Multi feature space combination for authorship clustering:
Notebook for PAN at CLEF 2016

Muharram Mansoorizadeh[1], Mohammad Aminiyan[2], Taher Rahgooy, Mahdy Eskandari

Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran

Abstract . The Author Identification task for PAN 2016 consisted of three different Sub-tasks: authorship clustering, authorship links and author diarization. We developed a machine learning approaches for two of three of these tasks. For the two authorship related tasks we created various sets of feature spaces. The challenge was to combine these feature spaces to enable the machine learning algorithms to detect these difference authors across multiple feature spaces. In the case of authorship clustering we combine these feature spaces and use a two-step approach for clustering. Then we use results of the clustering, and employ new feature space to determine links between documents in given problems.

Keywords: authorship clustering, authorship link, tf-idf, feature space combintion

1 Introduction

In the following we provide a detailed description of our approaches to solve the two subtasks of the Author Identification track of PAN 2016. The problem instance is a tuple <K; U; L> where K is a set of documents <k1, k2, k3,…, kn> authored by the different authors, U is the genre of the document and L is the enumerated value specifying the language of the documents: English, Dutch or Greek. All documents in a problem instance are in the same language and same genre. This lab report is structured as follows: In section 2 we propose a number of different features that characterize documents from widely different points of view: character, word, part-of speech, sentence length, punctuation. We construct non-overlapping groups of homogeneous feature. In section 3 we present the two-step unsupervised method for authorship clustering task by employing a graph based approach and the standard k-means++ algorithm. Then we employ new feature space to determine links

[1] mansoorm@basu.ac.ir

[2] M.Aminiyan@Gmail.com

between documents. Finally, in section 4 we describe our results on the training corpus and the final evaluation corpus of PAN-2016.

2 Preprocessing

We extract a number of different features from each document. For ease of presentation, we group homogeneous features together, as described below.

2.1 Features

Word ngrams (WG): We convert all characters to lowercase and then we transform the document to a sequence of words. We consider white spaces, punctuation characters and digits as word separators. We count all word ngrams, with n ≤ 3, and we obtain a feature for each different word ngram which occurs in the training set documents of a given language [1]. It should be mentioned that, we use word unigrams and 2-gram in clustering task and preprocesses related to it and word 3-gram only used in link computation phase.

In order to normalize these set of features we use term frequency-inverse document frequency (tf-idf) for each set of documents (each problem)[2].

POS (part-of-speech) tag ngrams (PG): We apply a part of speech (POS) tagger on each document, which assigns words with similar syntactic properties to the same POS tag. We count all POS ngrams, with n≤ 2, and we obtain a feature for each different POS ngram which occurs in the training set documents of a given language [2].

Sentence lengths (SL): We transform the document to a sequence of tokens, a token being a sequence of characters separated by one or more blank spaces. Next, we transform the sequence of tokens to a sequence of sentences, a sentence being a sequence of tokens separated by any of the following characters: ., ;, :, !, ?. We count the number of sentences whose length in tokens is n, with n ∈ {1,..,15}: we obtain a feature for each value of n [2].

Punctuation ngrams (MG): We transform the document by removing all characters not included in the following set: {,, ., ;, :, !, ?, "}—the resulting document thus consists of a (possibly empty) sequence of characters in that set. We then count all character ngrams of the resulting document, with n≥2, and we obtain a feature for each different punctuation ngrams which occurs in the training set documents of a given language [2].

In order to preprocess documents we use python NLTK 3.0 package [3]. After creating the feature space we simply separate word 2grams for authorship link task and use the rest of features for clustering. We assume that word 2grams consist of very specific relation which can effect better inside of each cluster for determining the level of similarity between documents.

2.2 Data normalization

After feature extraction, we normalize value of each feature using min-max normalization in order to remove the impact of different scale spaces:

$$X_{new} = \frac{X_{old} - Max}{Max - Min} \tag{1}$$

Where X_{old} is the old value of X and Max is the maximum value of feature X and Min is the Minimum value for feature X.

3 Two-step unsupervised method

In order to solve the task, we use two step method.

3.1 Step 1: Determining the number of authors

Considering the fact that number of authors is unknown first we have to determine the number of authors for each problem, namely, we have to determine number of clusters for clustering algorithm. The number of clusters should be set by the developer based on specifications of problem. Assigning a proper number is a challenging task. A document similarity graph (DSG) algorithm has been used. DSG is an undirected graph showing similarity relations between documents based on their contents [4]. The nodes of this graph are documents and the edges between documents are defined by the similarities and dissimilarities between them using (2):

$$\begin{gathered} Z(i,j) = 1 - \frac{X\,Y}{|X|.|Y|} = 1 - \frac{\sum_{i=1}^{n} x_i . y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}} \\ VS_{mat(i,j)} = \begin{cases} 1 & Z(i,j) \geq \delta \\ 0 & Z(i,j) < \delta \end{cases} \end{gathered} \tag{2}$$

Where x_k and y_k are features of X_i and Y_j documents respectively and δ is the threshold which define the existence of the similarity between two documents. In this paper, the δ parameter is set to 0.5. Also Z is the cosine similarity between two documents [5].

The number of clusters has been determined using the number of sub graphs resulted with DSG. To find the number we just count the nodes with value more than 65 percent of number of all document for example if we have 100 documents in problem folder, we count nodes which have more than 65 incoming edges.

3.2 Step 2: clustering and computing links

After calculating the number of clusters, we use k-means++ [6] scikit-learn python package in order to perform clustering task.

When clustering completed, we collect the result and employ simple similarity task in each of clusters. We compute similarity based word 3grams features and cosine similarity (2).

4 Results

In order to evaluate our work, we use training corpus and the final evaluation corpus of PAN-2016. These datasets consist of set of problems, each problem comes with different number of documents in specific language (English, Dutch and Greek) and two different genres (newspaper articles and reviews). The clustering output will be evaluated according to BCubed F-score [7] and the ranking of authorship links will be evaluated according to Mean Average Precision (MAP) [8]. In order to evaluate our work, first the software has been executed on TIRA platform [9].

Table 1 shows the result of train dataset. It is obvious that our method have high Bcubed recall hence we can say the method cluster same items almost great in each cluster but by investigating our method's Bcubed precision, we can clearly say that the number of cluster or even the way we measure similarity does not tune well.

Table 1. Results of test dataset

problem	language	Genre	F-Bcubed	R-Bcubed:	P-Bcubed	Average Precision
problem01	English	Articles	0.71947	0.71333	0.72571	0.00083612
problem02	English	Articles	0.58281	0.50444	0.69	0.00022599
problem03	English	Articles	0.58665	0.87333	0.44167	0.00052301
problem04	English	Reviews	0.76012	0.69583	0.8375	0.0015432
problem05	English	Reviews	0.2648	0.97083	0.15331	0.0028001
problem06	English	Reviews	0.24887	0.89667	0.14449	0.017832
problem07	Dutch	Articles	0.45478	0.96491	0.2975	0.0084819

problem08	Dutch	Articles	0.68125	0.59223	0.80175	0.030246
problem09	Dutch	Articles	0.42888	0.8538	0.28636	0.016233
problem10	Dutch	Reviews	0.36209	0.61333	0.25687	0.0063669
problem11	Dutch	Reviews	0.33539	0.71167	0.21939	0.001594
problem12	Dutch	Reviews	0.33242	0.92	0.20286	0.0008547
problem13	Greek	Articles	0.1365	0.89697	0.073869	0.023333
problem14	Greek	Articles	0.53793	0.77939	0.4107	0.0095962
problem15	Greek	Articles	0.56034	0.93939	0.39924	0.007211
problem16	Greek	Reviews	0.56877	1	0.3974	0.013893
problem17	Greek	Reviews	0.5674	0.74697	0.45743	0.045313
problem18	Greek	Reviews	0.57607	0.90303	0.42294	0.028917

Like Table 1, Table 2, results of test dataset, also illustrates, high level of Bcubed recall in most of problem sets, in contrast with Bcubed precision which is not high. But it is obvious that results from test dataset are better than train data. It shows ability of system to generalize new problems. But the major defect of system with lower Bcubed precision than recall one still exists.

Notice that you can find complete evaluation on overview [10].

Table 2. Results of test dataset

problem	language	Genre	F-Bcubed	R-Bcubed:	P-Bcubed	Average Precision
problem01	English	Articles	0.4492	0.77619	0.31605	0.0045282
problem02	English	Articles	0.51302	0.6165	0.43929	0.018634
problem03	English	Articles	0.45086	0.9381	0.29674	0.0022228
problem04	English	Reviews	0.46821	0.80833	0.32955	0.0033492
problem05	English	Reviews	0.54696	0.92083	0.38902	0.00076857
problem06	English	Reviews	0.4896	0.64458	0.3947	0.0046202
problem07	Dutch	Articles	0.06261	1	0.032318	0.017739
problem08	Dutch	Articles	0.33159	0.95906	0.2004	0.0053791
problem09	Dutch	Articles	0.15954	0.94987	0.087	0.032996
problem10	Dutch	Reviews	0.33115	0.89667	0.20308	0.00097662
problem11	Dutch	Reviews	0.31324	0.56167	0.21718	0.0022789
problem12	Dutch	Reviews	0.3371	0.73167	0.219	0.00074413
problem13	Greek	Articles	0.43173	0.76429	0.3008	0.02234
problem14	Greek	Articles	0.46847	0.7119	0.34909	0.015947

problem15	Greek	Articles	0.43579	0.9	0.2875	0.00087489
problem16	Greek	Reviews	0.48623	0.83571	0.34286	0.007296
problem17	Greek	Reviews	0.46259	0.98095	0.30266	0.003199
problem18	Greek	Reviews	0.47588	0.79524	0.33953	0.0095474

5 Conclusion and future works

In this research we propose a two-step unsupervised method in order to perform author clustering. In our approach we combine different feature spaces and use them to cluster documents based on their authors. Then, we rank documents based on their cosine similarity using new set of feature which are different from the set we use for clustering.

Results illustrates that our work have a good Bcubed recall. But major problem of our method was its Bcubed precision. The problem may come from cluster number selection or the feature space. Hence as a future work, we suggest researchers work on a way of better cluster parameter selection. Also, it would be suggested that the task tested on more complex clustering method without the need on parameter selection like self-organized map (SOM) and so on.

References

1. Tuarob, S., Pouchard, L. C., Mitra, P., & Giles, C. L. (2015). A generalized topic modeling approach for automatic document annotation. International Journal on Digital Libraries, 16(2), 111-128.
2. Alberto Bartoli, Alex Dagri, Andrea De Lorenzo, Eric Medvet, and Fabiano Tarlao. An Author Verification Approach Based on Differential Features—Notebook for PAN at CLEF 2015. In Linda Cappellato, Nicola Ferro, Gareth Jones, and Eric San Juan, editors, CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France, September 2015. CEUR-WS.org. ISSN 1613-0073.
3. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python. " O'Reilly Media, Inc.".
4. B. Seah, S. S. Bhowmick, and A. Sun, “PRISM : Concept-preserving Social Image Search Results Summarization,” in Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR ’14, 2014, pp. 737–746
5. Deza, Michel Marie, and Elena Deza. Encyclopedia of distances. Springer Berlin Heidelberg, 2009.
6. Arthur, D., Vassilvitskii, S.: k-means ++ : The Advantages of Careful Seeding. Proceedings of the eighteenth annual ACMSIAM symposium on Discrete algorithms 8(2006-13), 1027–1035 (2007
7. Amigó, E., Gonzalo, J., Artiles, J., & Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints.Information retrieval, 12(4), 461-486.

8. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval (Vol. 1, No. 1, p. 496). Cambridge: Cambridge university press.
9. 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).
10. 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)