Author Diarization Using Cluster-Distance Approach
Notebook for PAN at CLEF 2016

Abdul Sittar, Hafiz Rizwan Iqbal, and Rao Muhammad Adeel Nawab

COMSATS Institute of Information Technology,
M.A. Jinnah Campus, Off-Raiwind Road, Lahore, Pakistan
abdulsittar72@yahoo.com,{rizwan.iqbal,adeelnawab}@ciitlahore.edu.pk

Abstract. *Author Diarization* is a new task introduced in PAN'16, to identify portion(s) of text with in a document written by multiple authors. This paper presents, our proposed approach for author diarization task. Various types of *stylistic features* which include *lexical features*, used to uniquely identify an author. Furthermore, to find anomalous text with in a single document, *ClustDist* method used. Finally, clusters were generated by using simple *k-means* clustering algorithm. Experiments were performed both on training and testing data sets. It has been observed that by changing the text fragments length, promising results can be achieved.

1 Introduction

Plagiarism detection is the task of determining text reuse in a document or collection of documents. It is of two types, 1) Extrinsic Plagiarism Detection - focuses on finding source documents which were used to generate suspicious documents. It is a complex task because it may happen at various levels ranges from word to sentences, paraphrasing of text and plagiarism of ideas, and 2) Intrinsic Plagiarism Detection or Author Diarization - focuses on identifying whether a document is written by a single or multiple authors.

Author Diarization term came from the domain of speaker diarization - focuses on clustering and identifying various speakers from a single audio speech signal, by analyzing frequency range of speaker's voices e.g. a class discussion on a particular topic and a conference conversation on mobile phones etc. Similarly, author diarization deals with the written document instead of audio conversation [4]. Such documents may be resulted from a collaborative work or plagiarism. It's objective is to identify and cluster different authors with in a single document on the basis of written text.

In literature, stylometric features like function words, part of speech tag, spelling mistakes, average sentence length and average word length were used for Author Diarization task [7]. David Guthrie [2] introduced the term *authorship anomalies*, to detect portion of text that deviate from the original context as written by other author.

PAN[1] is the competition held as a part of CLEF Conference. The PAN'16 [8] competition is designed for three different tasks namely, *Author Identification*, *Author Profiling* and *Author Obfuscation*. *Author Identification* has further divided into *Author clustering* and *Author Diarization*. Identify and cluster portions of text with respect to individual author for a given document, is the task for *Author Diarization*. Moreover, the task divided into three sub problems, 1) Traditional intrinsic plagiarism detection, 2) Diarization with a given number (n) of authors and 3) Diarization with unknown number of authors, to explore different variants of the parent task. Participants have to developed a software for the given task and deploy it on TIRA (An engine to perform evaluation of software) [5]. All training and testing documents are provided only in English language.

For each subtask, all text files have been read from it's respective provided training data set. Each file text splitted into sentences. For each sentence, 15 lexical features (see table 1) were computed. Using *ClustDist*[2] anomaly detection technique, a feature vector generated which contains average distances of all sentences from each other. On the basis of these distances, training models were created using WEKA and saved to use at the time of prediction. Our developed software deployed on TIRA and evaluation performed on testing corpora provided by PAN. Our software generated promising results on both training and testing corpora.

Rest of this paper is organized as follows: The details of the methodology is explained in the Section 2 while experimental setup, results of training phase and testing phase are discussed in the Section 3 and Section 4 respectively. Section 5 provides conclusion and future work.

2 Proposed Approach

A wide variety of algorithms have been reported in literature for authorship identification [7, 2], which includes Cluster distance, Counting Words, Stylometric features, Syntactic features, Lexical features, Content specific and Content free features.

2.1 Lexical Features

A language independent approach for author identification, considers that any text (i.e. sentences, paragraphs, documents) is a sequence of tokens [2]. On the basis of these tokens, various types of statistics (e.g. average word length, average sentence length, characters count) could be drawn from any text of any language etc. [7]. Table 1 shows lexical features, which were used in our developed software to find unique writing style of an author.

[1] `http://pan.webis.de/` Last visited: 25-05-2016

[2] A token could be a word, character, punctuation mark or numeric number.

Table 1. Lexical Features

Feature. No.	Feature Name
1	Characters Count
2	Digits Count
3	Uppercase letters Count
4	Spaces Count
5	Letters Count
6	Tabs Count
7	Words Count
8	Ratio of Interrogative Sentences
9	Average Word Length
10	Average Sentence Length
11	Ratio of Digits
12	Ratio of Uppercase letters
13	Ratio of Spaces
14	Ratio of letters
15	Ratio of tabs

2.2 Clustering using ClustDist

ClustDist [2] - a straightforward technique to compute the average distance from one portion (i.e. sentences) of text to all other pieces of text. Consider a document D with n number of sentences. At first, each sentence i was distinguished by computing the p lexical features (see Table 1) and a feature vector Vi for this sentence would be generated. For our experiments, a matrix V of order n X p was created. Each matrix row shows a feature vector for each sentence. *ClustDist* computed by using equation 1, where d is the distance between any pair of vectors. The resultant score for each sentence distance from others, generates a ranking which describes that how a sentence is different from all other sentences in the given document. To generate clusters, we used simple *k-means* algorithm [3]. For detailed insight into the proposed approach, see section 2.2.

$$ClustDist(\boldsymbol{x}, V) = \frac{\sum_{k} d(\boldsymbol{x}, \boldsymbol{v})}{n} \tag{1}$$

Example: Step-by-Step Author Diarization by ClustDist Approach This example demonstrates author diarization process from an input text to output clusters.

– **Step 1: Read Raw Input Text**
"*In what follows, we give a detailed overviewof Barack Obama's Family. We shed light on himself, his immediateand extended family, including maternal and paternal relations.Moreover, we give insights into the relations of Michelle Obama Barack Obama's wife, as well as some distant relations ofboth. Barack Obama Barack Hussein Obama II is the44th and current President of the United States. He is the firstAfrican American to hold the office.*

Obama was the junior UnitedStates Senator from Illinois from 2005 until he resigned followinghis election to the presidency. Obama is a graduate of ColumbiaUniversity and Harvard Law School. "

– **Step 2: Break Down Text into Sentences**

1. In what follows, we give a detailed overview of Barack Obama's Family.
2. We shed light on himself, his immediate and extended family, including maternal and paternal relations.
3. Moreover, we give insights into the relations of Michelle Obama Barack Obama's wife, as well as some distant relations of both.
4. Barack Obama Barack Hussein Obama II is the 44th and current President of the United States.
5. He is the first African American to hold the office.
6. Obama was the junior United States Senator from Illinois from 2005 until he resigned following his election to the presidency.
7. Obama is a graduate of Columbia University and Harvard Law School.

– **Step 3: Lexical Features Computation**

Table 2 shows the computations of lexical features used in our software. For each sentence, all of the features (see Table 1) are computed in our software. For the sake of demonstration only 4 features computations shown here.

Table 2. Lexical Features Computations

Sentences	Letters Ratio	Average Word Length	Spaces Ratio	Character Count
1	0.91	4.35	0.19	61
2	0.97	5.68	0.16	91
3	0.93	4.73	0.19	109
4	0.92	4.93	0.18	79
5	0.91	4.09	0.22	45
6	0.92	5.19	0.18	109
7	0.94	4.83	0.18	58

– **Step 4: Distance Calculation**

Table 3 shows distance calculation of each sentence from all other sentences in the text.

– **Step 5: ClustDist Computation**
For each sentence, it's *ClustDist* score is as follows:
ClustDist (1) = 0 + 30 + 48 + 18 + 16 + 48 + 3.03 = 163.03
ClustDist (2) = 30 + 0 + 18.02 + 12.02 + 46.02 + 18 + 33.01= 157.07

Table 3. Distance Calculations

Sentences	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	0	30	48	18.0	16	48	3.03
2	30.0	0	18.02	12.02	46.02	18.0	33.01
3	48	18.02	0	30	64	0.46	51
4	18.0	12.02	30.0	0	34.01	30.0	21
5	16	46.02	64	34.01	0	64.0	13.02
6	48	18.0	0.46	30.0	64.0	0	51.0
7	3.03	33.01	51.0	21.0	13.02	51.0	0

ClustDist (3) = 48 + 18.02 + 0 + 30 + 64 + 0.46 + 51 = 211.48
ClustDist (4) =18.0 + 12.02 + 30.0 + 0 + 34.01 + 30.0 + 21 = 145.03
ClustDist (5) = 16 + 46.02 + 64 + 34.01 + 0 + 64.0 + 13.02 =237.05
ClustDist (6) = 48 + 18 + 0.46 + 30 + 64 + 0 + 51 =211.46
ClustDist (7) = 3.03 + 33.01 + 51 + 21 + 13.02 + 51 =172.06

– **Step 6: Generating Clusters**
On the basis of *ClustDist* score, we applied simple *k-means* clustering algorithm and got the following clusters:
Cluster 1: [237.05, 211.46, 211.48]
Cluster 2: [163.03, 157.07, 172.06]
Cluster 3: [145.03]

3 Experimental Setup

This section provides a detailed insight into the experimental environment, setup for the development and evaluation of Author Diarization software.

3.1 Fabrication: Training and Classification Models

For each of the three sub tasks (see section 1), different training data set's are provided in PAN'16. Each training data set contains three files, *".txt"* file contains actual text written by multiple authors, *".meta"* file contain *JSON* object which provides *text language*, *problem type(plagiarism or diarization)* and *number of authors*, and a *".truth"* file which provides the output required against the given text file.

For the generation of trained models, all text files were read from the corpus in a sequence. Each file break down into sentences. All of the lexical features (see section 2.1) were computed for each sentence, created a feature vector containing distances of this sentence from all other sentences using *ClustDist* technique (see section 2.2). WEKA [3](A machine learning tool kit) used to generate and save training models by generating *".arff"* file from this resultant distance vector. For cluster generation, we used simple *k-means* algorithm.

[3] `http://www.cs.waikato.ac.nz/ml/weka/` Last visited: 25-05-2016

3.2 Software Evaluation

PAN'16 also provided the data set to evaluate the developed software but this data set is not publicly available because they will launch it after the competition. However, after training our software on training data set, we deployed it on *TIRA* for the evaluation phase and executed evaluation software on it. This time software takes input files from testing corpus. For each input file, *"test.arff"* file generated using *ClustDist* feature vector. Finally, we got clusters with respect to each subtask as per requirement of PAN'16. e.g. For subtask 1, only two clusters were generated, one for main author and second for the rest of the authors. For subtask 2, clusters were created as per required number in *".meta"* file. For subtask 3, random number of clusters were created because its number of author were not given. As final step, clusters generated from evaluation corpus compared with the existing trained models to predict new instances.

3.3 Evaluation Measures

PAN'16 recommended different evaluation measures for sub tasks 1, 2 and 3. For subtask 1, *micro-recall*, *micro-precision*, *macro-recall*, *macro-precision*, *micro-f* and *macro f* [6] measures used to evaluate the performance of the system. For subtasks 2 and 3 *bcubed-recall*, *bcubed-precision* and *bcubed-f* measure used for evaluation [1].

4 Results and Analysis

This section presents the results, generated using training and testing data. For each of the three sub tasks, size of text fragments were increased to get better results. Experiments showed improvement in results upto some extent. Section 4.1 discusses results on training data and Section 4.2 elaborates results on testing data for each of the three sub tasks.

4.1 Results: Training Phase

For Task 1, results are shown in Table 4. It can be analyzed that highest results for all evaluation measures are obtained for sentences of length 7. Table 5 and Table 6 show the results for subtask 2 and subtask 3 respectively. Sentences of length 5 demonstrated best results for subtask 2 while in subtask 3, sentences of similar length shows highest values of the required evaluation measures.

4.2 Results: Testing Phase

Based upon the result of training data set, we used only those sentence lengthes which demonstrated highest results for each sub task, because both training and testing data sets contain almost similar type of texts. Therefore, for sub task 1, we used sentence length 7. For task 2 and 3, sentence length 5 has been used respectively. Table 7 shows the results for sub task 1 while for sub task 2 and 3, the results are shown in table 8.

Table 4. Training Data Set : Sub Task 1 Results

Sentence Length	Micro-recall	Micro-Precision	Micro-F	Macro-Recall	Macro-Precision	Macro-F
2	0.1338	0.2006	0.1605	0.1216	0.2006	0.1514
3	0.1045	0.1828	0.1330	0.1109	0.1823	0.1379
4	0.1291	0.2492	0.1701	0.1178	0.2492	0.1600
5	0.1392	0.2599	0.1813	0.1337	0.2599	0.1766
6	0.1461	0.2572	0.1864	0.1421	0.2572	0.1830
7	**0.1493**	**0.2655**	**0.1911**	**0.1648**	**0.2664**	**0.2036**
8	0.1130	0.1998	0.1444	0.1129	0.1995	0.1442
9	0.1280	0.2323	0.1651	0.1304	0.2322	0.1670
10	0.1045	0.1828	0.1330	0.1109	0.1823	0.1379
11	0.1379	0.2547	0.1875	0.1481	0.2523	0.1866
12	0.1165	0.2315	0.1550	0.1103	0.2307	0.1492
15	0.1301	0.2573	0.1728	0.1242	0.2565	0.1674

Table 5. Training Data Set : Sub Task 2 Results

Sentence Length	Bcubed-Recall	Bcubed-Precision	Bcubed-F
5	0.4823	0.2861	**0.3591**
10	0.5951	0.1315	0.2154
12	0.6143	0.1080	0.1838
13	0.6260	0.1051	0.1800
14	0.6376	0.0887	0.1558

Table 6. Training Data Set : Sub Task 3 Results

Sentence Length	Bcubed-Recall	Bcubed-Precision	Bcubed-F
5	0.5464	0.2822	**0.3722**
10	0.6253	0.1339	0.2206
12	0.6386	0.1076	0.1842

Table 7. Testing Data Set : Sub Task 1 Results

Sentence Length	Micro-recall	Micro-Precision	Micro-F	Macro-Recall	Macro-Precision	Macro-F
7	0.0672	0.1427	0.0914	0.0951	0.1427	0.1141

Table 8. Testing Data Set : Sub Tasks 2 and 3 Results

Sentence Length	Task	Bcubed-Recall	Bcubed-Precision	Bcubed-F
5	2	0.4700	0.2791	0.3502
5	3	0.4676	0.3140	0.3757

5 Conclusion

In this paper we have discussed our participation in PAN'16 Author Diarization task. We have developed a software for all of the 3 subtasks. Our proposed approach based upon a language independent technique to uniquely identify an

author based upon his/her written text *Lexical Features*. Fifteen lexical features were used in combination with *ClustDist* approach. We used *ClustDist* method for the detection of anomalous text with a document. Experiments were performed on training and testing data sets of PAN'16. Different size of text fragments were used to improve the results in training phase while in testing phase only those fragments sizes were used which gave best results in training phase. It has been analyzed that by changing the fragment sizes of text, improvement in results can be obtained. As future work, content based, topic based and stylistic features in combination with the *ClustDist* method will be explored.

References

1. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information retrieval 12(4), 461–486 (2009)
2. Guthrie, D.: Unsupervised Detection of Anomalous Text. Ph.D. thesis, University of Sheffield (2008)
3. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA. (1967)
4. Miro, X.A., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., Vinyals, O.: Speaker diarization: A review of recent research. Audio, Speech, and Language Processing, IEEE Transactions on 20(2), 356–370 (2012)
5. 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 AuthorProfiling. 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)
6. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. In: Proceedings of the 23rd international conference on computational linguistics: Posters. pp. 997–1005. Association for Computational Linguistics (2010)
7. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), 538–556 (2009)
8. 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)