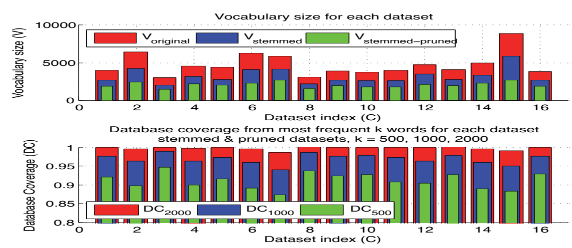

Figure 1: Data vocabulary sizes and word coverages

Neural Network Language Model (NNLM)

Similar to N-gram methods, the NNLM is also used to answer one of the fundamental questions in language modeling: predicting the best target word $\mathcal{W}^*$, given a context of $N-1$ words. The target word is typically the last word within context size N. Fig. 2 shows the structure of the proposed NNLM with multinomial classification cost function $C = -\sum_V t_j \log y_j, j \in V$, where V is the vocabulary size, y_j and t_j are the final output and the target label. This NNLM setup contains 4 types of layers. The word layer contains $N-1$ input words represented by V-dimensional index vectors with $V-1$ "0"s and one "1" positioned in a different location to differentiate it from all other words. Words are then transformed to their distributed representation and concatenated in the embedding layer. Outputs from this layer forward propagate to the hidden sigmoid layer, then softmax layer to predict the probabilities of the possible target words. Weights/biases between layers are initiated randomly and with zeros respectively, and their error derivatives are computed through backward propagation. The network is iteratively updated with model training parameters.

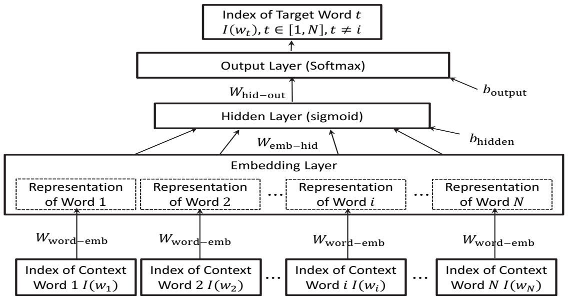

Figure 2: A feed-forward NNLM setup (I: index, $\mathcal{W}$: word, N: number of context words, W: weight, b: bias)

In implementation, the processed text data for each course are randomly split into training, validation, and test sets with ratio 8:1:1. This segmentation is performed 10 times with different randomization seeds, so the mean/variance of NNLM performance can then be measured. We optimized a 4-gram NNLM with mini-batch training through 10 to 20 epochs for each course. The model parameters, such as number of nodes in each layer, learning rate, and momentum are customized for obtaining the best individual models.

Classification with Perplexity Measurement

Denote $\mathcal{W}_1^n$ as a word sequence $(\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_N)$ and $P(\mathcal{W}_1^n)$ as the probability of $\mathcal{W}_1^n$ given a LM, perplexity is an intrinsic measurement of the LM fitness defined by:

$$PP(\mathcal{W}_1^n) = P(\mathcal{W}_1^n)^{-\frac{1}{n}} \quad (1)$$

Using Markov chain theory, $P(\mathcal{W}_1^n)$ can be approximated by the probability of the closest N words $P(\mathcal{W}_{n-N+1}^n)$, so $PP(\mathcal{W}_1^n)$ can be approximated by

$$PP(\mathcal{W}_{n-N+1}^n) = (\prod_{k=1}^{n} P(\mathcal{W}_k|\mathcal{W}_{k-N+1}^{k-1}))^{-1/n} \quad (2)$$

The mean perplexity of applying 4-gram NNLMs to the test sets are 67.3 ± 2.4. This is lower (better) than the traditional N-gram method (69.0 ± 2.4 with 4-gram SRILM). The classification is performed by finding the author with his/her NNLM that maximizes the accumulative perplexity of the test sentences. By randomly selecting 1 to 20 test sentences from the test set, Fig. 3 shows the 16-way classification accuracy using 3 methods, for one particular course and for all courses on average. There are 2 courses from the same instructor, intentionally added to investigate the topic impact on accuracy. They are excluded when computing the average accuracy in Fig. 3. Similarly, the accuracies for courses using two methods with differing text lengths are compared in Fig. 4. Both figures show the NNLM method is slightly better than the SRI baselines at the 4-gram level. A classification confusion matrix (not included due to space limits) was also computed to show the similarity between authors. The results show higher confusion on similar courses, which indicates the topic does impact accuracy. The NNLM has higher confusion values than the SRI baseline on the two different courses from the same instructor, so it is more biased toward the author rather than the topic in that sense.

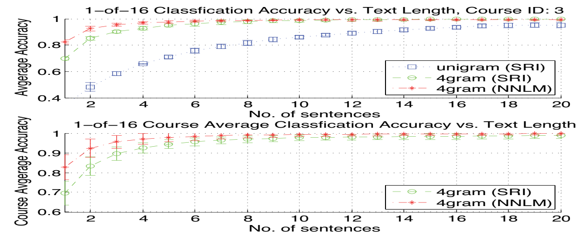

Figure 3: Individual and mean accuracies vs. text length in terms of the number of sentences

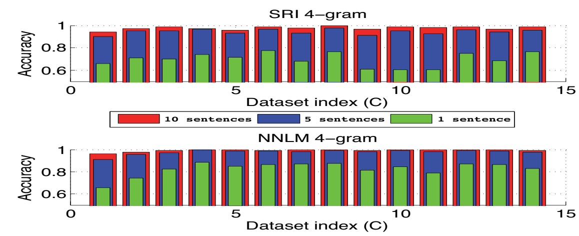

Figure 4: Accuracies at 3 stages differed by text length for 14 courses (2 courses from the same instructor are excluded)
.

Conclusion and Future Work

The NNLM-based work achieves promising results compared with the N-gram baseline. The nearly perfect accuracies given 10+ test sentences are competitive with the state-of-the-art, which achieved accuracy 95%+ on a similar author size (Coyotl-Morales et al. 2006), or $80\%+$ with tens of authors and small datasets (Seroussi, Zukerman, and Bohnert 2011). However, it may also indicate the task is not challenging enough, probably due to the training/test data consistency and the topic distinction. In the future, datasets with more authors taken from book collections or transcribed speeches can be explored. We can also use a nonlinear function optimization scheme with conjugate gradient (Rasmussen 2006), to automatically selects the best training parameters and improve training efficiency. To compensate for the small training set, LMs may also be trained with multiple authors and then adapted to the individuals.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research* 3:1137–1155.

Coyotl-Morales, R. M.; Villaseñor-Pineda, L.; Montes-y Gómez, M.; and Rosso, P. 2006. Authorship attribution using word sequences. In *Progress in Pattern Recognition, Image Analysis and Applications*. Springer.

Kešelj, V.; Peng, F.; Cercone, N.; and Thomas, C. 2003. N-gram-based author profiles for authorship attribution. In *PACLING*.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010*.

Rasmussen, C. E. 2006. Gaussian processes for machine learning.

Seroussi, Y.; Zukerman, I.; and Bohnert, F. 2011. Authorship attribution with latent dirichlet allocation. In *CoNLL*.