API Document

Main objective: Disambiguate the relation of Author and Paper. As there are false Author-Paper pair in the given dataset, filter them out and give true Author-Paper pairs with highest probabilities.

1.Dependencies

1)Python Version 3
2)Scikit-learn
3)Pandas
4)Pickle
5)Unidecode
6)Numpy
7)Python-Levenshtein
8)NLTK

2.Dataset

Datasets given from the organizers are saved in dataRev2 folder. There are 9 different csv files.
1)Author.csv: Contains information of author profile. Names and affiliations of authors are given in the file.

2)Conference.csv: Contains information of conference. Shortnames, fullnames and homepages of conferences are given in the file.

3)Journal.csv: Contains information of journal. Shortnames, fullnames and homepages of journals are given in the file.

4)Paper.csv: Contains information of paper. Title, year, corresponding journal and conference ids of papers are given in the file.

5)PaperAuthor.csv: Contains noisy data representing the matching between paper and author. Names and affiliations of authors are also given in the file. However, there are many false matching papers in the csv.

6)Train.csv: Contains information of true and false matching between paper and author. This file is used to train the classifier.

7)Valid.csv: Contains paper author pair query that needs to be determined true or false. It is used to modify the structure of trained classifier.

8)ValidSolution.csv: Contains the true query occurred in Valid.csv. It is used to calculate the MAP score for the valid dataset.

9)Test.csv: Contains paper author pair query that needs to be determined true or false. It is used to give the final score for our system.

2.How to run

0)Quick Start
Run “min_run.sh” to go through the entire process. (You might have to change “python3” command to “python” depending on your system.)

1)Extract feature
In results folder, there are already extracted features saved in “.pickle” format. However, to extract features, run following commands in the terminal. (Take several hours.)
“python extract_token.py”
“python extract_feature.py”

2)Train classifier
After feature extraction is done, you can train your classifier with saved features. The default classifier is Gradient Boosting Machine (XGBoost package). You can change the classifiers by erasing comments in train.py file. Run following command in the terminal.
“python train.py”

3)Predict result
Prediction for queries can be now made with saved classifier. Run following commands in the terminal to generate predict result in csv file. (Generated file name is “basicCoauthorBenchmarkRev2” in results folder). Run following command in the terminal.
“python predict.py”

4)Calculate score
We can now compare the prediction with ground truth. Run following command in the terminal to calculate MAP score.
“python compare.py”

* If you want to check accuracy for test submission, you can submit “testSubmssion.csv” file in “results” folder to official Kaggle KDD-cup 13 site.