Text Classification Based on Partial Least Square Analysis

Xue-Qiang Zeng
School of Computer
Shanghai University
Shanghai 200072, China
Computer Center
Nanchang University
Nanchang 330006, China
xqzeng@ncu.edu.cn

Ming-Wen Wang
School of Computer Science and Technology
Jiangxi Normal University
Nanchang 330027, China
mwwang@jxnu.edu.cn

Jian-Yun Nie
DIRO, Université de Montréal
C.P. 6128, succursale Centre-ville
Montreal Quebec, H3C 3J7, Canada
nie@iro.umontreal.ca

ABSTRACT

Latent Semantic Indexing (LSI) is a favorite feature extraction method used in text classification. Since when important global features for all the classes can be determined by LSI, important local features for small classes may be ignored, this leads to poor performance on these small classes. To solve this problem, a novel method based on Partial Least Square (PLS) analysis is proposed by integrating class information into the latent classification structure. Important features are extracted according to both their descriptive power of document contents as in LSI, and their capacity of discriminating classes. The extracted features are applied to several classification algorithms: SVM, kNN, C4.5 and SMO. Experiments on Reuters prove that the features extracted by our method outperform those extracted by LSI in all the cases. In particular, the gain obtained by our method is the most apparent on small classes.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Indexing methods.*

General Terms

Algorithms, Theory

Keywords

Text Classification, Partial Least Square, Text Indexing, Dimensionality Reduction, Feature Extraction

1. INTRODUCTION

Text classification often suffers from the problem of high dimensionality of the texts. Therefore dimensionality reduction or feature selection/extraction is of great importance. Among various methods, Latent Semantic Indexing (LSI) turns out to be a successful feature extraction approach and is widely used in text classification and information retrieval (IR) [3, 4].

LSI finds a latent semantic space (linear mapping) from the input document space, while trying to preserve original data as much as possible. However, the latent space is created without consideration of the classes of the documents and thus may not always be the best one for the purpose of classification. Some important features for discriminating documents from different categories (classes) may be be removed or attributed with small importance.

In order to solve this problem, some previous studies [6, 8, 13] were proposed to create local LSI, each for a class. The principle is to create an LSI for each class by using the relevant documents in that class, on the one hand, and a set of similar documents not belonging to the class, on the other hand. By extracting features using LSI from these documents, important features of each class can be obtained. In this approach, a key problem is to select the negative documents not belonging to the class for the construction of LSI. The method used (e.g. in [8]) try to select a set of documents that are related to the same topics (local region) as the positive documents. This selection leads to an LSI that is able to extract important features common to these documents. However, on one hand, the selected features are only descriptive of these documents but not necessarily discriminative to the documents; one the other hand, it would be difficult to put features from different LSI into competition in order to determine the most appropriate class.

We believe that an appropriate approach is the one that integrates both global features and local features: the former allow us to represent common important features of the whole document collection, while the latter enables us to represent adequately specific features of classes (especially for rare classes). Therefore, in this study, we propose to extend the LSI approach by integrating class information into the feature extraction process. The best features are those that can well represent document contents, as well as class information. As we will show, this method naturally leads to an analysis similar to Partial Least Squares (PLS) [5, 16]. So, we will call our method Partial Least Squares Semantic Indexing (PLSSI). PLS is a technique that generalizes and combines features from principal component analysis and multiple regression [5, 16]. It is particularly useful for predicting a set of dependent variables from a (very) large set of independent variables (i.e. predictors).

In comparison with LSI, PLSSI considers not only document contents, but also class information during feature

SAC'07 March 11-15, 2007, Seoul, Korea

extraction. PLSSI allows us to capture important information for small classes that LSI cannot. In this study, we will carry our experiments on the Reuters-21578 data with serval classification methods such as SVM, kNN, C4.5 and SMO (Sequential Minimal Optimization). The experiments show that the features extracted by PLSSI outperform those extracted by LSI. Our analysis will reveal that PLSSI is particularly useful for extracting important features of small classes in comparison with LSI.

The paper is organized as follows. We describe our in section 2 in detail. Experiments on Reuter are described in section 3. Finally, some conclusions are given in section 4.

2. THE PLS METHOD

To describe our method, some notations are needed. Let X be an $N \times M$ matrix of N documents and M terms. The element x_{ij} is the weight of the term j in the document i, for example, the *ltc* weighting [2]. Let Y denote an $N \times L$ matrix of N documents and L classes. When the document i belongs to the class j, the element y_{ij} is 1; otherwise it is 0. In the following, lower-case bold letters denote vectors, and upper-case ones denote matrices. $\|\cdot\|$ denotes Frobenius norm for matrices and 2-norm for vectors.

2.1 General Principle

The general principle used to extend the LSI method can be illustrated as Figure 1. The main idea is integrate the matrix Y which corresponding to the classification information with the matrix X which corresponding to the document content. To do this, two series of latent variables t and u are used to encode X and Y respectively.

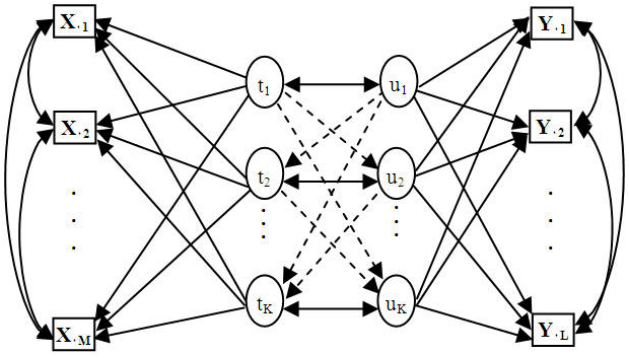

Figure 1: Diagram for PLS.

The key problem of our principle is to relate the two sets of variables. In fact, we try to determine the sets of variables in such a way that they represent matrices X and Y well, and at the same time, correlated with each other as strongly as possible. However, if we admit any relationship between them, the model would be extremely complex. In order to make the model tractable, we assume that pair of variables t_i and u_i are tied together. This is exactly the same idea as PLS. With respect to the Figure 1, this means that we only allow direct dependency between these variables. This is represented by the double arrows between t_i and u_i. However, there may exist indirect dependencies between different pairs of such variables, because the lower-level pairs of variables are intended to capture the remaining information that the higher-level pairs fail to capture. The indirect dependencies are represented by the dotted arrow in the Figure 1.

Following the diagram, we are now interested in the cross-covariance of X and Y. We wish to model the cross-covariance by K pairs of latent variables, i.e. (t_1,u_1), (t_2,u_2), ..., (t_K,u_K), with $K \ll M$. We assume that (t_i,u_i) are in the decreasing order of their importance to the matrices X and Y. That is, (t_1,u_1) captures the most important features, (t_2,u_2) captures the next most important ones, and so on.

The approach described above is not new. In fact, Partial Least Squares (PLS) method is a classical data regression model used in many fields [5, 15, 16]. It has been used for classification in several studies [9, 10]. However, to our knowledge, it has not been applied to text classification.

2.2 Mathematical formulation

As the first pair of latent variables (t_1,u_1) is the most discriminative, they have to meet the following requirements:

a) t_1 represent the information of X as well as possible;

b) u_1 represent the information of Y as well as possible;

c) (t_1,u_1) represent the correlation between X and Y as well as possible.

From a statistical point of view, a latent variable can represent the most information of a matrix if and only if the variance is maximal. Therefore, the condition a) is equivalent to require t_1 being a variable that has the maximal $Var(t_1)$, where $Var(\cdot)$ represents variance. Similarly, the condition b) is equivalent to maximize the $Var(u_1)$. The condition c) is equivalent to maximize the $r(t_1, u_1)$, where $r(\cdot,\cdot)$ represents the correlation coefficient between two stochastic variables.

The above maximization problems are very difficulty to solve directly. So the classical method is to weaken it. PLS combines the three maximization problems into the maximization of covariance $Cov(t_1, u_1) = \sqrt{(Var(t_1)Var(t_2))} \times r(t_1, u_1)$ [15]. The combined solution is not exactly the solutions to the three separate maximization problems, but is a reasonable approximation.

The latent variables t_1 and u_1 and can be considered as a linear combination of original matrices X and Y as expressed in Equation 1 and Equation 2.

$$t_1 = X\xi_1 \tag{1}$$

$$u_1 = Y\omega_1 \tag{2}$$

where ξ_1, ω_1 are the normalized projection vectors of t_1 and u_1 respectively. Then, the problem can be translated as Equation 3.

$$argmax(Cov(t_1, u_1)) \Leftrightarrow \underset{\|\xi_1\|=\|\omega_1\|=1}{argmax} ((X\xi_1)^T Y\omega_1) \tag{3}$$

where $\|\xi_1\|$ and $\|\omega_1\|$ represent the lengths of vectors ξ_1 and ω_1, and required to be unit length.

To solve Equation 3, the classical Lagrange Algorithm can be applied as Equation 4.

$$s = \xi_1^T X^T Y\omega_1 - \eta_1(\xi_1^T\xi_1 - 1) - \eta_2(\omega_1^T\omega_1 - 1) \tag{4}$$

where s, η_1 and η_2 are coefficients in Lagrange Algorithm.

The solution of Equation 4 is most of technical interest. There are many algorithms proposed to solve this problem. Interested readers can refer to [16] for more details.

After the first pair of latent variables (t_1,u_1) extracted, we can obtain the information represented by these variables by regressing the matrices X and Y on t_1. Then by subtracting the obtained information, we can go through the same process from the remaining matrices to determine (t_2,u_2) and other pairs of variables in turn.

2.3 PLSSI

By the process we described above, we can get a set of latent variables t_i , which are linear combinations of the terms in matrix X. Using these features, we create a latent space to represent documents. The difference between these features and those extracted from LSI is that the new features extracted by PLSSI also consider category information.

The projection matrix Ξ is formed by the projection vectors ξ_i , where $\Xi = (\xi_1, \ldots, \xi_K) \in \Re^{M \times K}$, and the matrix $\Xi^T\Xi$ is diagonal by the orthogonality of ξ_i [15]. Then, a document can be represented in the K-dimension PLSSI subspace by the matrix Ξ in a simple way as Equation 5.

$$\begin{aligned} T^T &= X^T\Xi \\ &= (X^T\xi_1, X^T\xi_2, \ldots, X^T\xi_K) \end{aligned} \tag{5}$$

where $T = (t_1, t_2, \ldots, t_K)^T$ is the representation of the original documents in the new K-dimension subspace, with $K \ll M$. Then the classification methods can then be used upon this space.

3. EXPERIMENTS

In order to test our notion, we carry our experiments to compare LSI and PLSSI on the Reuter collection.

3.1 Data sets and preparation

The Reuters-21578 collection is divided into a training set and a test set by the ModApte split, as in the previous studies [17]. By removing some corrupted documents, we obtain 7,770 training documents and 3,019 test documents. There are 135 different categories. The categories distribution is skewed; the most common category has a training set frequency of 2,877, but most of the categories have less than 100 instances. Furthermore, some categories have no training document. We remove them from our experiments, and only keep 90 categories which appear in both training and test sets. The above process is standard [14, 17].

We preprocess the data in a normative way: all numbers and stopwords are removed, words are converted into lowercase, and word stemming is performed using the Porter stemmer. This procedure results in 17,827 unique terms. Then in order to reduce some word noises (especially for spell error), we remove the terms which corresponding document frequency below 3. At the end, we get 6,883 unique terms.

We compute the weight vector using the *ltc* weighting [2],a form of $TF \times idf$ weighting. This gives term $X_{\cdot j}$ in document $X_{i\cdot}$ an weight of $x_{ij} = (1 + \log_e n(X_{i\cdot}, X_{\cdot j})) \times \log_e(N/n(X_{\cdot j}))$, where $n(X_{\cdot j})$ is the number of documents that contain $X_{\cdot j}$, $n(X_{i\cdot}, X_{\cdot j})$ is the number of occurrences of term $X_{\cdot j}$ in document $X_{i\cdot}$, and N is the total number of documents used in computing *idf* weights.

For the evaluation, we use the standard Macro$_{avg}$F1 and Micro$_{avg}$F1 measure [17]. Macro$_{avg}$F1 measure gives the same weight to all categories, and thus it will be equally influenced by the performance of rare categories. On the contrary, Micro$_{avg}$F1 measure will be dominated by the performance of common categories.

3.2 Experimental design and results

We compare the classification performance using features extracted by LSI and PLSSI respectively. Our goal is to evaluate whether PLSSI trained on the training set is able to derive high-quality features for new test documents and obtain good classification results. In order to avoid bias, we select four commonly used classification models: SVM^{light} [7], kNN [1], SMO [11] and C4.5 [12]. All these models have been applied with the same data preprocessing.

We compare the following cases in our experiments:

1. Term Features: Original Features (all 6,883 terms) are used by classification models, and this serves as the baseline for comparison. kNN and SVM^{light} are tested with this feature size. For the other two algorithms, the computation complexity makes it difficult to use the whole set of terms. Therefore, we select 500 best terms by the standard Chi statistic. These 500 terms are used in SMO and C4.5 as a substitute baseline case.

2. LSI: Standard unsupervised feature extraction is performed which maps the input data into a low-dimensional space. Then classification models are trained on this latent semantic subspace.

3. PLSSI: Additional category information for training set is used for obtaining a latent semantic subspace. Similarly, classification models are trained on this latent semantic subspace.

In order to examine how the classification performance varies with the dimension of latent variables increasing, we vary the dimension of semantic subspace from 10 to 500 for both LSI and PLSSI method.

Note that, the kNN model we used is the weighted kNN: the vote by the neighbors is weighted according to their distance to the given document. The parameter k is set to 100 for Original Features and 10 for other cases (optimized by experiments). As for SVM^{light}, we choose the linear version and use default model parameters of SVM^{light}.

We show the Macro$_{avg}$F1 and Micro$_{avg}$F1 results in Figure 2 and Figure 3 respectively. From figures, we can observe that PLSSI outperforms LSI in all the cases. Several further observations can be made:

1. Dimensionality reduction by LSI is not always effective for all the classification methods. From the above figures, compared to the original document space, LSI improves the Macro$_{avg}$F1 and Micro$_{avg}$F1 scores for SVM^{light}, but decreases these scores for kNN. Furthermore, for C4.5 model, the scores obtained with LSI are much worse than that with Chi statistic measure.

2. PLSSI method can increase Micro$_{avg}$F1 scores for all classification models. For example with SVM^{light}, the highest MicroavgF1 scores with the original features, LSI and PLSSI are 0.8627, 0.8808 and 0.8904 respectively. This Micro$_{avg}$F1 score with PLSSI is advantageously compared to the scores reported in the literature for the same collection [14, 17].

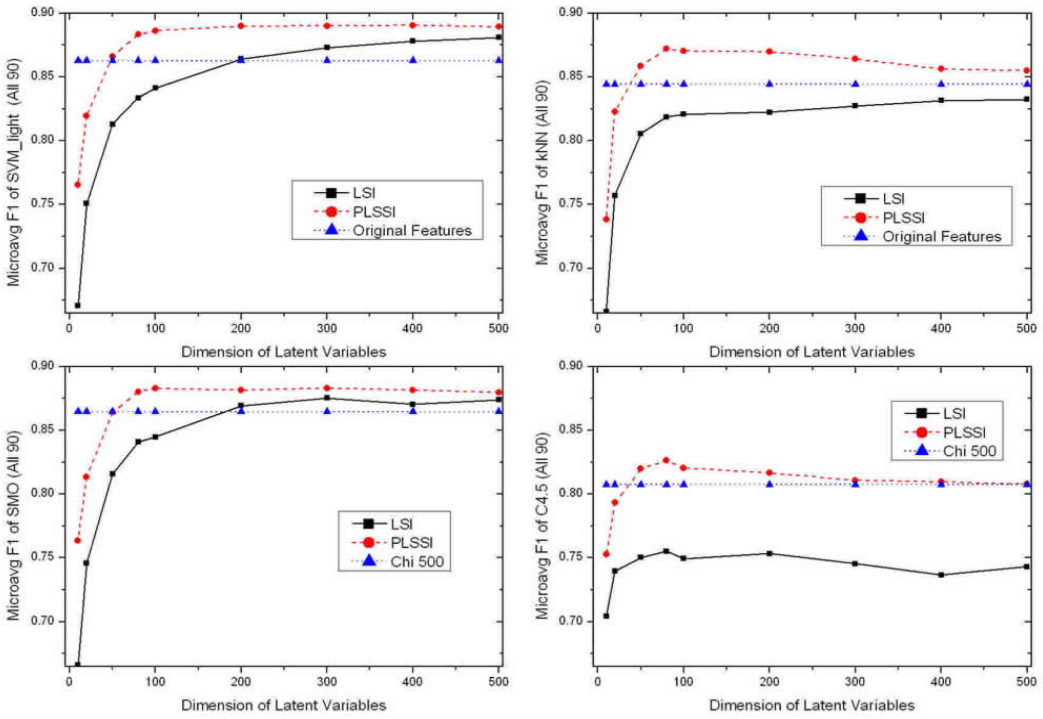

Figure 2: Micro$_{avg}$F1 vs. latent variables.

3. Macro$_{avg}$F1 scores have been improved substantially by PLSSI method for each classification model. As an unsupervised feature extraction method, LSI tries to obtain a small latent semantic subspace that can describe the original document space as much as possible. Due to the absence of category information, the latent semantic information for common and rare categories could not been treated equally. The semantic subspace obtained by LSI is dominated by the positive instances of common categories. In other words, the latent semantic information for rare categories will be omitted. In the empirical results, the LSI's Macro$_{avg}$F1 scores, which are influenced equally by common and rare categories, are always much lower than PLSSI. This comparison strongly supports our initial hypothesis that by incorporating class information into feature selection, the small classes can be better dealt with.

4. The optimized subspace for Micro$_{avg}$F1 scores with PLSSI has a very low dimensionality. In Figure 2, the best dimension of latent subspace varies from 80 to 100.

These empirical results confirmed that the latent semantic subspace obtained by PLSSI is superior to the one generated by LSI not only for common categories but also, and especially, for rare categories. So, compared to LSI, PLSSI can increase both Macro$_{avg}$F1 and Micro$_{avg}$F1 scores. However compared to LSI, the improvement of Micro$_{avg}$F1 score with PLSSI is relatively small. A possible reason is that the LSI's subspace can already represent well the latent information of common categories; so the room for further improvement is limited. For Macro$_{avg}$F1 score, we see that it is always very helpful to add category information into latent semantic space.

4. CONCLUDING REMARKS

In this paper we have described a feature extraction method named PLSSI for text classification. This method extends LSI by incorporating class information into the extraction process. The process is based on Partial Least Square Analysis. Instead of using only the document-term matrix as in LSI, PLSSI also uses the document-class matrix in our approach. The addition of this latter matrix makes it possible to take category information into account when determining the most important features. In PLSSI model, two sets of latent variables are used to capture respectively semantic indexing and classification information. The relationships between these variables denote the correspondence between latent semantic contents and the possible classes. In order to make the model practically tractable, PLSSI pair up the latent variables into (t_i,u_i). Then such pairs of variables are determined according to the principle of partial least square analysis. In comparison with the LSI approach, we determine pairs of latent variables (t_i,u_i) instead of single singular values.

From our experiments, this new feature extraction method has proven to be very effective: it produced generally better results than features extracted by LSI. By integrating the output information, PLSSI can indeed capture strong latent semantic variables related to categories. This results in higher Macro$_{avg}$F1 and Micro$_{avg}$F1 scores.

5. ACKNOWLEDGMENTS

This work was supported by NSFC(60663007, 20503015).

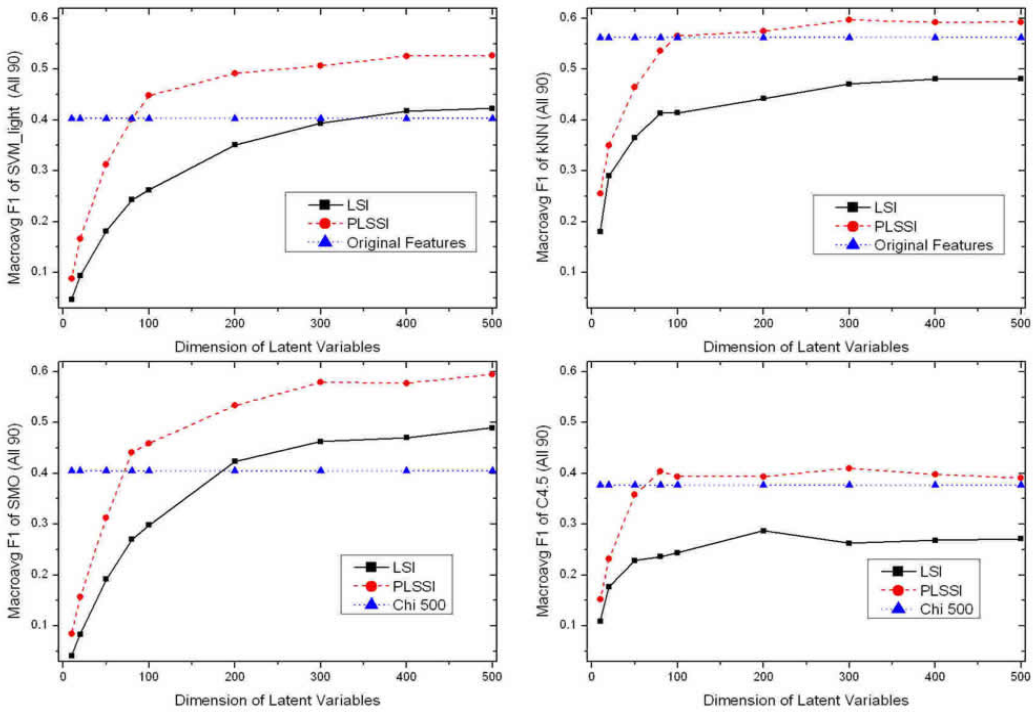

Figure 3: Macro$_{avg}$F1 vs. latent variables.

6. REFERENCES

[1] D. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[2] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using smart: Trec 3. In *the Third Text Retrieval Conference (TREC-3)*, 1994.

[3] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[4] S. Dumais. Using lsi for information filtering. In *the Third Text Retrieval Conference (TREC-3)*. National Institute of Standards and Technology Special Publication, 1995.

[5] I. Helland. On the structure of partial least squares regression. *Communications in statistics. Simulation and computation*, 17(22):581–607, 1988.

[6] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 282–289. Springer-Verlag, 1994.

[7] T. Joachims. *Making large-Scale SVM Learning Practical*. MIT Press, 1999.

[8] T. Liu, Z. Chen, B. Zhang, W.-Y. Ma, and G. Wu. Improving text classification using local latent semantic indexing. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 162–169. IEEE Computer Society, 2004.

[9] D. Nguyen and D. Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18(1):39–50, 2002.

[10] H. Nocairi, E. Qannari, E. Vigneau, and D. Bertrand. Discrimination on latent components with respect to patterns - application to multicollinear data. *Computational Statistics and Data Analysis*, 48(1):139–147, 2005.

[11] J. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. MIT Press, 1999.

[12] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[13] H. Schutze, D. Hull, and J. Pedersen. Comparison of classifiers and document representations for the routing problem. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 229–237. ACM Press, 1995.

[14] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Survey*, 34(1):1–47, 2002.

[15] H. Wold. Partial least squares. *Encyclopedia of Statistical Science*, 1985.

[16] S. Wold, M. Sjostrom, and L. Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(22):109–130, 2001.

[17] Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM Press, 1999.