MULTI-LEVEL TEXT CLASSIFICATION METHOD BASED ON LATENT SEMANTIC ANALYSIS

Hongxia Shi, Guiyi Wei
College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, P. R. China

Yun Pan
College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, P. R. China

Keywords: Text classification, LSA, Vector space model.

Abstract: In multi-level text classification, all categories have level relation. The categories in the same layer have a certain generality. By applying LSA theory to multi-level text classification, the words' semantic relationship is better represented and their weight equations are adjusted so they are more reasonable. This method extends the traditional vector space model to LSA space model and consequent experiments got very good results.

1 INTRODUCTION

Along with the development of the Internet, users can find the web pages they want in the enormous database of web pages represented by the Internet, and many search engines are available to users, including the popular Yahoo, Google and Baidu etc. Typical search engines work through keyword inputs. However, pages retrieved in this manner usually include invalid links and irrelevant web pages. A good web page classification method is thus an urgent need in facilitating user searches.

There are many classification methods for web pages. Traditional method includes: Bayesian method, k-nearest neighbor (k-NN) method, and maximal average entropy method etc. All those methods are based on the vector space model (VSM). The category is determined by calculating the distance between vectors. However vector space cannot show the semantic relation between the vectors. In the literature of recent years, category system is mostly plane system. All the categories are parallel, have no relationships. Only few considered multi-level structure. But they mostly use the traditional text classification method. These methods cannot distinguish the categories that have close relationship. For example, router and exchanger are all network equipments. And the text containing "router" often contains "exchanger".

On the fact of it, this paper proposes a multi-level text classification method based on latent semantic analysis (LSA) theory. This method mainly makes 4 improvements: (1) we consider the part of speech when choosing the term; (2) we modify the traditional weight formula, and make it more reasonable; (3) we extend the traditional vector space to the LSA space, and calculate similarity in the LSA space; (4) we use multi-level tree structure to instead the traditional plane structure.

2 LATENT SEMANTIC ANALYSIS

Assume a term-document matrix D, which is a $r \times m$ matrix, where r is the number of terms and m the number of documents. D is denoted as $D=[d_{ij}]_{r \times m}$, where d_{ij} is the weight of the i_{th} term in the j_{th} document.

Shi H., Wei G. and Pan Y. (2007).
MULTI-LEVEL TEXT CLASSIFICATION METHOD BASED ON LATENT SEMANTIC ANALYSIS.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - SAIC*, pages 320-324
DOI: 10.5220/0002401703200324

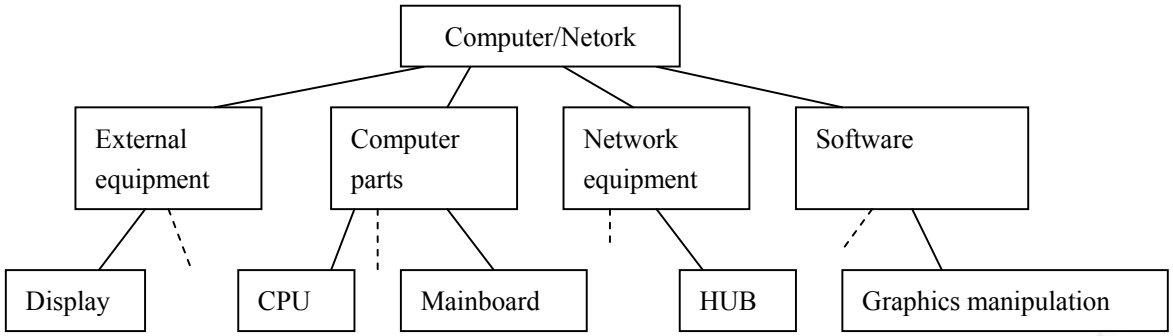


Figure 1: The tree structure of the computer and network.

The D of SVD (Singular value decomposition) is defined as $D=U\Lambda V^T$. Λ= diag($\lambda_1 \ldots \lambda_n$), where the elements of Λ are all singular values of D. Let $n = min\{r, m\}$, and the singular value is represented by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$. U and V are $r \times r$, $m \times m$ matrices, respectively. After processing by the SVD, $D=U\Lambda V^T$ simplifies to $D_k=U_k\Lambda_k V_k^T$. The dimensions of U_k, Λ_k, V_k^T are reduced to $r \times k$, $k \times k$, $k \times m$. The common element k is less than the original vector space. Λ_k retains k large singular value in term-document. U_k is a document vector, V_k^T is a term vector. The LSA theory not only eliminates disadvantages factors and extracts common semantic relations between terms and documents, but also decreases the dimension of vector by the singular value decomposition.

3 MULTI-LEVEL TEXT CLASSIFICATION BASED ON LSA

3.1 Classification Tree

We can construct a classification system according to some relationships of all the terms. In this paper, we use the tree structure to illustrate the classification system. Figure 1 shows the tree structure of the computer and network.

In Figure 1, software can be further divided into many kinds, such as game software, educational software, and application software etc. All the nodes in the same layer have some similarity. For example, CPU, main board, and CD-ROM etc. are components of computer. Thus our purpose is to divide the web text as exactly as possible into all sub-categories, which are all the nodes of the category tree.

3.2 Training Text and Term Selection

We first convert the web file into text file. Then we set the terms as the leaf nodes respectively. All the terms make up of our training data. We use the program to partition the training text according to their parts of speech. In order to reduce the dimension, we further remove some words from the training text. The removed words normally have little contributions, such as empty words etc. Therefore, the training text only contains noun, verb, adjective and adverb. We can divide verbs into three categories: relation verbs, state verbs, and action verbs. We finally delete the relation verbs and state verbs. The final terms are thus established.

3.3 Term Weight

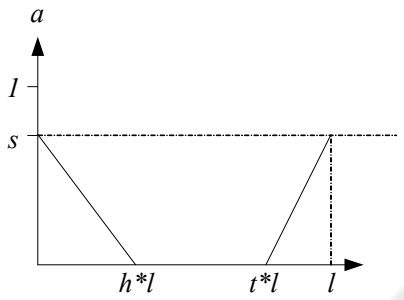
Vector space model (VSM) is used in the traditional text classification. The elements of those original vectors are 0 or 1. If a document contains some term, then the element in the corresponding position is 1, otherwise the corresponding element is 0. The VSM method cannot indicate how important the term in the document is. So term frequency is used to replace 0 or 1. The absolute term frequency is the number of occurrences of this word in the document. While the relative term frequency denotes the normalized term frequency. The relative term frequency is often determined by the term frequency–inverse document frequency (TF-IDF) formula. There are several TF-IDF formulas in the literature. One popular formula is as following.

$$W(t,\vec{d}) = \frac{tf(t,\vec{d}) \times \log(N/n_t + 0.01)}{\sqrt{\sum_{t \in \vec{d}} \left[tf(t,\vec{d}) \times \log(N/n_t + 0.01)\right]^2}}$$

Where $W(t,\vec{d})$ is the weight of term t in the document $\vec{d}$, $tf(t,\vec{d})$ is the frequency of term t in the document $\vec{d}$, N is the sum of all the training documents, and n_t is the number of the documents that contain the term t. The denominator is the normalization factor. However the position of the term in the document should be important to the weight. The terms in the title, starting part and the ending part may be more important. So we modify the weight as following.

$$W(t,\vec{d})=\frac{tf(t,\vec{d})\times\log(N/n_t+0.01)}{\sqrt{\sum_{t\in\vec{d}}[tf(t,\vec{d})\times\log(N/n_t+0.01)]^2}}(1+\alpha_{(p)}) \quad (1)$$

Where $\alpha_{(p)}$ is illustrated as the following figure.



From this figure, we obtain

$$\alpha(p)=\begin{cases}\frac{s}{h*l}p+s & when\quad p\le h*l\\ 0 & when\quad h*l<p<t*l\\ \frac{s}{l*(1-l)}p-\frac{st}{1-t} & when\quad t*l\le p\le l\end{cases}$$

where p is the position of the considered term in the document, l is the length of the considered document, s is the largest value of the term,$0<s<1$, h is the ratio of the starting part to the whole document, and t is the ratio of the ending part to the whole document.

3.4 The LSA Model

For the leaf nodes, we let the nodes that have the same father as a group. For each group, we construct a LSA model, where the weight matrix $D=[d_{ij}]_{r\times m}$ is determined by (1). For the non-leaf nodes, we also let the nodes that have the same father as a group. For each group, we first find its all leaf nodes, then construct corresponding LSA model according to these leaf nodes.

3.5 Web Text Classification

The main idea: For a given text, we compare it with the row vectors of V_k. The category of the text is determined as the category of the similar row vector. The formal algorithm is as following.

Step 1. Convert the web file into text file. (We can use the Spider program).

Step 2. Filter the text as Section 3.2. We get the final training terms. Let C be the root node.

Step 3. The son nodes of C construct a group as Section 3.4. Let the LSA model corresponding to this group is $D_k=U_k\Lambda_k V_k^T$. Suppose the considered text be X , which is a r-dimensional vector. The projection of X to D_k is denoted as $XX=X^TU_k\Lambda_k^{-1}$. Let $XX=(x_1,\ldots x_k)$, and $V=(v_1,\ldots v_k)$ be one row vector of V_k. Then the similarity is calculated by

$$sim(V,X)=\frac{\sum_{i-1}^{k}v_i x_i}{\sqrt{\sum_{i=1}^{k}v_i^{\,2}\sum_{i=1}^{k}x_i^{\,2}}}.$$

If the similarity of any row is less than the threshold that is given at first, then the considered text does not belong to any category. Stop. Otherwise, for each row whose similarity is not less than the threshold, we calculate the sum of the similarities between the considered text and all the training texts according to this row. The considered text is belong to the category whose sum is maximum. Therefore we can put this considered text on some node. Let C be this node and go to Step 4.

Step 4. If C is a leaf node, then stop. Otherwise, go to Step 3.

4 EXPERIMENTS

In the experiments, the classification system is similar to figure 1. The number of the training text on each leaf node is 100. The number of the whole tested text and the tested text on each leaf node is 720 and 50 respectively. The threshold is 0.32. The experimental results are listed in Table 1 and 2.

In Table 2, the recall ratio is the ratio of the final obtained texts to 50 (the tested texts on each leaf node). The correct $ratio=\frac{N_1}{N_2}$, where N_1 is the number of the texts belong to the corresponding category, and N_2 is the number of the final obtained

texts. The number of the external equipment in Table 1 and 2 is 154 and 145 respectively. The gap is 9. It shows that, for each text of the 9 texts, the similarity between it and each row of V_k is less than the threshold. From Table 1 and 2, we obtain that the average correct ratio of the high layer is larger than that of the low layer. For the case not using multi-level, we also have made an experiment. The results show that the average correct ratio is lower than that of the case using multi-level in Table 2.

5 CONCLUSIONS

The category system is determined subjectively. Different experimenter may propose different category system. A reasonable category system will improve the correct ratio. In future research, it will be worth considering the design of efficient and effective category system.

Table 1: The classification results based on the first layer.

	External equipment	Computer parts	Network equipment	Software
The final obtained texts	154	196	150	150
The texts belong to the corresponding category	147	193	150	150
The else texts	7	3	0	0
Recall ratio	0.98	0.97	1.00	1.00
Correct ratio	0.95	0.98	1.00	1.00

Table 2: The classification results based on the second layer.

		The final obtained texts	The texts belong to the corresponding category	The else texts	Recall ratio	Correct ratio
External equipment	Display	49	46	3	0.92	0.94
	Scanner	48	48	0	0.96	1.00
	Sound box	48	44	4	0.88	0.92
Computer parts	Hard disk	46	46	0	0.92	1.00
	CPU	49	49	0	0.98	1.00
	Audio device	45	43	2	0.86	0.96
	Display device	46	45	1	0.90	0.98
Network equipment	HUB	48	47	1	0.94	0.98
	Router	52	47	5	0.94	0.90
	Exchanger	46	43	3	0.86	0.93
Software	Graphics manipulation	47	47	0	0.94	1.00
	Operating system	47	46	1	0.92	0.98
	Office software	47	47	0	0.94	1.00

REFERENCES

Liu Q, Li S J. The calculation of semantic similarity between vocabularies based on “http://www.keenage.com”. http://www.nlp.org.cn/categories/default.php?cat_id=14

Zhou S G, Guan J H, Hu Y F. Latent semantic indexing (LSI) and its applications in applications in Chinese text processing [J] . Mini-Micro Systems, (2001), (2): 987—991

Huang H Y, Lin S M, Yan X W. A study of text classification based on concept space [J] Computer Science, (2003), 30(3): 46—49

Wang G Y, Xu J S. A new method of text categorization based on LSA and Kohonen network [J] Computer Applications, (2004), 24(2)

Schapire R, Singer Y . Boos Texter : a boosting-based system for text categorization . Machine Learning(2000), 39(2/3): 135—168

Campbell C ,Cristianin N ,Smo1a A .Query Learning with Large Margin Classifiers[A] . proceedings of the Seventeenth International Conference on Machine Learning[C] . (2000) . 111—118

Hsu C W ,Lin C J .A Compare of Methods for Multi-class Support Vector Machine [Z], (2001)