Cross-Domain Classification: Trade-Off between Complexity and Accuracy

Elisabeth Lex, Christin Seifert, Michael Granitzer and Andreas Juffinger
Know-Center GmbH
elex,cseifert,mgrani,ajuffinger@know-center.at

Abstract

Text classification is one of the core applications in data mining due to the huge amount of not categorized digital data available. Training a text classifier generates a model that reflects the characteristics of the domain. However, if no training data is available, labeled data from a related but different domain might be exploited to perform cross-domain classification. In our work, we aim to accurately classify unlabeled blogs into commonly agreed newspaper categories using labeled data from the news domain. The labeled news and the unlabeled blog corpus are highly dynamic and hourly growing with a topic drift, so a trade-off between accuracy and performance is required. Our approach is to apply a fast novel centroid-based algorithm, the Class-Feature-Centroid Classifier (CFC), to perform efficient cross-domain classification. Experiments showed that this algorithm achieves a comparable accuracy than k-NN and is slightly better than Support Vector Machines (SVM), yet at linear time cost for training and classification. The benefit of this approach is that the linear time complexity enables us to efficiently generate an accurate classifier, reflecting the topic drift, several times per day on a huge dataset.

1. Introduction

Automatic classification of texts has become one of the core applications in data mining due to the huge amount of unstructured digital documents. Text classifiers are constructed by training characteristic features from already labeled documents[1]. However, training on a specific domain generates a classifier model that reflects the characteristics of exactly this domain. In context of text classification, especially the vocabulary is important and discriminant to characterize classes. Note that in the vector space model (VSM), each unique term reflects an input dimension or feature and therefore each model trained for this vector space is heavily dependent on the vocabulary or domain.

In cases without training data available for a domain a possible approach is to train on an akin domain which exhibits similar features and characteristics. However, it is not clear whether a trained model generated on one domain can be applied to another domain. Correctly applied machine learning algorithms avoid over-fitting on the training set to maximize the generalization capabilities. In case of cross-domain classification, an implicit fitting to the training corpus vocabulary is unavoidable. One approach to overcome this is to exploit the latent semantic structures in text and to create a more domain independent latent semantic feature space. Several approaches exploit these semantic relations between different domains to transfer knowledge from a source domain to a target domain. In [2], an algorithm based on Latent Semantic Analysis (LSA) [3] is proposed to exploit common topics and their semantic relations between domains. This knowledge then supports text classification in the target domain. The algorithm performs well, yet it is computationally complex. In [4], an algorithm to make the latent semantic relations between two domains explicit is proposed that uses a co-clustering approach. However, this method is also computationally complex.

In our work, we aim to classify blogs into commonly agreed upon newspaper categories. For blogs, there is usually little training data available, although blogs are often tagged. Yet, these tagging information is mostly subjective and not consistent across blogs [5], and most important, does not match our predefined newspaper categories. Besides, the tagging vocabulary of each individual blog is different and dynamically changing. Consequently, it is not possible to directly use individual blog tagging information, respectively folksonomies [1] in our setting. We would need to automatically learn a mapping from the evolving folksonomy to our categories which would result in another text classification problem. Besides, in our project, we already have high quality labeled data from a German news corpus [6]. The articles were carefully assigned to predefined newspaper categories by newspaper editors. On the other hand, our blog corpus is quite similar to the news domain [7], because in the project setting we are primarily interested in blogs highly correlated with news articles.

[1] http://www.iskoi.org/doc/folksonomies.htm

The analysis of the corpora revealed significant differences in the term distributions. Yet, we also identified that significant terms for a topic remain the same across both corpora which is in fact a crucial property for cross-domain text classification.

Given the labeled news articles and the unlabeled blog corpus, the question is: can we use this data to apply high quality cross-domain classification from news to blogs? Our corpora are highly dynamic and grow daily. Therefore, can the classification be performed fast and efficient?

In our approach we apply several text classification algorithms on the problem setting and evaluate the performance of these algorithms for different scenarios. We claim that the generalization abilities of text classification algorithms are sufficient when the classifiers implicitly concentrate only on the most important text features weighted with state-of-the-art techniques. For a visual evaluation, we use the classification visualization proposed in [8].

The remainder of this paper is structured as follows: Section 2 describes the classification task of our work. In Section 3, the classification visualization is depicted and its application to our classification task is explained. Section 4 describes the experimental settings and Section 5 results and discussions. Finally, we conclude the work in Section 6.

2. Classification Task

In our work, the classification task is a cross-domain multi-class problem with five classes. The corpus consists of two sub corpora: The first corpus, further referred to as news corpus, is collected from Austrian newspapers. This corpus was manually labeled (by the newspaper editors) in five newspaper categories: politics, economy, sports, culture and science. The second corpus, further referred to as blog corpus, was crawled from the World Wide Web.

In our project setting, we need a classification system with low computational complexity due to the highly dynamic data nature. Therefore, it is natural to use a centroid base text classifier. Centroid based classifiers are known to achieve good results in terms of accuracy and time complexity [9, 10]. In this work we implemented a novel centroid-based text classifier, the Class Feature Centroid (CFC), recently introduced in [11]. The algorithm is described in more detail in Section 2.1. The algorithm is extremely fast and is reported to outperform SVM and all other centroid based text classifiers.

To compare the performance of the cross-domain classification task, we evaluated CFC, and two standard text classifiers: Firstly, a k-Nearest Neighbor (k-NN) [12] and secondly a Support Vector Machine (SVM) [13] based on LibSVM [2] with a linear kernel. As outlined by Sebastiani [1], SVMs and k-NN Algorithms are the best performing standard text classification algorithms.

2.1. CFC Algorithm

The class-feature-centroid (CFC) classifier, proposed in [11], is a novel approach to centroid-based text classification. The advantage of centroid-based classifiers is clearly their computational efficiency and their simplicity [9]. However, their accuracy strongly depends on the quality of the centroids [10]. In the CFC algorithm, a novel centroid weight representation, given in Formula (1), is introduced which takes into account both the inter-class term distribution and the inner-class term distribution. Both are then combined to generate the term weights for the centroids.

$$w_{ij} = b^{\frac{DF^j_{t_i}}{|C_j|}} \times log(\frac{|C|}{CF_{t_i}}) \qquad (1)$$

With this weighting schema, highly discriminant centroids are derived. In addition, a denormalized cosine similarity is used to compute the distance of test instances to the class centroids. The denormalized cosine similarity computes the similarity between a document vector and a centroid vector using a standard cosine similarity whereas the centroids are not normalized. This preserves the discriminant capabilities of the centroids. In [11] the algorithm's performance is compared to a SVM and other centroid-based approaches on Reuters-21578 corpus and 20-newsgroup corpus. In these experiments, the algorithm outperforms both the SVM and other centroid-based approaches.

3. Classification Visualization

To study the confidence distribution of the classifiers we used the classification visualization proposed in [8]. For the classification visualization the outputs of all classifiers need to be an a-posteriori distribution over the classes. Our k-NN implementation outputs probabilities per se, for SVM and CFC, the classification outputs need to be mapped to a probability distribution, as described in [14].

The visualization shows the a-posteriori probabilities for all test items. The classes are represented by unique colored squares and are placed equally distributed around the perimeter of a circle. The classifier is learned on the training items and visualized on the test set. Also, accuracy and other common performance measures are calculated. Our goal is to evaluate a classifiers' performance and to get insights to the confidence distributions. With the performance measures we are able to compare different algorithms and

[2] http://www.csie.ntu.edu.tw/˜cjlin/libsvm

rank them in terms of accuracy. Additionally, the classification visualization provides us with the possibility to interactively identify problematic classes and items at a glance.

If labels are available for the test set, the misclassifications on the test set can be investigated. The items are represented by a green "+" for correctly classified and by a red "x" for misclassified. This visualization for our problem is depicted in Figure 2. The misclassification view can be used to indicate problem samples and to identify classes which are hard to separate.

4. Experimental Settings

For the visualization and the evaluation of our classification task, we split the datasets into a fixed training and test set. We randomly selected 70% of the data as training set and 30% as test set. To measure the performance of our classifiers we then evaluated the algorithms on the following four scenarios:

NN NewsNews: The training set of the news corpus has been used to train the classifiers and we report the performance on the news evaluation set.

BB BlogBlog: The training set of the blog corpus has been used to train the classifiers and we report the performance on the blog evaluation set.

NB NewsBlog: The training set of the news corpus has been used to train the classifiers and we report the performance on the blog evaluation set.

BN BlogNews: The training set of the blog corpus has been used to train the classifiers and we report the performance on the news evaluation set.

4.1. Dataset Properties

The news corpus with about 28k documents contains about 237k nouns with an average document length of 92.5 nouns. Each class of the news corpus contains nearly the same number of documents ($\sim$5600), so the corpus is balanced. The blog corpus consists of about 11k blog entries from 56 blogs which are selected according to the given newspaper sections: 10 politics blogs, 10 economy blogs, 10 sports blogs, 11 culture blogs, and 15 science blogs. In the blog corpus, the classes are represented by about 2800 politic blog entries, 2800 economy blog entries, 2400 sports blog entries, 1400 science blog entries, and 1100 culture blog entries. The blog entries were labeled with the class of the blog, and therefore all blog entries within a single blog belong to the same class. The blogs were selected and their blog entries were randomly checked whether they really belong to the class. Note that we did not examine all blog entries per blog therefore we cannot completely ensure that there is no mislabeled data. This naturally limits the theoretically achievable accuracy to less than 100%. The blog corpus contains about 110k nouns with an average document length of 61.5 nouns and a total noun token count of 675k. The merged corpora dictionary holds 302k different terms. Note that the sum of the distinct terms in the news and blogs dictionary is 347k, so consequently these two corpora share only 45k terms. We vectorized the data with an information extraction and vectorization module based on OpenNLP [3]. We used the Part-Of-Speech tags to construct our noun vector space. As a measure for the statistical difference between the news and blog dataset, we calculated the Kullback-Leibler divergence (KL) [15]. The Kullback-Leibler divergence, also known as relative entropy, is a measure of the difference between two probability distributions B and N. The KL divergence between two corpora (Blog B, News N) is calculated as

$$KL(B||N) = \sum_{t} \left[P_B(t) log \left(\frac{P_B(t)}{P_N(t)} \right) \right] \qquad (2)$$

whereby $P_B(t)$ states the probability of the term t in corpus B and $P_N(t)$ the probability in corpus N. The Kullback Leibler divergence for our cross domain corpora is shown in Table 1. The KL divergence 0.535 clearly reflects the statistical difference between the term statistics of these two corpora.

Table 1. Kullback Leibler divergence for Blog vs. News Corpus

	BlogNews	NewsBlog	Mean
KL Global	0.535	0.430	0.483

As mentioned above, we split the news and blog messages according to a 70/30 split into train (17k news, 6k blogs) and test set (7k news, 2.5k blogs). From this, we have then created the scenario datasets through permutation between news and blogs. Note that the number of documents per class is not necessarily equally distributed across the different sets (train/test) due to the purely random split procedure.

4.2. Parameter Settings

For a weighting schema, we used BM25 [16] for k-NN and SVM with the standard parameters $k = 2$ and $b = 0.75$. We also experimented with variants of TF-IDF for both algorithms, yet the algorithms performed best with BM25. For CFC, we used a standard TF-IDF weighting, as recommended by the authors. We also tested the algorithm with

[3] http://opennlp.sourceforge.net

BM25, but the results got worse than with TF-IDF, as expected. For the k-NN algorithm, we conducted a manual parameter search and identified $k = 10$ to be the best parameter setting. For the SVM, we used a linear kernel which is reported to outperform non-linear kernels in text classification [17]. We also experimented with various values for parameter b in CFC. However, different from findings in the original publication, where $b = e - 1.7$, we found that $b = e - 1.0$ performs best for our problem.

5. Results and Discussion

In our evaluations, we first computed the performance of all three classifiers on the mono-domain classification task (scenarios NN and BB). From these experiments, we determined the achievable performance for all classifiers for the cross-domain task. Our experiments revealed that the CFC achieves an accuracy value of 0.95 in the mono-domain task, equally good as the accuracy of the SVM. The k-NN performs slightly worse with an accuracy of 0.93. We achieved similar results for scenario BB, whereby k-NN, SVM, and CFC perform all with accuracy of 0.94. For scenario NB, the most important scenario for our work, unfortunately the performance of all three classifiers drops significantly. The k-NN algorithm is best with accuracy of 0.80. The CFC algorithm performs with accuracy of 0.78, and the SVM with accuracy of 0.76. Additionally, we reduced this scenario NB to a binary classification task, taking only news articles and blog entries from the classes “politics” and “sports”. The results for the binary task versus the five classes problem are shown in Table 2. Note, in this scenario, the accuracy dropped less from scenario NN to scenario NB due to the lower complexity of the binary task.

Table 2. Accuracy for NB with 2 and 5 Classes

	2	5	$t5_{train}$	$t5_{test}$	t_{total}
k-NN	0.85	0.80	∼7sec	∼166sec	∼173sec
SVM	0.82	0.76	∼900sec	∼24sec	∼924sec
CFC	0.84	0.78	∼10sec	∼2sec	∼12sec

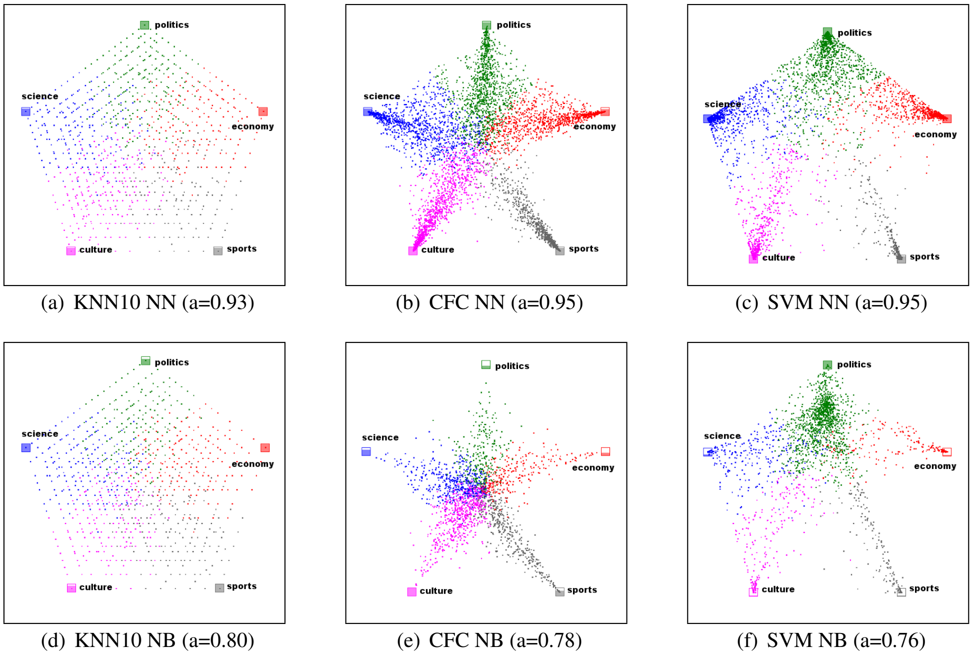
For completeness of our experiments we also evaluated scenario BN. The k-NN algorithm performs slightly better (accuracy of 0.83) than CFC with accuracy of 0.82. The SVM is worst with accuracy of 0.78. From the results of the conducted experiments, we can derive that in many cases, the k-NN algorithm works slightly better than the CFC and the SVM. However, when regarding the computation time, the CFC is by far the best (see Table 2). To get deeper insights into the algorithm’s decisions, we analyzed the class centroids of CFC. For this, we investigated the terms with weight $w > 0$ (remember, these terms are claimed to be the most discriminant terms). We calculated the KL divergence between the news and blog term distributions, but we only took centroid terms into account. As shown in Table 3, the KL divergence significantly decreased (by a factor 4 on average). This reveals that the CFC selects those terms which are characteristic for a class and these remain the same across both corpora.

Table 3. KL divergence for CFC centroids

	BlogNews	NewsBlog	Mean
KL Local C_1	0.048	0.065	0.056
KL Local C_2	0.127	0.194	0.160
KL Local C_3	0.216	0.117	0.166
KL Local C_4	0.081	0.067	0.074
KL Local C_5	0.123	0.127	0.125
KL Local Ø	0.119	0.114	0.116

We visually analyzed the classification results for scenario NB with our visualization tool. The results of the visualization tool for scenario NN are depicted in Figure 1. The visualization shows that the k-NN algorithm exhibits a discrete probability distribution. This can be derived from the fact that between two classes a maximum of 11 discrete confidence steps can occur. This also holds for the multi-class assignment. However, an interesting artifact of our dataset is that the k-NN has difficulties to make a clear decision between the classes “science” and “culture” as well as “economy” and “sports”.

From the visualization, it can be derived that the SVM has a clear tendency towards the category “politics”. Also, the visualization gives the impression that the SVM analyses a set of binary classification problems since the test items are placed along the connecting lines between all classes. However, we need to investigate this artifact in more detail. The CFC algorithm exhibits a relatively balanced distribution of the test items. Comparing the image of SVM and CFC, the CFC places less test items on the outer boundaries, between the categories “politics” and “economy”, as well as “politics” and “science”. The reason for this is that the centroid vectors overlap to a certain extent. The visualization also reveals that the CFC does not prefer any class, in contrast to SVM (“politics”). This better reflects the a-priori probabilities that we train on an equally distributed corpus. For the NB scenario, all classifiers exhibit a very similar visual distribution. That is why we expect that the algorithms perform similar on the cross-domain task. However, the correctness of the algorithms’ decisions can only be verified when investigating the mis-classifications (as depicted in Figure 2). The mis-classification of k-NN is equally distributed as one can see in Figure 2(a). The SVM has several mis-classifications in category “science” with confidence nearly 1. This indicates



(a) KNN10 NN (a=0.93) (b) CFC NN (a=0.95) (c) SVM NN (a=0.95)

(d) KNN10 NB (a=0.80) (e) CFC NB (a=0.78) (f) SVM NB (a=0.76)

Figure 1. Visualization for Test Item Confidence for k-NN, CFC, and SVM

that some of the support vectors cannot be generalized from the news domain to the blog domain. The visual impression is reflected in the accuracy result. In contrary, the CFC has nearly no misclassification very close to the classes whereas the really misclassified items are placed more in the center of the visualization (region of low confidence). In reference to our dataset, we conclude that the CFC confidences are more trustworthy than the SVM and k-NN confidence values for the cross-domain task.

6. Conclusions

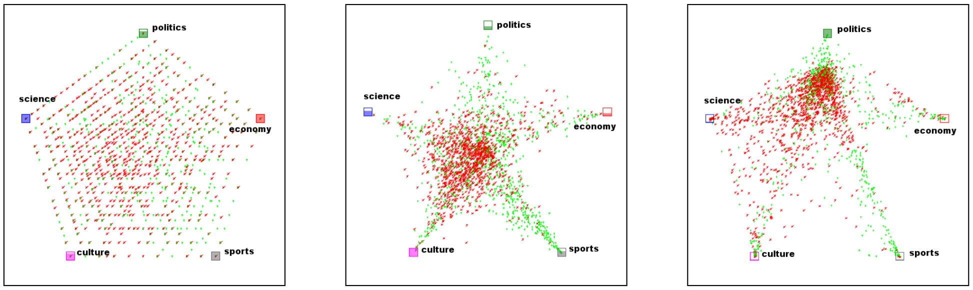
In this publication, we applied and evaluated a novel linear time classifier (CFC) for a cross-domain classification task. Our experiments showed that this classifier performs comparably with SVM and k-NN while being remarkably faster. Also, in terms of memory complexity the CFC definitely outperforms SVM and k-NN. In our setting, the CFC only stores the five centroid vectors, the SVM has to store about 1000 support vectors, and the k-NN the full training set (17k for scenario NN and NB). We also identified that in our setting the CFC model is more general and therefore better applicable than SVM for the cross-domain task. Besides, the experiments revealed that the accuracy drops less for CFC from the mono-domain task to the cross-domain task. This also emphasizes that the CFC model generalizes better. The visualization shows that the confidence distribution for CFC is more trustworthy than the SVM and k-NN confidence values. Both SVM and k-NN mis-classify items to wrong classes with high confidences, significantly more often than CFC. Furthermore, the CFC rather assigns low confidence values to misclassified items which clearly reflects the decision uncertainty for such items. To sum up, in our case, the CFC is the is the optimal trade-off between accuracy and performance. In the future, we want to analyze whether incremental algorithms are applicable for our problem setting. To provide other researchers the possibility to reproduce our results and in order to establish a common dataset for cross-domain classification, we will publish our datasets online.

7. Acknowledgement

APOSDLE is partially funded under the FP6 of the European Commission within the IST Workprogramme (project number 027023). The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

8. References

[1] Fabrizio Sebastiani, "Machine learning in automated text categorization", *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.



(a) KNN NB (a=0.80)

(b) CFC NB (a=0.78)

(c) SVM NB (a=0.76)

Figure 2. Misclassification Visualization for k-NN, CFC, and SVM

[2] Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu, "Topic-bridged PLSA for cross-domain text classification", in *Proc. Int. ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, New York, NY, USA, 2008, pp. 627–634, ACM.

[3] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman, "Indexing by latent semantic analysis", *Journal of the American Society for Inf. Science*, vol. 41, 1990.

[4] Pu Wang, Carlotta Domeniconi, and Jian Hu, "Using Wikipedia for co-clustering based cross-domain text classification", in *Proc. IEEE Int. Conference on Data Mining (ICDIM)*, Washington, DC, USA, 2008, IEEE Computer Society.

[5] Umer Farooq, Thomas G. Kannampallil, Yang Song, Craig H. Ganoe, John M. Carroll, and Lee Giles, "Evaluating tagging behavior in social bookmarking systems: metrics and design heuristics", in *Proc. Int. ACM Conf. on Supporting Group Work (GROUP)*, New York, NY, USA, 2007, pp. 351–360, ACM.

[6] Elisabeth Lex, Christin Seifert, Wolfgang Kienreich, and Michael Granitzer, "A generic framework for visualizing the news article domain and its application to real-world data", *Journal of Digital Information Management*, vol. 6, no. 6, pp. 434–442, 2008.

[7] Andreas Juffinger, Michael Granitzer, and Elisabeth Lex, "Blog credibility ranking by exploiting verified content", in *Proc. Workshop on Information Credibility on the Web (WICOW)*, New York, NY, USA, 2009, pp. 51–58, ACM.

[8] Christin Seifert and Elisabeth Lex, "A Novel Visualization Approach for Data-Mining-Related Classification", in *Proc. Int. Conference on Information Visualization (IV)*, 2009, to appear.

[9] Eui-Hong Han and George Karypis, "Centroid-based document classification: Analysis and experimental results", in *Proc. European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD)*, London, UK, 2000, pp. 424–431, Springer-Verlag.

[10] Verayuth Lertnattee and Thanaruk Theeramunkong, "Effect of term distributions on centroid-based text categorization", *Information Sciences, Informatics and Computer Science*, vol. 158, no. 1, 2004.

[11] Hu Guan, Jingyu Zhou, and Minyi Guo, "A class-feature-centroid classifier for text categorization", in *Proc. Int. Conf. on World Wide Web (WWW)*, New York, NY, USA, 2009, ACM.

[12] David W. Aha, Dennis Kibler, and Marc K. Albert, "Instance-based learning algorithms", *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

[13] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[14] John C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", in *Advances in Large Margin Classifiers*. 1999, MIT Press.

[15] S. Kullback and R. Leibler, "On information and sufficiency", *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.

[16] K. Sparck Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments", *Inf. Process. Manage.*, vol. 36, no. 6, 2000.

[17] Yiming Yang and Xin Liu, "A re-examination of text categorization methods", in *Proc. Int. ACM Conf. on Research and Development in Information Retrieval(SIGIR)*, New York, NY, USA, 1999, pp. 42–49, ACM.