Continuous Name Disambiguation

Shaosong Shi
709501937@qq.com
School of Computer Science and Technology, Shandong University
Qingdao, Shandong

ABSTRACT

The online system adds a large number of papers every day. How to accurately and quickly distribute the papers to the existing author files in the system is the most urgent problem of the online academic system. So the abstract definition of the problem is: given a batch of new papers and the system's existing author papers, the ultimate goal is to allocate the new papers to the correct author file. The task in this project is to assign new papers. The method that is easy to think of here is to extract the information of previous authors and assign papers. The method I use here is to compare new papers with author files, extract traditional features such as each author's coauthor, author's institution, and conferences where the author's paper was published, and then use the SVM classifier to classify. The final classification is to extract the paper information in the test set and put it into the SVM model for prediction. After the score is completed, the author who has the highest score is selected to be considered the author of the paper.

1 INTRODUCTION

Author name disambiguation is a difficult problem in these automatic filing of papers. On the one hand, the author's names of the papers may have the name ones in the same institution; on the other hand, the author's signature in English papers may have multiple forms. All of these have brought high challenges to the distribution of papers [4]. So we are going to build on previous authors. Find relevant information between authors and their papers based on previous authors' published papers. For example, for an author, the possible characteristics of his paper may include: the same coauthors, because he may have long-term cooperation with an author, or a research institution, because the author will perform research work in a certain institution for a long time, and make certain keywords, for example, an author has been exploring a certain field, it is shown that his papers' key words often appears in the field of his research, and so on [1] . We use conditions such as these to extract features and set weights, and we can judge the author's paper information based on the features. The way to extract information is the LDA feature extraction model. The main role of this model is to give the theme of each document in the document set as a probability distribution. After analyzing some documents and extracting their themes, they can be classified according to the theme. The extracted features are our judgment factors.

In terms of classifier selection, the traditional SVM [6] classifier is selected here. SVM is a type of generalized linear classifier that performs binary classification of data according to supervised learning. The basic idea is to solve the separation hyperplane that can correctly divide the training data set and has the largest geometric interval. Because there is only one true author for a paper, the authors can be divided into positive authors and negative authors [7]. By inputting the extracted paper features into the SVM model, training can be performed to determine the relationship between an author and a paper. During the test, feature information is extracted from the papers in the test set, and the data is predicted and scored in the trained SVM model [5]. Finally, the score of the possible authors of a paper is obtained. The author with the highest score can be selected.

2 BACKGROUND

LDA

LDA is a topic model that can be used to extract topics from documents in a document set. The traditional method of judging the similarity of two documents is by looking at the number of words that appear together in the two documents, such as TF-IDF. This method does not take into account the semantic association behind the text. Little or no, but the two documents are similar. If two traditionally synonymous sentences are judged by traditional methods and the sentences contain different vocabularies, they are definitely not similar. Therefore, when judging the relevance of a document, the semantics of the document need to be considered [3]. The tool for semantic mining is the topic model, LDA is one of them A more effective model. In the topic model, a topic represents a concept, an aspect, and is represented as a series of related words, which is the conditional probability of these words. In terms of images, the topic is a bucket filled with words with a high probability of occurrence, and these words have a strong correlation with this topic. How can I generate a theme? How should we analyze the topic of the article? This is the problem to be solved by the topic model. First, you can use the generated model to look at both the document and the subject. The so-called generative model means that we think that each word of an article is obtained

through a process of "selecting a certain topic with a certain probability and selecting a certain word from this topic with a certain probability" [2]. Then, if we want to generate a document, we need to generate the probability of each word in the document, and finally generate a matrix. The topic model is trained through this matrix to learn a new matrix.

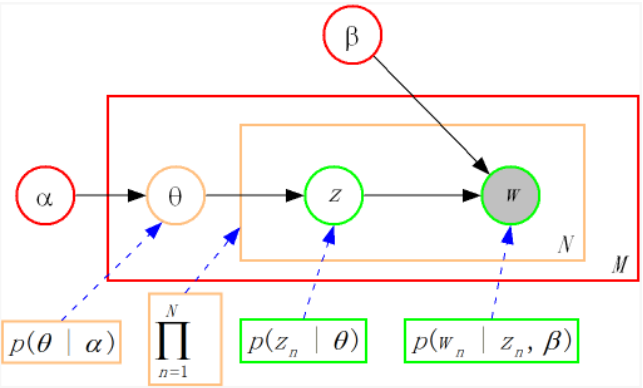

Figure 1: Model representation of LDA

In summary, the LDA model contains three layers of this item, theme and document.:

- In essence, LDA simply and rudely believes that each word in the article is obtained by "selecting a certain topic with a certain probability, and then selecting a word from the topic with a certain probability"
- A word may be associated with multiple topics, so you need to calculate the probability distribution in each case to determine which topic is most likely.
- An article may cover several topics, so you also need to calculate the probability of multiple topics.

SVM

Classification is a very important task in the field of data mining. Its purpose is to learn a classification function or classification model (also called a classifier). The support vector machine itself is a supervised learning method. In statistical classification and regression analysis. Support vector machine (SVM) is a machine learning method based on statistical learning theory developed in the mid-1990s [8]. It seeks to minimize the structural risk to improve the generalization ability of the learning machine, and to minimize the empirical risk and confidence range. In the case of a small number of statistical samples, the purpose of good statistical regularity can also be obtained. SVM: Support Vector Machine. Is a method based on classification boundaries. The basic principle is (using two-dimensional data as an example): if the training data is distributed over the points on a two-dimensional plane, they are clustered in different areas according to their classification. The goal

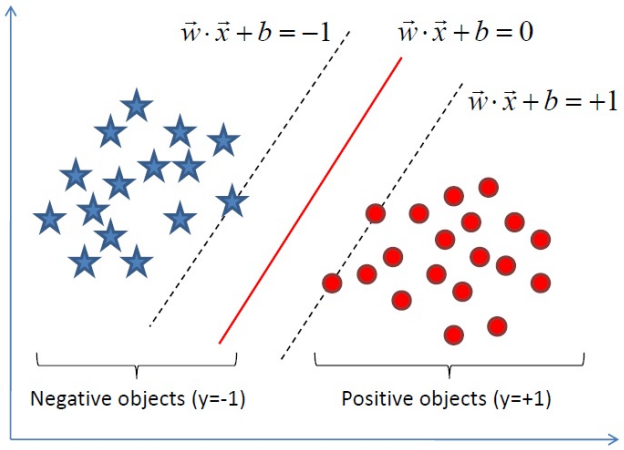

Figure 2: Model representation of SVM

of a classification algorithm based on classification boundaries is to find the boundaries between these classifications through training (linear-called linear partitioning, and curve-called non-linear partitioning). For multi-dimensional data (such as N-dimensional), they can be regarded as points in N-dimensional space, and the classification boundary is a surface in N-dimensional space, called a hypersurface (a hypersurface is one dimension less than N-dimensional space). The linear classifier uses hyperplane type boundaries, and the nonlinear classifier uses hypersurfaces. The principle of the support vector machine is to map the points in the low-dimensional space to the high-dimensional space so that they become linearly separable, and then use the principle of linear division to determine the classification boundary. In the high-dimensional space, it is a linear partition, but in the original data space, it is a non-linear partition. VMSVM shows many unique advantages in solving small sample, nonlinear and high-dimensional pattern recognition problems, and can be applied to other machine learning problems such as function fitting.

3 APPROACH

Generating training data

The training data used to train the SVM model should first be generated [9]. Here, a data set containing author information and meta-information is selected, and features for training are extracted from it. In order to improve the training efficiency and to simplify the process, only names with more than 5 people with the same name are selected as the training set. The training example is then sampled. Take 500 training examples as an example. Because a paper has only one real author, the true author of a paper is determined as the positive author, and five other authors with the same name are selected as negative authors. The ratio of positive

authors and negative authors is set to 1: 5. This completes the division of authors. Sample 10,000 papers as a training set to prepare for the next step of extracting the information characteristics of the paper.

Generating feature

After obtaining the author information, we would preprocess the data. Because the author name formats in different countries do not agree, they are unified into the name format in advance. Then, among all authors of the same name in a paper, find the author who actually wrote the paper. The next step is to generate feature information between the author and the paper. Here, the following dimensions of information are selected, including: co-authors of the authors, and the author's institution. Key words and abstracts of the paper, and LDA models are used to extract common topic words in the paper. After obtaining the above characteristics, they were added to the positive and negative authors, respectively. In the future training, the information between positive authors and essays is useful information to help us determine whether the paper belongs to a certain author. The relationship between negative authors and papers better helps us filter out authors who do not belong to the paper.

Training with SVM

Train the SVM model containing the author and the information of the paper. First, construct positive and negative examples of the SVM model, and generate x and y training sets at the same time. Add the positive author information generated in the previous step to the x training set, and add the negative author information to the y training set. Call the function for training. The parameters include: probability = True, C = 0.5, kernel = 'linear', decisionfunctionshape = 'ovo'

Loading and processing test data

The test data must be read in first, the data must be preprocessed, and the naming format of the paper must be the same. The paper features of the test set are then generated. This is basically similar to what was done in the second step. First, the author's co-author is extracted, and the author's institution. Key words, abstract, key words, etc [**?**]. Once generated, you can use the SVM model to make predictions. For each paper in the paper list, the SVM model grades each paper based on the feature information between the paper and the author; the papers are scored, and the scores are ranked, and the author with the highest score is selected as the author of the currently detected paper.

In this project,we use 2 datasets as train data, including train-author and train-paper, consisting of the information of the authors and the information of the papers.We use 4 datasetS as test data,including whole-author-profile and whole-author-profilepub and cna-test-unasscompetition and cna-test-pub ,consisting of the information of the information of the authors and the papers which are prepared to be distrubute. The result will generate as a json called result. The papers in the result have been distributed.There are authors with his papers.The final score is 0.633.

4 CONCLUSION

This project is all over so far. By giving a list of tens of thousands of papers and the identity information of the authors, we used the LDA model and other methods to find the relationship between the authors and the papers, and used SVM for training. The trained model was tested with the test set and got good results. The next improvement direction is to use another classifier or another algorithm to extract the key information of the paper and further optimize the use of parameters.

REFERENCES

[1] Ana Paula de Carvalho, Anderson Almeida Ferreira, Alberto Henrique Frade Laender, and Marcos André Gonçalves. 2011. Incremental unsupervised name disambiguation in cleaned digital libraries. (2011).

[2] Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 795–804.

[3] Matthew Van Gundy, Davide Balzarotti, and Giovanni Vigna. 2007. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies (WOOT '07)*. USENIX Association, Berkley, CA, Article 7, 9 pages.

[4] David J Hand. 2006. Data Mining. *Encyclopedia of Environmetrics* 2 (2006).

[5] Egoitz Laparra and German Rigau. 2009. Integrating wordnet and framenet using a knowledge-based word sense disambiguation algorithm. In *Proceedings of the International Conference RANLP-2009*. 208–213.

[6] Okumura Manabu and Honda Takeo. 1994. Word sense disambiguation and text segmentation based on lexical cohesion. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 755–761.

[7] Alan Filipe Santana, Marcos André Gonçalves, Alberto HF Laender, and Anderson A Ferreira. 2017. Incremental author name disambiguation by exploiting domain-specific heuristics. *Journal of the Association for Information Science and Technology* 68, 4 (2017), 931–945.

[8] SVM Vishwanathan and M Narasimha Murty. 2002. SSVM: a simple SVM algorithm. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, Vol. 3. IEEE, 2393–2398.

[9] Tong Zhang, Jue Wang, Liang Xu, and Ping Liu. 2006. Fall detection by wearable sensor and one-class SVM algorithm. In *Intelligent computing in signal processing and pattern recognition*. Springer, 858–863.