Enriching Cold Start Personalized Language Model Using Social Network Information

Yu-Yang Huang[†], **Rui Yan**[*], **Tsung-Ting Kuo**[‡], **Shou-De Lin**[†‡]
[†]Graduate Institute of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
[‡]Graduate Institute of Network and Multimedia,
National Taiwan University, Taipei, Taiwan
[*]Computer and Information Science Department,
University of Pennsylvania, Philadelphia, PA 19104, U.S.A.
{r02922050, d97944007, sdlin}@csie.ntu.edu.tw, ruiyan@seas.upenn.edu

Abstract

We introduce a generalized framework to enrich the personalized language models for cold start users. The cold start problem is solved with content written by friends on social network services. Our framework consists of a mixture language model, whose mixture weights are estimated with a factor graph. The factor graph is used to incorporate prior knowledge and heuristics to identify the most appropriate weights. The intrinsic and extrinsic experiments show significant improvement on cold start users.

1 Introduction

Personalized language models (PLM) on social network services are useful in many aspects (Xue et al., 2009; Wen et al., 2012; Clements, 2007), For instance, if the authorship of a document is in doubt, a PLM may be used as a generative model to identify it. In this sense, a PLM serves as a proxy of one's writing style. Furthermore, PLMs can improve the quality of information retrieval and content-based recommendation systems, where documents or topics can be recommended based on the generative probabilities.

However, it is challenging to build a PLM for users who just entered the system, and whose content is thus insufficient to characterize them. These are called *"cold start"* users. Producing better recommendations is even more critical for cold start users to make them continue to use the system. Therefore, this paper focuses on how to overcome the cold start problem and obtain a better PLM for cold start users.

The content written by friends on a social network service, such as Facebook or Twitter, is exploited. It can be either a reply to an original post or posts by friends. Here the hypothesis is that friends, who usually share common interests, tend to discuss similar topics and use similar words than non-friends. In other words, we believe that a cold start user's language model can be *enriched* and better *personalized* by incorporating content written by friends.

Intuitively, a linear combination of document-level language models can be used to incorporate content written by friends. However, it should be noticed that some documents are more relevant than others, and should be weighted higher. To obtain better weights, some simple heuristics could be exploited. For example, we can measure the similarity or distance between a user language model and a document language model. In addition, documents that are shared frequently in a social network are usually considered to be more influential, and could contribute more to the language model. More complex heuristics can also be derived. For instance, if two documents are posted by the same person, their weights should be more similar. The main challenge lies in how such heuristics can be utilized in a systematic manner to infer the weights of each document-level language model.

In this paper, we exploit the information on social network services in two ways. First, we impose the *social dependency* assumption via a finite mixture model. We model the true, albeit unknown, personalized language model as a combination of a biased user language model and a set of relevant document language models. Due to the noise inevitably contained in social media content, instead of using all available documents, we argue that by properly specifying the set of relevant documents, a better personalized language model can be learnt. In other words, each user language model is enriched by a *personalized* collection of background documents.

Second, we propose a factor graph model (FGM) to incorporate prior knowledge (e.g. the heuristics described above) into our model. Each mixture weight is represented by a random variable in the factor graph, and an efficient algorithm is proposed to optimize the model and infer the marginal distribution of these variables. Useful information about these variables is encoded by a set of potential functions.

The main contributions of this work are summarized below:

- To solve the cold start problem encountered when estimating PLMs, a generalized framework based on FGM is proposed. We incorporate social network information into user language models through the use of FGM. An iterative optimization procedure utilizing perplexity is presented to learn the parameters. To our knowledge, this is the first proposal to use FGM to enrich language models.
- Perplexity is selected as an intrinsic evaluation, and experiment on authorship attribution is used as an extrinsic evaluation. The results show that our model yields significant improvements for cold start users.

2 Methodology

2.1 Social-Driven Personalized Language Model

The language model of a collection of documents can be estimated by normalizing the counts of words in the entire collection (Zhai, 2008). To build a user language model, one naïve way is to first normalize word frequency $c(w,d)$ within each document, and then average over all the documents in a user's document collection. The resulting unigram user language model is:

$$\begin{aligned} P_u(w) &= \frac{1}{|\mathcal{D}_u|}\sum\nolimits_{d\in\mathcal{D}_u}\frac{c(w,d)}{|d|} \\ &= \frac{1}{|\mathcal{D}_u|}\sum\nolimits_{d\in\mathcal{D}_u} P_d(w) \end{aligned} \tag{1}$$

where $P_d(w)$ is the language model of a particular document, and $\mathcal{D}_u$ is the user's document collection. This formulation is basically an equal-weighted finite mixture model.

A simple yet effective way to smooth a language model is to linearly interpolate with a background language model (Chen and Goodman, 1996; Zhai and Lafferty, 2001). In the linear interpolation method, all background documents are treated equally. The entire document collection is added to the user language model $P_u(w)$ with the same interpolation coefficient.

Our main idea is to specify a set of relevant documents for the target user using information embedded in a social network, and enrich the smoothing procedure with these documents. Let $\mathcal{D}_{rel}$ denote the content from relevant persons (e.g. social neighbors) of u_1, our idea can be concisely expressed as:

$$P'_{u_1}(w) = \lambda_{u_1}P_{u_1}(w) + \sum_{d_i\in\mathcal{D}_{rel}} \lambda_{d_i}P_{d_i}(w) \tag{2}$$

where λ_{d_i} is the mixture weight of the language model of document d_i, and $\lambda_{u_1} + \sum\lambda_{d_i} = 1$. Documents posted by irrelevant users are not included as we believe the user language model can be better personalized by exploiting the social relationship in a more structured way. In our experiment, we choose the first degree neighbor documents as $\mathcal{D}_{rel}$.

Also note that we have made no assumption about how the "base" user language model $P_{u_1}(w)$ is built. In practice, it need not be models following maximum likelihood estimation, but any language model can be integrated into our framework to achieve a better refined model. Furthermore, any smoothing method can be applied to the language model without degrading the effectiveness.

2.2 Factor Graph Model (FGM)

Now we discuss how the mixture weights can be estimated. We introduce a *factor graph model* (FGM) to make use of the diverse information on a social network. Factor graph (Kschischang et al., 2006) is a bipartite graph consisting of a set of random variables and a set of factors which signifies the relationships among the variables. It is best suited in situations where the data is clearly of a relational nature (Wang et al., 2012). The joint distribution of the variables is factored according to the graph structure. Using FGM, one can incorporate the knowledge into the potential function for optimization and perform joint inference over documents. As shown in Figure 1, the variables included in the model are described as follows:

Candidate variables $y_i = \langle u, d_i\rangle$**.** The random variables in the top layer stand for the degrees of belief that a document d_i should be included in the PLM of the target user u.

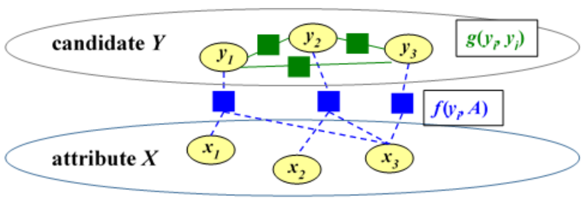

Figure 1: A two-layered factor graph (FGM) proposed to estimate the mixture weights.

Attribute variables x_i**.** Local information is stored as the random variables in the bottom layer. For example, x_1 might represent the number of common friends between the author of a document d_i and our target user.

The potential functions in the FGM are:

Attribute-to-candidate function. This potential function captures the local dependencies of a candidate variable to the relevant attributes. Let the candidate variable y_i correspond to a document d_i, the attribute-to-candidate function of y_i is defined in a log-linear form:

$$f(y_i, A) = \frac{1}{Z_\alpha} exp\{\alpha^T \mathbf{f}(y_i, A)\} \quad (3)$$

where A is the set of attributes of either the document d_i or target user u; **f** is a vector of feature functions which locally model the value of y_i with attributes in A; Z_α is the local partition function and α is the weight vector to be learnt.

In our experiment, we define the vector of functions as $\mathbf{f} = \langle f_{sim}, f_{oov}, f_{pop}, f_{cmf}, f_{af}\rangle^T$ as:

- **Similarity function** f_{sim}**.** The similarity between language models of the target user and a document should play an important role. We use cosine similarity between two unigram models in our experiments.
- **Document quality function** f_{oov}**.** The out-of-vocabulary (OOV) ratio is used to measure the quality of a document. It is defined as

$$f_{oov} = 1 - \frac{|\{w: w \in d_i \cap w \notin V\}|}{|d_i|} \quad (4)$$

  where V is the vocabulary set of the entire corpus, with stop words excluded.
- **Document popularity function** f_{pop}**.** This function is defined as the number of times d_i is shared to model the popularity of documents.
- **Common friend function** f_{cmf}**.** It is defined as the number of common friends between the target user u_1 and the author of d_i.
- **Author friendship function** f_{af}**.** Assuming that documents posted by a user with more friends are more influential, this function is defined as the number of friends of d_i's author.

Candidate-to-candidate function. This potential function defines the correlation of a candidate variable y_i with another candidate variable y_j in the factor graph. The function is defined as

$$g(y_i, y_j) = \frac{1}{Z_{ij,\beta}} exp\{\beta^T \mathbf{g}(y_i, y_j)\} \quad (5)$$

where **g** is a vector of feature functions indicating whether two variables are correlated. If we further denote the set of all related variables as $G(y_i)$, then for any candidate variable y_i, we have the following brief expression:

$$g(y_i, G(y_i)) = \prod_{y_j \in G(y_i)} g(y_i, y_j) \quad (6)$$

For two candidate variables, let the corresponding document be d_i and d_j, respectively, we define the vector $\mathbf{g} = \langle g_{rel}, g_{cat}\rangle^T$ as:

- **User relationship function** g_{rel}**.** We assume that two candidate variables have higher dependency if they represent documents of the same author or the two authors are friends. The dependency should be even greater if two documents are similar. Let $a(d)$ denote the author of a document d and $\mathcal{N}[u]$ denote the closed neighborhood of a user u, we define

$$g_{rel} = \mathbb{I}\{a(d_j) \in \mathcal{N}[a(d_i)]\} \times sim(d_i, d_j) \quad (7)$$

- **Co-category function** g_{cat}**.** For any two candidate variables, it is intuitive that the two variables would have a higher correlation if d_i and d_j are of the same category. Let $c(d)$ denote the category of document d, we define

$$g_{cat} = \mathbb{I}\{c(d_i) = c(d_j)\} \times sim(d_i, d_j) \quad (8)$$

2.3 Model Inference and Optimization

Let Y and X be the set of all candidate variables and attribute variables, respectively. The joint distribution encoded by the FGM is given by multiplying all potential functions.

$$P(Y, X) = \prod_i f(y_i, A) g(y_i, G(y_i)) \quad (9)$$

The desired marginal distribution can be obtained by marginalizing all other variables. Since under most circumstances, however, the factor graph is densely connected, the exact inference is intractable and approximate inference is required. After obtaining the marginal probabilities, the mixture weights λ_{d_i} in Eq. 2 are estimated by normalizing the corresponding marginal probabilities $P(y_i)$ over all candidate variables, which can be written as

$$\lambda_{d_i} = (1 - \lambda_{u_1}) \frac{P(y_i)}{\sum_{j: d_j \in \mathcal{D}_{rel}} P(y_j)} \quad (10)$$

where the constraint $\lambda_{u_1} + \sum \lambda_{d_i} = 1$ leads to a valid probability distribution for our mixture model.

A factor graph is normally optimized by gradient-based methods. Unfortunately, since the ground truth values of the mixture weights are not available, we are prohibited from using supervised approaches. Here we propose a two-step iterative procedure to optimize our model. At first, all the model parameters (i.e. α, β, λ_u) are randomly initialized. Then, we infer the marginal probabilities of candidate variables. Given these marginal probabilities, we can evaluate the perplexity of the user language model on a held-out dataset, and search for better parameters. This procedure is repeated until convergence. Also, notice that by using FGM, we reduce the number of parameters from $1 + |\mathcal{D}_{rel}|$ to $1 + |\alpha| + |\beta|$, lowering the risk of overfitting.

3 Experiments

3.1 Dataset and Experiment Setup

We perform experiments on the *Twitter* dataset collected by Galuba et al. (2010). Twitter data have been used to verify models with different purposes (Lin et al., 2011; Tan et al., 2011). To emphasize on the cold start scenario, we randomly selected 15 users with about 35 tweets and 70 friends as candidates for an authorship attribution task. Our experiment corpus consists of 4322 tweets. All words with less than 5 occurrences are removed. Stop words and URLs are also removed and all tweets are stemmed. We identify the 100 most frequent terms as categories. The size of the vocabulary set is 1377.

We randomly partitioned the tweets of each user into training, validation and testing sets. The reported result is the average of 10 random splits. In all experiments, we vary the size of training data from 1% to 15%, and hold out the same number of tweets from each user as validation and testing data. The statistics of our dataset, given 15% training data, are shown in Table 1.

Loopy belief propagation (LBP) is used to obtain the marginal probabilities of the variables (Murphy et al., 1999). Parameters are searched with the pattern search algorithm (Audet and Dennis, 2002). To not lose generality, we use the default configuration in all experiments.

# of	Max.	Min.	Avg.
Tweets	70	19	35.4
Friends	139	24	68.9
Variables	467	97	252.7
Edges	9216	231	3427.1

Table 1: Dataset statistics

3.2 Baseline Methods

We compare our framework with two baseline methods. The first (“*Cosine”*) is a straightforward implementation that sets all mixture weights λ_{d_i} to the cosine similarity between the probability mass vectors of the document and user unigram language models. The second (“*PS”*) uses the pattern search algorithm to perform constrained optimization over the mixture weights. As mentioned in section 2.3, the main difference between this method and ours (“*FGM”*) is that we reduce the search space of the parameters by FGM. Furthermore, social network information is exploited in our framework, while the PS method performs a direct search over mixture weights, discarding valuable knowledge.

Different from other smoothing methods that are usually mutually exclusive, any other smoothing methods can be easily merged into our framework. In Eq. 2, the *base language model* $P_{u_1}(w)$ can be already smoothed by any techniques before being plugged into our framework. Our framework then enriches the user language model with social network information. We select four popular smoothing methods to demonstrate such effect, namely additive smoothing, absolute smoothing (Ney et al., 1995), Jelinek-Mercer smoothing (Jelinek and Mercer, 1980) and Dirichlet smoothing (MacKay and Peto, 1994). The results of using only the base model (i.e. set $\lambda_{d_i} = 0$ in Eq. 2) are denoted as “*Base”* in the following tables.

Train %	**Additive**				**Absolute**			
	Base	Cosine	PS	FGM	Base	Cosine	PS	FGM
1%	900.4	712.6	725.5	**537.5****	895.3	703.1	722.1	**544.5****
5%	814.5	623.4	690.5	**506.8****	782.4	607.9	678.4	**510.2****
10%	757.7	566.6	684.8	**481.2****	708.4	552.7	661.0	**485.8****
15%	693.8	521.0	635.2	**474.8****	647.4	504.3	622.3	**474.1****
Train %	**Jelinek-Mercer**				**Dirichlet**			
	Base	Cosine	PS	FGM	Base	Cosine	PS	FGM
1%	637.8	571.4	643.1	**541.0****	638.5	571.3	643.1	**541.0****
5%	593.9	526.1	602.9	**505.4****	595.0	526.6	616.5	**507.2****
10%	559.2	494.1	573.8	**483.6****	560.4	494.9	579.6	**486.0****
15%	535.3	473.4	560.2	**473.0**	535.7	**473.6**	563.2	474.4

Table 2: Testing set perplexity. ****** indicates that the best score among all methods is significantly better than the next highest score, by t-test at a significance level of 0.05.

3.3 Perplexity

As an intrinsic evaluation, we first compute the perplexity of unseen sentences under each user language model. The result is shown in Table 2.

Our method significantly outperforms all of the methods in almost all settings. We observe that the *"PS"* method takes a long time to converge and is prone to overfitting, likely because it has to search about a few hundred parameters on average. As expected, the advantage of our model is more apparent when the data is sparse.

3.4 Authorship Attribution (AA)

The authorship attribution (AA) task is chosen as the extrinsic evaluation metric. Here the goal is not about comparing with the state-of-the-art approaches in AA, but showing that LM-based approaches can benefit from our framework.

To apply PLM on this task, a naïve Bayes classifier is implemented (Peng et al., 2004). The most probable author of a document d is the one whose PLM yields the highest probability, and is determined by $u^* = \mathrm{argmax}_u\{\prod_{w \in d} P_u(w)\}$.

The result is shown in Table 3. Our model improves personalization and outperforms the baselines under cold start settings. When data is sparse, the *"PS"* method tends to overfit the noise, while the *"Cosine"* method contains too few information and is severely biased. Our method strikes a balance between model complexity and the amount of information included, and hence performs better than the others.

4 Related Work

Personalization has long been studied in various textual related tasks. Personalized search is established by modeling user behavior when using search engines (Shen et al., 2005; Xue et al., 2009). Query language model could be also expanded based on personalized user modeling (Chirita et al., 2007). Personalization has also been modeled in many NLP tasks such as summarization (Yan et al., 2011) and recommendation (Yan et al., 2012). Different from our purpose, these models do not aim at exploiting social media content to enrich a language model. Wen et al. (2012) combines user-level language models from a social network, but instead of focusing on the cold start problem, they try to improve the speech recognition performance using a mass amount of texts on social network. On the other hand, our work explicitly models the more sophisticated document-level relationships using a probabilistic graphical model.

5 Conclusion

The advantage of our model is threefold. First, prior knowledge and heuristics about the social network can be adapted in a structured way through the use of FGM. Second, by exploiting a well-studied graphical model, mature inference techniques, such as LBP, can be applied in the optimization procedure, making it much more effective and efficient. Finally, different from most smoothing methods that are mutually exclusive, any other smoothing method can be incorporated into our framework to be further enhanced. Using only 1% of the training corpus, our model can improve the perplexity of base models by as much as 40% and the accuracy of authorship attribution by at most 15%.

6 Acknowledgement

This work was sponsored by AOARD grant number No. FA2386-13-1-4045 and National Science Council, National Taiwan University and Intel Corporation under Grants NSC102-2911-I-002-001 and NTU103R7501 and grant 102-2923-E-002-007-MY2, 102-2221-E-002-170, 101-2628-E-002-028-MY2.

Train %	**Additive**				**Absolute**			
	Base	Cosine	PS	FGM	Base	Cosine	PS	FGM
1%	54.67	58.27	61.07	**63.74**	49.47	57.60	58.27	**64.27****
5%	61.47	63.20	62.67	**68.40****	59.60	62.40	61.33	**66.53****
10%	61.47	65.73	66.27	**69.20****	61.47	65.20	64.67	**71.87****
15%	64.27	67.07	62.13	**70.40****	64.67	68.27	63.33	**71.60****
Train %	**Jelinek-Mercer**				**Dirichlet**			
	Base	Cosine	PS	FGM	Base	Cosine	PS	FGM
1%	54.00	60.93	62.00	**64.80****	52.80	60.40	61.87	**64.67****
5%	62.67	65.47	64.00	**68.00**	60.80	65.33	62.40	**66.93**
10%	63.87	68.00	67.87	**68.53**	62.53	67.87	66.40	**68.53**
15%	65.87	**70.40**	64.14	69.87	65.47	**70.27**	64.53	68.40

Table 3: Accuracy (%) of authorship attribution. ** indicates that the best score among all methods is significantly better than the next highest score, by t-test at a significance level of 0.05.

Reference

Charles Audet and J. E. Dennis, Jr. 2002. Analysis of generalized pattern searches. *SIAM J. on Optimization*, 13(3):889–903, August.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Paul Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. 2007. Personalized query expansion for the web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 7–14, New York, NY, USA. ACM.

Maarten Clements. 2007. Personalization of social media. In *Proceedings of the 1st BCS IRSG Conference on Future Directions in Information Access*, FDIA'07, pages 14–14, Swinton, UK, UK. British Computer Society.

Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. 2010. Outtweeting the twitterers - predicting information cascades in microblogs. In *Proceedings of the 3rd Conference on Online Social Networks*, WOSN'10, pages 3–3, Berkeley, CA, USA. USENIX Association.

Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *In Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands: North-Holland, May.

F. R. Kschischang, B. J. Frey, and H. A. Loeliger. 2006. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theor.*, 47(2):498–519, September.

Jimmy Lin, Rion Snow, and William Morgan. 2011. Smoothing techniques for adaptive online language models: Topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 422–429, New York, NY, USA. ACM.

David J.C. MacKay and Linda C. Bauman Peto. 1994. A hierarchical dirichlet language model. *Natural Language Engineering*, 1:1–19.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 467–475, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1995. On the estimation of 'small' probabilities by leaving-one-out. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(12):1202–1212, December.

Fuchun Peng, Dale Schuurmans, and Shaojun Wang. 2004. Augmenting naive bayes classifiers with statistical language models. *Inf. Retr.*, 7(3-4):317–345, September.

Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 824–831, New York, NY, USA. ACM.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1397–1405, New York, NY, USA. ACM.

Zhichun Wang, Juanzi Li, Zhigang Wang, and Jie Tang. 2012. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 459–468, New York, NY, USA. ACM.

Tsung-Hsien Wen, Hung-Yi Lee, Tai-Yuan Chen, and Lin-Shan Lee. 2012. Personalized language modeling by crowd sourcing with social network data for voice access of cloud applications. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 188–193.

Gui-Rong Xue, Jie Han, Yong Yu, and Qiang Yang. 2009. User language model for collaborative personalized search. *ACM Trans. Inf. Syst.*, 27(2):11:1–11:28, March.

Rui Yan, Jian-Yun Nie, and Xiaoming Li. 2011. Summarize what you are interested in: An optimization framework for interactive personalized summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1342–1351, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 516–525, Stroudsburg, PA, USA. Association for Computational Linguistics.

ChengXiang Zhai. 2008. *Statistical Language Models for Information Retrieval*. Now Publishers Inc., Hanover, MA, USA.

Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’01, pages 334–342, New York, NY, USA. ACM.