Name Disambiguation - aiming at disambiguating WhoIsWho

Zhang Xvdong201934750*
Shandong University
Jimo Qu, Qingdao Shi, China
zxd7799@foxmail.com

ABSTRACT

In many applications, the eponymous disambiguation(Name Disambiguation - aiming at disambiguating WhoIsWho) has been regarded as a challenging issue, such as scientific document management, character search, social network analysis, etc., and with the proliferation of scientific literature, the resolution of the problem has become more difficult and urgent. Although the eponymous disambiguation has been extensively studied in academia and industry, the problem has not been solved well due to the clutter of data and the complexity of the eponymous scenario.

KEYWORDS

data mining, clustering, isambiguation, text tagging

ACM Reference Format:
Zhang Xvdong201934750. 2019. Name Disambiguation - aiming at disambiguating WhoIsWho. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online academic search systems (such as Google Scholar, Dblp and AMiner, etc.) that include a variety of papers, have become the most important and popular academic exchange and paper search platform in the global academic community.[1] However, due to the limitation of the paper distribution algorithm, there are a large number of papers distribution errors within the existing academic system, in addition, a large number of new papers will enter the system every day. Therefore, how to accurately and quickly assign papers to the existing author files in the system and maintain the consistency of author files is an urgent problem to be solved in the existing online academic system. Because of the huge amount of data within the academic system (AMiner has a file of approximately 130,000,000 authors and more than 200,000,000 papers), [2] the author's eponymous scenario is complex, and there are still significant obstacles to resolving the problem of the eponymous disambigfoot quickly and accurately. The competition hopes to propose a problem-solving model that can distinguish papers of the same name belonging to different authors based on the details of the paper and the link between the author and the paper, and obtain good results from the paper. [3] The good results of demystifying results can also affect other related fields by ensuring the effectiveness of expert knowledge search, high-quality content management of digital libraries and personalized academic services in academic systems.

*Both authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND

2.1 Data Mining Steps

Data mining is to obtain useful regular information according to the algorithm model from a large amount of data for subsequent work and decision-making.[4] So the general data mining steps are: data acquisition, data preprocessing, data modeling:

Question Category. First determine whether the problem is a classification problem or a regression problem. The incremental disambiguation of the paper is a classification problem.

Feature Selection Methods. Feature selection model can also be divided into 4 parts:

- `Embedded selection`: combining feature selection with a learner to allow automatic feature selection during model training, such as various tree models;
- `Wrapped selection`: using the model's final learning performance as a feature set Evaluation criteria, choose a good feature set, but because it need s to train multiple models, it is expensive;
- `Filter selection`: calculate the correlation between each feature and the corresponding variable, filter out the less relevant features, but the actual In the application, the relationship with the predictor is generally analyzed on a feature-by-feature basis. In Pandas, there are many related functions (describe, value_counts (), etc.) that can clearly show the relationship between the two.
- `Dimension reduction`: Use related algorithms to process the data set, rank the importance of features, and take the features with high importance, such as PCA.

Model Establishment. The establishment of a model refers to the various algorithms used in order to mine useful information, whether it is a traditional machine learning algorithm or a popular deep learning algorithm in recent years.[5] According to different learning methods, machine learning algorithms can be divided into: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Facing different data situations, different algorithms such as classification, regression, clustering, and association analysis can be used. At present, no single model can get good results, and all adopt the concept of ensemble learning to fuse models.

Field	*Type*	*Meaning*
id	string	paper ID
title	string	title
authors.name	string	author name
author.org	string	author unit
venue	string	Conference / Journal
year	int	year of publication
keywords	list of strings	keywords
abstract	string	abstract

Table 1: Data format for each paper

2.2 Dataset

Training Set. The training set has two files, train_author.json and train_pub.json.

- `train_author.json`: The data in this file is organized into a dictionary (recorded as dic1) and stored as a JSON object. The key of dic1 is the author's name. The value of dic1 is a dictionary representing the author (denoted as dic2). The key of dic2 is the author ID, and the value of dic2 is a list of the author's paper IDs.
- `train_pub.json`: This file contains metadata for all the papers in train_author.json, and the data is stored as JSON objects. The data in this file is represented as a dictionary, whose key is the paper ID, and its value is the corresponding paper information. The data format of each paper is as table1.

User profile already. The training set has two files, whole_auth or_profile.json and whole_author_profile_pub.json.

- `whole_author_profile.json`: Second-level dictionary, the key value is the author id, and the value is divided into two fields: the 'name' field represents the author name, and the 'papers' field represents the papers (author profile) owned by the author. Existing author files;
- `whole_author_profile_pub.json`: The meta-information involved in whole_author_profile.json, in the same format as train_pub.json;

Validation set. The training set has three files, cna_valid_unass _competition.json and valid_example_evaluation_continuous.json and cna_valid_pub.json.

- `cna_valid_unass_competition.json`: Paper list, which represents the list of papers to be assigned. The elements in the list are the paper id + '-' + the index of the author to be assigned (starting from 0); the contestant needs to assign the corresponding author of each paper in the file To an existing author profile (whole_author_profile.json).
- `valid_example_evaluation_continuous.json`: Sample submission file. The second-level dictionary, the key value is the author ID, and the value value represents the paper ID (from cna_valid_unass_competition.json) assigned to the author.
- `cna_valid_pub.json`: The meta-information of cna_valid_u nass_competition.json is in the same format as train_pub.json.

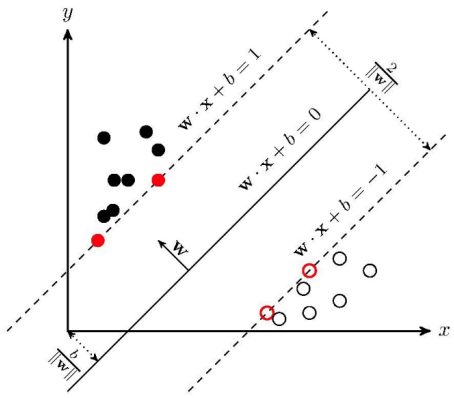


Figure 1: SVM hyperplane

3 TECHNIQUE

3.1 TF-IDF

Feature generation is particularly critical in data mining. In the disambiguation of the same name in the paper, TF-IDF algorithm is used to extract features.

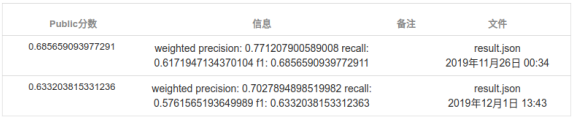
TF-IDF is a statistical method to evaluate the importance of a word to an article or an article in a corpus. The importance of a word increases proportionally with the number of times it appears in the file, but at the same time decreases inversely with the frequency of its appearance in the corpus.[6]

The main idea of TF-IDF is that if a word or phrase appears frequently in an article with a high TF and rarely appears in other articles, it is considered that the word or phrase has a good class discrimination ability and is suitable for To classify. TF-IDF is actually $TF * IDF$, where TF (Term Frequency) indicates how often a term appears in the document Document; IDF (Inverse Document Frequency), the main idea is that if there are fewer documents containing a word Word , The more differentiated the word is, the greater the IDF. For how to get the keywords of an article, we can calculate the TF-IDF of all nouns appearing in this article. The larger the TF-IDF, the higher the discrimination of this noun to this article. Take TF-IDF Larger words can be used as keywords in this article.

3.2 SVM

In machine learning, support vector machines (SVMs) are supervised learning models with related learning algorithms that analyze data for classification and regression analysis. Given a set of training examples, each example marked as belonging to one or the other of two categories, the SVM training algorithm builds a model and assigns the new example to one category or another, making it non-probabilistic binary linear Classifier. The SVM model is to represent examples as points in space, and the mapping is such that the examples of the individual categories are divided by the clearest gaps as wide as possible. The new examples are then mapped into the same space and predicted to belong to a category based on which edge they fall on.[7]

In addition to performing linear classification, SVM can also use so-called kernel tricks to efficiently perform non-linear classification,



Public分数	信息	备注	文件
0.685659093977291	weighted precision: 0.771207900589008 recall: 0.6171947134370104 f1: 0.6856590939772911		result.json 2019年11月26日 00:34
0.633203815331236	weighted precision: 0.7027894898519982 recall: 0.5761565193649989 f1: 0.6332038153312363		result.json 2019年12月1日 13:43

Figure 2: Submit score

implicitly mapping its input to a high-dimensional feature space.

4 SOLUTION

After previous analysis, the detailed steps of disambiguation of the same name of the paper are as follows:

Generate Training Data. First extract the author's paper information in the training set and the meta-information on the training set from train
_author.json and train_pub.json. Then filter the training set, and only take names with 5 or more authors of the same name as the training set. 500 training examples, one training example contains the paper, the real author of the paper, and 5 negative authors, the ratio of positive and negative examples is 1: 5.

Generate Feature. Feature extraction is particularly critical in data mining. Good features can better reflect the core properties of the extracted content. Here, we use the TF-IDF algorithm to extract features from the title, abstract, and keywords of the literature. After data cleaning, we generate all positive and negative features.

Training With SVM. After constructing positive and negative SVM examples, use SVC for training.

Load and Process Test Fata. Similar to generating training data, first extract the author's paper information in the training set and the meta-information on the training set from whole_author_profile.json and whole
_author_profile_pub.json, and load the set of papers to be assigned; then use the same feature extraction method Extract features; finally generate a list of candidate authors.

Use Trained Models to Predict Results. Use the trained SVM model to score all candidate authors for each paper to be assigned, and use the highest score as the prediction result.

5 EVALUATION

In daily submissions, the score is 0.68566; the final score is 0.63320.In the final test set, there are a lot of dirty data that are different from the daily training, resulting in a final result that is not as good as the daily performance.The submission score is shown in Figure 2.

REFERENCES

[1] Y. Zhang, F. Zhang, P. Yao, and J. Tang, "Name disambiguation in aminer: Clustering, maintenance, and human in the loop." in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1002–1011.

[2] X. Wang, J. Tang, H. Cheng, and S. Y. Philip, "Adana: Active name disambiguation," in *2011 IEEE 11th international conference on data mining*. IEEE, 2011, pp. 794–803.

[3] S. B. Roy, M. De Cock, V. Mandava, S. Savanna, B. Dalessandro, C. Perlich, W. Cukierski, and B. Hamner, "The microsoft academic search dataset and kdd cup 2013," in *Proceedings of the 2013 KDD cup 2013 workshop*. ACM, 2013, p. 1.

[4] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 990–998.

[5] F. Wang, J. Li, J. Tang, J. Zhang, and K. Wang, "Name disambiguation using atomic clusters," in *2008 The Ninth International Conference on Web-Age Information Management*. IEEE, 2008, pp. 357–364.

[6] T. Joachims, "A probabilistic analysis of the rocchio algorithm with tfidf for text categorization." Carnegie-mellon univ pittsburgh pa dept of computer science, Tech. Rep., 1996.

[7] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.