Exploring Word Embeddings and Character N-Grams for Author Clustering

Notebook for PAN at CLEF 2016

Yunita Sari and Mark Stevenson

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello
Sheffield S1 4DP, United Kingdom
E-Mail:{y.sari, mark.stevenson}@sheffield.ac.uk

Abstract We presented our system for PAN 2016 Author Clustering task. Our software used simple character n-grams to represent the document collection. We then ran K-Means clustering optimized using the Silhouette Coefficient. Our system yields competitive results and required only a short runtime. Character n-grams can capture a wide range of information, making them effective for authorship attribution. We also present a comparison study of two different features: character n-grams and word embeddings.

1 Author Clustering

This report describes our system that participated in the PAN 2016 Author Clustering task [18]. The task is to create clusters from a document collection, where each cluster represents a different author. The task itself consists of two different scenarios: The first, **complete author clustering** is to create k different clusters represent k authors and assign each document to exactly one of those clusters. The second, **authorship-link ranking** is reminiscent of information retrieval. Given a group of documents in the same cluster, we have to provide confidence scores between pairs of documents, indicating the likelihood that the document pair was written by the same author.

This year's task[1] consists of 18 problems in 3 languages (English, Dutch and Greek) and 2 genres (newspaper articles and reviews). For each problem the language and genre are uniform, but topics may differ. The lengths of documents vary from a few hundred to a few thousand words.

Evaluation The author clustering task is evaluated on both scenarios, thus two different outputs need to be produced. The BCubed F-Score [1] is used to estimate the quality of clustering. In this case, precision and recall of each item are computed. Precision of an item corresponds to how many items in the same cluster belong to its category, while recall is calculated by counting items from its category that appeared in its cluster. To evaluate the authorship-link ranking, Mean Average Precision (MAP) [9] is used. This

[1] http://pan.webis.de/clef16/pan16-web/author-identification.html

metrics is commonly used in information retrieval task. High score of MAP will be obtained if the system could retrieve most relevant documents to the queries.

To achieve a good performance in both scenarios, we need an optimized clustering algorithm and at the same time choose the right features that could effectively discriminate each author's writing style. We therefore, divide our exploration into two parts: first, we investigate two different features, character n-grams and word embeddings which have been known for their success in text classification tasks. As both of the features work in different ways, we are particularly interested on how they characterize each author's writing. Second, we perform hyperparameter tuning on the clustering algorithm in order to find a model with the optimal number of clusters.

Our system is described in the next section. Results are reported in Section 3 and conclusions drawn in Section 4.

2 System Description

During system development, two different features were used: word embeddings and character n-grams. Our main goal is to investigate whether word embeddings could perform well on a multi-topic author attribution task. The semantic information in word embeddings has been shown to effectively capture similarities between documents [8,19,7]. We expect similar performance on authorship attribution. We also developed another system using character n-grams. Previous work [3,5,6] found that character n-grams are a highly effective feature for authorship attribution.

K-Means was used to define clusters in the document collection. We optimized the hyperparameter k by calculating the Silhouette Coefficient [15] for each of the sample. Our system is developed using Python. We also used Scikit-learn library[2] [12] to implement TfIdfVectorizer for character n-gram, K-Means and the Silhouette Coefficient. Word embeddings were trained using Gensim word2vec[3] [14]. We compared performance of each feature on the training dataset and only submitted the system with the best result.

2.1 Document Representation

Word Embeddings Semantic information has rarely been used in authorship attribution. This is mainly due to the unavailability of NLP tools that could perform semantic analysis with relatively high accuracy [17]. In addition, writing style usually can be characterized using more common stylometric features such as character, lexical, and syntactic information.

Recently, neural network based methods have enjoyed a resurgence in popularity including word embeddings [2,10]. Word embeddings represent word in low-dimensional

[2] http://scikit-learn.org/

[3] https://radimrehurek.com/gensim/

vector based on its contexts. Thus the vector representation might capture not only grammatical and syntactic information but also semantic feature of the word [19,7]

Training word embeddings Word embeddings were implemented on English and Dutch (we were unable to implement word embeddings for Greek due to problems caused by text encoding). For English, we used pre-trained Google word2vec vectors. The vectors have dimensionality of 300 and were trained on 100 billion words from Google News [11]. For Dutch, we used the implementation of word2vec tools from Gensim to train word embeddings on 3.7Gb of Dutch texts. We set the dimensionality of word2vec vectors to 300 and window size to 5. By default, Gensim word2vec uses continuous bag-of-words architecture in the training. To overcome the problem of variable-length documents, we simply used the average word vectors in a given document.

Character n-grams Sapkota et. al [16] provide evidence of how useful character n-grams can be to capture the characteristics of author's writing. We extracted 5000 most frequent character n-grams, which include n ranging from 3 to 8. We then calculated tf-idf score using TfIdfVectorizer from Scikit-Learn library. We did not apply any pre-processing steps, meaning that all function words are included.

2.2 Clustering algorithm

One of the most important steps in author clustering is to determine the correct number of clusters since this corresponds to the number of authors. It is especially challenging in this task, since a large portion clusters consist of only one author. We chose K-Means clustering and used the implementation from Scikit-Learn machine learning library. To optimize the number of clusters, we performed hyperparameter tuning using the Silhouette Coefficient.

Silhouette Coefficient The Silhouette Coefficient[4] works by evaluating the clustering model with different number of k. For each sample, a score is produced. Higher scores correspond to a model with better defined clusters. Equation 3 describes how to calculate the Silhouette Coefficient s for a single sample:

$$s = \frac{b - a}{max(a, b)} \tag{1}$$

where:
a: The mean distance between a sample and all other points in the same class.
b: The mean distance between a sample and all other points in the next nearest cluster.

We calculated the Silhouette Coefficient on a range of values k and picked the value with the highest score.

Authorship Links To produce authorship links score, we simply took the formed clusters which consist of more than one member and calculated their pairwise similarity using cosine similarity metric.

[4] http://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient

2.3 Comparison Study

We ran our system on TIRA [13,4] with different feature settings on the training dataset. The output was evaluated based on the BCubed and MAP metrics. Table 1 shows the average results of author clustering on English and Dutch datasets. Overall, word embeddings perform as well as character n-grams. Our hypothesis is that the word embeddings successfully captured topic differences between author. However, implementation of word embeddings is computationally expensive. It took almost 26 minutes of execution times and significant memory requirements (more than 10Gb) to perform the task. While character n-grams only need 7 minutes and less than 500Mb of memory.

To confirm the effectiveness of word embeddings in authorship attribution, we suggest that further experiment on larger corpus is needed. It would also be interesting to investigate whether word embeddings still achieve good performance on single-domain corpus. After comparing the overall performance of both features, we decided to include character n-grams in our final version of the software that was submitted for evaluation.

Table 1: Author Clustering performance using different feature sets

Lang	character n-grams				average word2vec			
	F-Bcubed	R-Bcubed	P-Bcubed	MAP	F-Bcubed	R-Bcubed	P-Bcubed	MAP
en	0.76902	0.70771	0.86441	0.02100	0.76878	0.70774	0.86708	0.03752
nl	0.80158	0.73102	0.91411	0.05950	0.79664	0.72507	0.91099	0.04678

3 Result and Discussion

We submitted our final software which used tf-idf of character n-grams for all three languages. Table 2 shows results of our system on the training data. On average, our system obtained 0.795 on F-BCubed which indicates that it successfully identified the correct cluster to most of the documents. However, the system failed to perform well when evaluated using the MAP metric. Similar results were obtained on test data, with worse MAP scores (see Table 3). Our system produced the 3rd and 4th best Mean F-Score and MAP respectively in the ranking of all PAN 2016 Author Clustering participants.

MAP is only calculated on clusters containing at least two items. Thus, this score is generally very dependent on how accurately the system assigned clusters to the document collection. High scores for the BCubed metrics but low scores on MAP indicate that the system still not good enough at capturing similarities between documents. In addition, we suspect the nature of the corpus (which contains a lot of single node clusters) is the main reason why the system performs well on BCubed metrics. Experiments on larger corpus with more non-single node clusters would be useful to explore this hypothesis.

Table 2: Author Clustering performance using character n-grams on training data

Problem	Lang	F-Bcubed	R-Bcubed	P-Bcubed	MAP
problem001	en	0.747	0.700	0.800	0.0000
problem002	en	0.662	0.537	0.863	0.0296
problem003	en	0.846	0.873	0.820	0.0209
problem004	en	0.817	0.731	0.925	0.0522
problem005	en	0.875	0.875	0.875	0.0000
problem006	en	0.668	0.530	0.903	0.0234
average	en	0.769	0.708	0.864	0.021
problem007	nl	0.895	0.924	0.868	0.1000
problem008	nl	0.714	0.566	0.965	0.1062
problem009	nl	0.795	0.731	0.871	0.0556
problem010	nl	0.715	0.570	0.960	0.0779
problem011	nl	0.785	0.685	0.920	0.0171
problem012	nl	0.905	0.910	0.900	0.0000
average	nl	0.802	0.731	0.914	0.0595
problem013	gr	0.670	0.539	0.885	0.0439
problem014	gr	0.779	0.703	0.873	0.0005
problem015	gr	0.879	0.885	0.873	0.0625
problem016	gr	0.928	0.965	0.895	0.5556
problem017	gr	0.753	0.618	0.964	0.1434
problem018	gr	0.868	0.806	0.939	0.2859
average	gr	0.813	0.753	0.905	0.1827
overall		**0.795**	**0.730**	**0.894**	**0.0877**

4 Conclusion

We have presented our system which was submitted for PAN 2016 Author Clustering task. We performed experiments using two different features: word embeddings and character n-grams. Results from the experiments show that word embeddings are useful predictive features especially for multi-topic authorship attribution. The utility of word embeddings on capturing semantic information helps to identify the author of the

Table 3: Author Clustering performance using character n-grams on testing data

Problem	Lang	F-Bcubed	R-Bcubed	P-Bcubed	MAP
problem001	en	0.779	0.714	0.857	0.0000
problem002	en	0.678	0.529	0.943	0.0407
problem003	en	0.885	0.914	0.857	0.0000
problem004	en	0.809	0.739	0.894	0.0104
problem005	en	0.887	0.900	0.875	0.0000
problem006	en	0.666	0.533	0.888	0.0016
average	en	0.784	0.722	0.886	0.0088
problem007	nl	0.841	0.784	0.906	0.1361
problem008	nl	0.877	0.896	0.859	0.0250
problem009	nl	0.657	0.531	0.859	0.0026
problem010	nl	0.899	0.890	0.910	0.0625
problem011	nl	0.664	0.520	0.917	0.0263
problem012	nl	0.793	0.710	0.900	0.0000
average	nl	0.789	0.722	0.892	0.0421
problem013	gr	0.808	0.743	0.886	0.0347
problem014	gr	0.708	0.569	0.936	0.0596
problem015	gr	0.871	0.886	0.857	0.0000
problem016	gr	0.843	0.786	0.909	0.0619
problem017	gr	0.899	0.929	0.871	0.0159
problem018	gr	0.749	0.621	0.943	0.2412
average	gr	0.813	0.756	0.900	0.0689
overall		**0.795**	**0.733**	**0.893**	**0.0399**

documents. However the PAN corpus is small and we suggest further experiments are needed.

Our final submission implemented tf-idf character n-grams with K-Means clustering. This simple approach has proved to be effective for author clustering. This year's PAN shared task has encouraged us to explore these approaches on other authorship attribution tasks. We are interested to know how well character n-grams perform on other problems such as large-scale or short text authorship attribution.

5 Acknowledgment

Both authors are grateful for the efforts of the CLEF PAN 2016 team. The first author would like to acknowledge Indonesia Endowment Fund for Education (LPDP) for support in the form of a doctoral studentship.

References

1. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Inf. Retr. 12(4), 461–486 (Aug 2009), http://dx.doi.org/10.1007/s10791-008-9066-8
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155 (Mar 2003), http://dl.acm.org/citation.cfm?id=944919.944966
3. Frantzeskou, G., Stamatatos, E., Gritzalis, S., Katsikas, S.: Artificial Intelligence Applications and Innovations: 3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI) 2006, June 7–9, 2006, Athens, Greece, chap. Source Code Author Identification Based on N-gram Author Profiles, pp. 508–515. Springer US, Boston, MA (2006)
4. Gollub, T., Stein, B., Burrows, S., Hoppe, D.: TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments. In: Tjoa, A., Liddle, S., Schewe, K.D., Zhou, X. (eds.) 9th International Workshop on Text-based Information Retrieval (TIR 12) at DEXA. pp. 151–155. IEEE, Los Alamitos, California (Sep 2012)
5. Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. J. Am. Soc. Inf. Sci. Technol. 60(1), 9–26 (Jan 2009), http://dx.doi.org/10.1002/asi.v60:1
6. Koppel, M., Schler, J., Argamon, S.: Authorship attribution in the wild. Lang. Resour. Eval. 45(1), 83–94 (Mar 2011), http://dx.doi.org/10.1007/s10579-009-9111-2
7. Liu, Y., Liu, Z., Chua, T.S., Sun, M.: Topical word embeddings. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 2418–2424. AAAI'15, AAAI Press (2015), http://dl.acm.org/citation.cfm?id=2886521.2886657
8. Ma, C., Xu, W., Li, P., Yan, Y.: Distributional representations of words for short text classification. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 33–38. Association for Computational Linguistics, Denver, Colorado (May–June 2015), http://www.aclweb.org/anthology/W15-1505
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013), http://arxiv.org/abs/1301.3781
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR abs/1310.4546 (2013), http://arxiv.org/abs/1310.4546

12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
13. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
14. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
15. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53 – 65 (1987), http://www.sciencedirect.com/science/article/pii/0377042787901257
16. Sapkota, U., Bethard, S., Montes, M., Solorio, T.: Not all character n-grams are created equal: A study in authorship attribution. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 93–102. Association for Computational Linguistics, Denver, Colorado (May–June 2015), http://www.aclweb.org/anthology/N15-1010
17. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. 60(3), 538–556 (Mar 2009), http://dx.doi.org/10.1002/asi.v60:3
18. Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Clustering by Authorship Within and Across Documents. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016)
19. Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F., Hao, H.: Semantic clustering and convolutional neural network for short text categorization. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 352–357. Association for Computational Linguistics, Beijing, China (July 2015), http://www.aclweb.org/anthology/P15-2058