Continuous Name Disambiguation

Xiang Weixing

ABSTRACT

In recent years, due to the large number of new papers added to the online system every day, how to accurately and quickly distribute the papers to existing author files in the system is the most urgent problem for online academic systems. So the problem can be defined as: given a batch of new papers and a collection of existing author papers in the system, the ultimate goal is to allocate the new papers to the correct author file. The task in this project is to assign new papers according to their characteristics. The method we can easily think of is to extract the information of the previous authors, and then assign the papers. The method used in this paper is to compare newly added papers with author files, and then extract traditional features such as authors' authors, authors' institutions, conferences published by authors' papers, and then use SVM classifiers for classification. Then based on the classifier, the paper information in the test set is extracted, the extracted features are put into the SVM model to predict and score, and the higher-scoring candidate authors are selected as the authors of the paper.

1 INTRODUCTION

In the automatic archiving of papers, disambiguation of authors of the same name is a difficult problem. On the one hand, the author name of the paper may have the same name 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. Therefore, we need to find the relevant information between the authors and the papers they have published based on the authors' past papers. For example, for an author, his paper's characteristics may include: the same co-author, 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, it can also be Certain keywords, for example, an author is always exploring a certain field, so his paper often appears in the field of his research, and so on [?]. We can use conditions such as these to extract features and set corresponding weights, and we can judge the author's paper information based on the features. The way to extract information is the TextRank feature extraction model. When TextRank extracts keywords, it is more important than which page is selected in the webpage. In fact, we only need to find a way to construct the graph. The nodes of the graph are words, and the article is divided into sentences, and each sentence is divided into words, and the words are the nodes. So how is edge defined? Here, the idea of n-gram can be used. In simple terms, a word is only related to n words near it, that is, the node corresponding to n words near it is connected with an undirected edge (two directed edges). Then delete some types of part-of-speech words, delete some custom words, and retain only a part of the words. Only these words can be connected to each other. The classifier selected is a traditional SVM classifier. 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. 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. Finally, the score of the possible authors of a paper is obtained. The author with the highest score can be selected.

2 BACKGROUND

TextRank

TextRank is an algorithm used for keyword extraction. It can also be used to extract phrases and automatic summaries. Because TextRank is based on PageRank, first introduce the PageRank algorithm briefly. (1)PageRank PageRank was originally designed for Google's web page ranking, named after the company's founder, Larry Page. Google uses it to reflect the relevance and importance of web pages, and is often used to evaluate the effectiveness of web page optimization in search engine optimization operations. PageRank uses a hyperlink relationship in the Internet to determine the ranking of a web page, and its formula is designed by a voting idea: if we want to calculate the PageRank value (hereinafter referred to as the PR value) of page A, then we need to know what The webpage links to webpage A, that is, to get the inbound link of webpage A first, and then calculate the PR value of webpage A by voting on webpage A through the inbound chain. This design can guarantee to achieve such an effect: when some high-quality web pages point to web page A, then the PR value of web page A will increase due to these high-quality votes, and web page A is pointed to by less web pages When a page with a low PR value points to it, the PR value of A will not be very large, which can reasonably reflect the quality level of a web page. Based on the above ideas, Page designed the following formula: In the

$$PR(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j)$$

Figure 1

formula, Vi represents a web page, Vj represents a web page linked to Vi (that is, Vi's inbound link), S (Vi) represents a PR value of the web page Vi, and In (Vi) represents a set of all inbound links of the web page Vi Out (Vj) represents the web page, and d represents the damping coefficient. It is used to overcome the inherent shortcomings of the part after "d *" in this formula: if there is only the summation part, then the formula will not be able to process The PR value of the webpage, because at this time, the PR value of these webpages is 0 according to the formula, but the actual situation is not the case, so a damping coefficient is added to ensure that each webpage has a PR value greater than 0. According to the experimental As a result, with a damping coefficient of 0.85, the PR value can converge to a stable value after more than 100 iterations. When the damping coefficient is close to 1, the number of iterations required will suddenly increase a lot, and the ranking will be unstable. The score in front of S (Vj) in the formula refers to the fact that all the web pages pointed to by Vj should divide the PR value of Vj evenly, so that it can be considered that its own votes are assigned to the web pages it links to. (2)TextRank

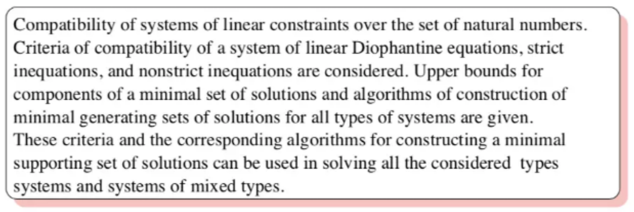
Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

Figure 2

Extract keywords TextRank is improved from PageRank, and its formula has many similarities. Here is the formula for TextRank: It can be seen that the formula only has one more weight term Wji than PageRank, which is used to indicate that the edge connection between two nodes has different degrees of importance. TextRank's algorithm for keyword extraction is as follows: (1) Segment the given text T according to the complete sentence; (2) For each sentence, perform word segmentation and part-of-speech tagging, and filter out stop words, leaving only words that specify the part of speech, such as nouns, verbs, adjectives; (3) Construct a candidate keyword graph G = (V, E), where V is the node set, composed of candidate keywords generated by (2), and

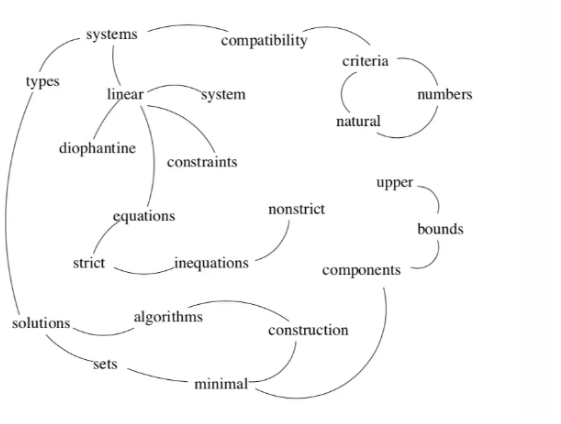

Figure 3

$$WS(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

Figure 4

then use co-occurrence to construct a relationship between any two points. An edge exists between two nodes only if their corresponding words co-occur in a window of length K, where K is the size of the window, that is, a maximum of K words are co-occurred. (4) According to the above formula, iteratively propagate the weights of each node until convergence. (5) Sort the node weights in reverse order to get the most important T words as candidate keywords. (6) get the most important T words from (5), mark them in the original text, and if they form adjacent phrases, combine them into multi-word keywords. For example, there is a sentence "Matlab code for plotting ambiguity function" in the text. If both "Matlab" and "code" are candidate keywords, they are combined into "Matlab code" to add the keyword sequence. 2.2 Support Vector Machine(SVM) SVM is a method based on classification boundaries. To understand svm, we need to understand what interval maximization is, and separate different sample spaces in linear binary classification. As shown in the figure below, we need to find a line to isolate samples of different classifications. The linear classifier is through this line, we can separate the samples of different categories. When a new sample comes, we can judge the part of this line to get the new sample category. As shown in the figure below, there are many separated lines that can classify the samples, as shown in the figure below L1, L2, L3. But how to choose an optimal line for segmentation? The principle of maximum separation is by choosing a midline that is as far away as possible from both samples. This is L2 in the figure below. The advantage is that it is far from the samples on both sides. Therefore, the situation of misjudgment is

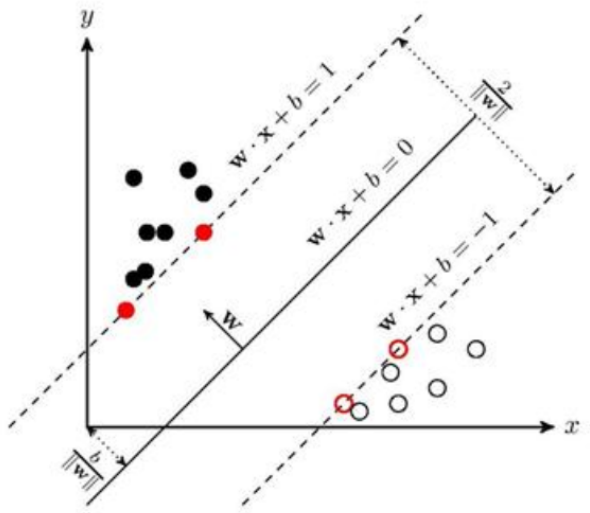

Figure 5

relatively small. The prediction is more accurate.So how to complete the selection of this interval maximum line. This part needs to be deduced by using rigorous mathematical formulas. Students who are interested in the process can view related materials. The conclusion is directly given here.

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2$$
$$s.t. \quad y_i(w \cdot x_i + b) - 1 \geq 0 \qquad i = 1,2,\ldots,N$$

Figure 6

By using the optimized processing method, we can get the method to obtain this optimal interval line.

2.2.1 Support vector After talking about the interval function for so long, the problem of the maximum interval, what is a support vector? As shown in the figure below, since the interval maximization needs to find the segmentation line with the largest interval in two different sample categories, the points of the samples with equal distances on both sides of the segmentation line are important. These points are the support vectors[4].

Since the maximum interval can be obtained by selecting the right support vector, during the algorithm iteration, only a few points need to be saved and updated in the memory, which will greatly save the memory space occupied by the algorithm, which is very important in actual engineering[1]. important.2.2.3 kernel function The algorithms of the support vector machines we have described so far are all linearly separable. But in actual engineering, the usage in many scenarios and environments is linear and inseparable. to deal with these problems. You need to find a way to separate the samples. So this part is the problem that the kernel function needs to consider.

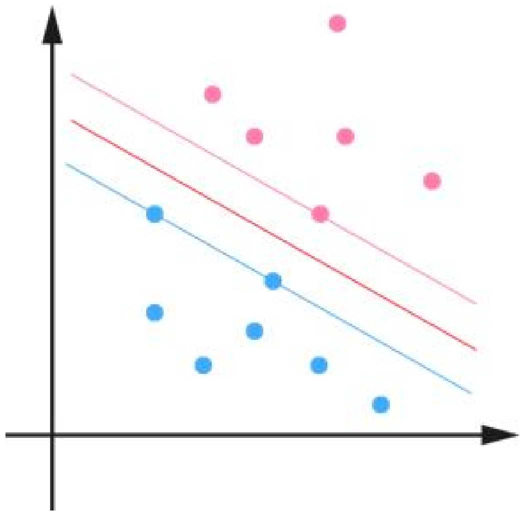

Figure 7: Model representation of SVM

SVM

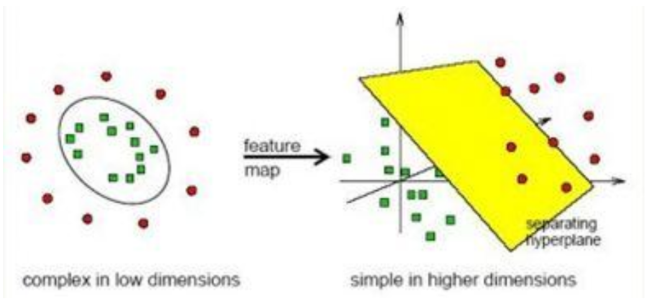

Figure 8: Model representation of SVM

As shown in the figure above, the so-called linear inseparable situation usually occurs when we consider the problem in two or low dimensions. If we introduce the sample space into a high-dimensional space, as shown on the right side of the figure above, it will be easier to find a hyperplane to segment the samples. Therefore, the kernel function solves the linear inseparability problem in low dimensions. At the same time, the mapping method is used instead of truly expanding to a high-dimensional space, and the complex calculations caused by full mapping to a high-dimensional are avoided. The support vector machine method is usually divided into the following steps 1). First, the kernel space method is used to transform the sample space to a linearly separable space. 2). Then use the method of maximizing the interval to obtain the dividing line with the largest interval. Then we get the support vector. 3). Finally, the segmentation line and support vector can be used to classify and predict new samples.

3 METHOD

Obtaining data for training the SVM model

In this project, 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[2]. 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 15,000 papers as a training set to prepare for the next step of extracting the information characteristics of the paper.

Data processing and feature extraction

After obtaining the author information, we 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. The keywords of the paper and the abstract of the paper, while using the TextRan model 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[3].

Start training with SVM model

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 two training sets, x and y. 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.

Paper Classification

For the test data read in, the data is preprocessed first, and the naming format of the paper is the same. The paper features of the test set are then generated. This is basically similar to what was done in the second step [5]. 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 scores each paper based on the feature information between the paper and the author, and sorts the scores. The candidate with the highest score is selected as the author of the currently detected paper.

4 CONCLUSION

In this project, we first clean the training data, then use TextRank to extract the features of the positive and negative authors' papers, and then add them to the SVM model for training. Finally, the treated papers use the trained svm model to score the authors of the same name. , Select the highest score as the author of the paper. The method scored 0.899 in the data to be predicted at the end[6]

REFERENCES

[1] David Harel. 1978. *LOGICS of Programs: AXIOMATICS and DESCRIPTIVE POWER.* MIT Research Lab Technical Report TR-200. Massachusetts Institute of Technology, Cambridge, MA.

[2] Donald E. Knuth. 1997. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.).* Addison Wesley Longman Publishing Co., Inc.

[3] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY.

[4] Charles J. Petrie. 1986. *New Algorithms for Dependency-Directed Backtracking (Master's thesis).* Technical Report. Austin, TX, USA.

[5] Poker-Edge.Com. 2006. Stats and Analysis. Retrieved June 7, 2006 from http://www.poker-edge.com/stats.php

[6] Asad Z. Spector. 1990. Achieving application requirements. In *Distributed Systems* (2nd. ed.), Sape Mullender (Ed.). ACM Press, New York, NY, 19–33. https://doi.org/10.1145/90417.90738