Learn to Weight Terms in Information Retrieval Using Category Information

Rong Jin RONGJIN@CSE.MSU.EDU
Joyce Y. Chai JCHAI@CSE.MSU.EDU
Dept. of Computer Science and Engineering, Michigan State University, MI 48824, USA
Luo Si LSI@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, MI 48824, USA

Abstract

How to assign appropriate weights to terms is one of the critical issues in information retrieval. Many term weighting schemes are unsupervised. They are either based on the empirical observation in information retrieval, or based on generative approaches for language modeling. As a result, the existing term weighting schemes are usually insufficient in distinguishing informative words from the uninformative ones, which is crucial to the performance of information retrieval. In this paper, we present supervised term weighting schemes that automatically learn term weights based on the correlation between word frequency and category information of documents. Empirical studies with the ImageCLEF dataset have indicated that the proposed methods perform substantially better than the state-of-the-art approaches for term weighting and other alternatives that exploit category information for information retrieval.

1. Introduction

Previous studies of information retrieval (Croft and Harper, 1979; Robertson et al., 1981; Salton and Buckley, 1988; Fuhr, 1992; Robertson et al., 1996; Singhal et al., 1996; Ponte, 1998; Miller et al., 1999; Robertson et al., 2000; Zhai and Lafferty, 2001) have shown that appropriate term weights are crucial to the performance of information retrieval systems. Sophisticated term weighting schemas, such as the Okapi formula, usually result in substantially better performance than the simple method that treats every term occurrence equally. In general, the current state-of-art term weighting methods can be divided into two categories:

1) The TF.IDF based term weighting methods (Robertson et al., 1981; Salton and Buckley, 1988; Robertson et al., 1996; Singhal et al., 1996). Most methods under this family are based on intuition and empirical observation. For example, the inverse document frequency (i.e., idf) weight is based on the intuition that words appearing infrequently in a collection tend to be more informative than the words that appear frequently across many documents. However, very often, this intuition is violated. For instance, typos tend to appear rarely across a collection but they are uninformative to the content of documents.
2) The language modeling approaches (Ponte, 1998; Ponte and Croft, 1998; Miller et al., 1999; Lafferty and Zhai, 2001; Zhai and Lafferty, 2001; Jin et al., 2002; Zhai and Lafferty, 2002). These approaches assume that documents are generated by simple statistical language models, which are estimated by the maximum likelihood estimation (MLE) combined with smoothing techniques. However, due to the generative nature of language modeling approaches, they usually lack of discriminative power in distinguishing informative words from uninformative ones. In fact, their limited discriminative power comes from the smoothing technique, not from the generative model itself.

The essential difficulty with determining term weights is the lack of supervision. Usually, term weights are only determined by term frequency across documents, which may not reflect the informativeness of terms. Some additional information (e.g., category information) associated with documents, on the other hand, can serve as "guidance" in determining which terms are more informative than the others. In fact, we can find such meta information along with documents in many collections (e.g., library collections). How to effectively utilize such information to improve term weighting becomes an important question. In this paper, we present two novel approaches that automatically learn term weights by incorporating the category information of documents. We

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

assume that the semantic meaning of a document can be represented, at least partially, by the set of categories that it belongs to. Thus, by measuring the similarity in category labels assigned to two documents, we will be able to tell content wise how similar they are. To distinguish important terms from common terms, we will search for term weights so that the term-based document similarities are consistent with the similarities that are computed from the category information of documents. Unlike inverse document frequency and language modeling approaches, where term weights are determined in an unsupervised manner, our approaches learn term weights under the guidance of category information. Notice that our work is different from text categorization problems, in which term weights are learned to determine if a document belongs to a specific category. In particular, term weights learned in text categorization are category dependent. A different category can result in a completely set of term weights. In contrast, term weights in information retrieval is independent from target categories and should reflect the overall informativeness of words to the content of documents.

The rest of this paper is organized as follows: Section 2 discusses the related work; Section 3 describes the proposed algorithm; Empirical studies are presented in Section 4; Section 5 concludes this paper with the future work.

2. Related Work

In the following subsections, we briefly review the two types of approaches for term weighting, namely the tf.idf based approaches and the language model based approaches, separately.

2.1 TF.IDF Based Term Weighting Methods

Most term weighting schemes within this family contain three factors: the term frequency factor (i.e., tf), the inverse document frequency factor (i.e., idf), and the term normalization factor (i.e., norm). The final term weight is the product of these three factors. Numerous term weighting schemas have been developed within the TF.IDF family (Robertson et al., 1981; Salton and Buckley, 1988; Robertson et al., 1996; Singhal et al., 1996). Among them, the Okapi formula (Robertson et al., 1996) is one of the most popular term weighting methods. It defines the similarity between a document *d* and a given query *q* as follows:

$$sim(d,q) = \sum_{t \in q} f(t,q) \frac{kf(t,d)}{f(t,d) + k\left(1 - b + b\frac{doclen(d)}{avg\_doclen}\right)} \log\left(\frac{N + 0.5}{N(t)}\right)$$

where $f(t,q)$ and $f(t,d)$ are the term frequency for word '*t*' in query *q* and document *d*. *N* is the total number of documents in the collection, $N(t)$ is the number of documents in the collection that have word '*t*'. Both *k* and *b* are parameters.

The most noticeable feature of the Okapi term weighting schema is $kf(t,d) \Big/ \left\{ f(t,d) + k\left(1 - b + b\frac{doclen(d)}{avg\_doclen}\right)\right\}$. It is based on the intuition that the first occurrence of a query word in a document is usually more important than other occurrences in determining the relevance of a document to a given query. Furthermore, the Okapi formula employs the pivoted normalization factor $1 - b + b\frac{doclen(d)}{avg\_doclen}$, which has been shown effective for information retrieval (Robertson et al., 1996; Singhal et al., 1996). The two parameters (k, d) in the Okapi formula are determined empirically. In our experiment, we set them to be 2 and 0.75, which are the typical values used in previous studies (Robertson et al., 1996).

2.2 Language Model based Term Weighting Methods

Recently, language modeling approaches have shown promising performance in information retrieval (Ponte, 1998; Ponte and Croft, 1998; Miller et al., 1999; Lafferty and Zhai, 2001; Zhai and Lafferty, 2001; Jin et al., 2002; Zhai and Lafferty, 2002). To determine the relevance of a document *d* to a given query *q*, the language modeling approaches estimate the conditional probability $p(q|d)$, which is the likelihood of generating query *q* given the content of document *d*. By assuming that each query word is generated independently, this likelihood is simply computed as: $p(q|d) \propto \prod_{w \in q} p(w|d)$ where $p(w|d)$ is the unigram probability for document *d*.

The key to language modeling approaches is how to estimate the unigram probabilities $\{p(w|d)\}$ for a document *d*. A simple approach that estimates unigram probabilities based on word occurrence will suffer severely from the sparse data problem. In particular, it assigns zero probability to any words that do not appear in a document. Thus, a smoothing technique is usually used in estimating language models. Smoothing techniques are critical to information retrieval in that they equip language modeling approaches with the discriminative power in distinguishing informative words from uninformative words. Take the Jelinek-Mercer (JM) smoothing as an example. Let $\{p(w|c)\}$ be the unigram probabilities estimated from a collection of documents. Using the JM smoothing, $p(w|d)$ is estimated as:

$$p(w|d) = p(w|c)\left(1 - \alpha + \alpha\frac{f(w,d)}{|d|\,p(w|c)}\right)$$

Since the ratio $\frac{f(w,d)}{|d|\,p(w|c)}$ for a common word is significantly smaller than an uncommon word, the JM smoothing is able to distinguish common words from uncommon words. Note that if without using the smoothing technique, which corresponds to $\alpha = 1$, a simple language modeling approach will assign large probabilities to common words and therefore it is unable to differentiate important words from common words.

3. Discriminative Approaches for Automatic Term Weighting

In this section, we will discuss how to automatically learn appropriate term weights using the category labels of documents through a discriminative analysis. The basic idea is to find term weights such that the term-based document similarity is consistent with the similarity computed based on category labels of documents.

Let $T = \{d_i\}_{i=1}^{N}$ be the collection of documents where *N* is the total number of documents. Each document d_i is represented by both a term vector $\vec{w}_i = \{w_{i,1}, w_{i,2}, ..., w_{i,n}\}$, and a category vector $\vec{c}_i = \{c_{i,1}, c_{i,2}, ..., c_{i,m}\}$. $w_{i,j}$ is the term frequency for the j-th word in the i-th document. $c_{i,j}$ is a binary variable, which is 1 when the i-th document belongs to the j-th category and zero otherwise. *n* is the size of vocabulary, and *m* is the number of different categories. Let $\vec{\mu} = \{\mu_i\}_{i=1}^{n}$ be the term weights, and $\vec{\eta} = \{\eta_i\}_{i=1}^{m}$ be the weights for categories. Let $S_c(d, d'; \vec{\eta})$ and $S_w(d, d'; \vec{\mu})$ be the category-based and term-based similarities between documents *d* and *d'*, respectively. They are defined as weighted dot products:

$$S_c(d_i, d_j; \vec{\eta}) = \sum_{k=1}^{m} \eta_k c_{i,k} c_{j,k}$$

$$S_w(d_i, d_j; \vec{\mu}) = \sum_{k=1}^{n} \mu_k w_{i,k} w_{j,k}$$

Since our goal is to find appropriate weights $\vec{\eta}$ and $\vec{\mu}$ such that two similarity measurements are consistent, it can be formulated as the following optimization problem:

$$\{\vec{\eta}^*, \vec{\mu}^*\} = \arg\min_{\vec{\eta},\vec{\mu}} \sum_{d \neq d' \in T} l\left(S_c(d, d'; \vec{\eta}), S_w(d, d'; \vec{\mu})\right) \quad (1)$$

where $l(\cdot,\cdot)$ measures the difference between the two similarity measurements. Note that, in the above discussion, we introduce weights for both words and categories. This is important because some categories are more general than the others. Two documents are more likely to have similar content when they match in rare categories than in common categories.

3.1 A Regression Approach

One choice for loss function $l(S_c, S_w)$ is the Euclidean distance, i.e., $l(S_c, S_w) = (S_c - S_w)^2$. Thus, the objective function in (1) is expressed as:

$$\begin{aligned} F_{reg} &= \sum_{i,j=1}^{N} \left(\sum_{k=1}^{m} \eta_k c_{i,k} c_{j,k} - \sum_{k=1}^{n} \mu_k w_{i,k} w_{j,k} \right)^2 \\ &= \left\| \mathbf{C}\boldsymbol{\eta}\mathbf{C}^T - \mathbf{W}\boldsymbol{\mu}\mathbf{W}^T \right\|_F \end{aligned} \quad (2)$$

Where $\mathbf{C} = (\vec{c}_1, \vec{c}_2, ..., \vec{c}_N)^T$, $\mathbf{W} = (\vec{w}_1, \vec{w}_2, ..., \vec{w}_N)^T$, $\boldsymbol{\eta} = diag(\vec{\eta})$ and $\boldsymbol{\mu} = diag(\vec{\mu})$. Eq. (2) can be further simplified into the following quadratic form:

$$F_{reg} = \begin{pmatrix} \vec{\eta}^T & \vec{\mu}^T \end{pmatrix} \begin{pmatrix} \mathbf{Q}_c & -\mathbf{P}^T \\ -\mathbf{P} & \mathbf{Q}_w \end{pmatrix} \begin{pmatrix} \vec{\eta} \\ \vec{\mu} \end{pmatrix} \quad (3)$$

where $[\mathbf{Q}_w]_{i,j} = \left(\vec{\xi}_i^T \vec{\xi}_j\right)^2$, $[\mathbf{Q}_c]_{i,j} = \left(\vec{\zeta}_i^T \vec{\zeta}_j\right)^2$, and $[\mathbf{P}]_{i,j} = \left(\vec{\xi}_i^T \vec{\zeta}_j\right)^2$. Here, $\vec{\xi}_j$ is the j-th column in matrix **W**, and $\vec{\zeta}_j$ is the j-th column in matrix **C**. Apparently, the trivial solution to (3) is $\vec{\eta} = \vec{\mu} = 0$. To remove the trivial solution, we consider a constraint on the L2 norm of weights, i.e.,

$$\|\vec{\eta}\|_2^2 + \|\vec{\mu}\|_2^2 = \sum_{i=1}^{m} \eta_i^2 + \sum_{i=1}^{n} \mu_i^2 = 1 \quad (4)$$

Then, the optimal solution to (4) is the minimum eigenvector for matrix $\begin{pmatrix} \mathbf{Q}_c & -\mathbf{P}^T \\ -\mathbf{P} & \mathbf{Q}_w \end{pmatrix}$. Given that this solution contains both negative and positive weights, it could be undesirable since a negative weight indicates that the corresponding word or category serves as negative evidence in determining document similarity, which contradicts our intuition. Thus, we further introduce positive constraints on weights, i.e.,

$$\eta_i \geq 0, \forall\ i \in [1...m],\ \mu_i \geq 0, \forall\ j \in [1...n] \quad (5a)$$

To further simplify computation, we replace L2 norm in (4) with L1 norm, i.e.,

$$\sum_{i=1}^{m} \eta_i + \sum_{i=1}^{m} \mu_i \geq 1 \quad (5b)$$

Now, given the quadratic objective in (3) and the linear constraints in (5a) and (5b), a standard quadratic programming technique (Gill et al., 1981) can be applied to acquire the solution.

Implementation Issues. One computational problem with the regression approach is that it requires computing the full matrix $\mathbf{Q}_w$, whose size is $n \times n$. When the number of words (i.e., *n*) is large (e.g., 20,000 in our case), computing the full matrix $\mathbf{Q}_w$ will be infeasible. In our experiment, we realize the regression approach by

iteratively sampling the word space. In particular, for each iteration, a small number of words are sampled out (e.g.., 2000 in our experiment) and only the weights for the selected words are updated during the iteration. Let $\vec{\mu}_s$ and $\vec{\mu}_u$ be the weights for selected words and unselected words, respectively. Rewrite matrix **P** and $\mathbf{Q}_w$ into the form:

$$\mathbf{P} = [\mathbf{P}_u, \mathbf{P}_s], \mathbf{Q}_w = \begin{bmatrix} \mathbf{Q}_w^s & \mathbf{Q}_w^{us} \\ \mathbf{Q}_w^{su} & \mathbf{Q}_w^u \end{bmatrix}$$

where index 'u' and 's' represent parts of matrices related to selected words and unselected words, respectively. Using the above notations, F_{reg} is rewritten as:

$$\begin{aligned} F_{reg} &= \begin{pmatrix} \vec{\eta}^T & \vec{\mu}_s^T \end{pmatrix} \begin{pmatrix} \mathbf{Q}_c & -\mathbf{P}_s^T \\ -\mathbf{P}_s & \mathbf{Q}_w^s \end{pmatrix} \begin{pmatrix} \vec{\eta} \\ \vec{\mu}_s \end{pmatrix} \\ &-2\left(\vec{\mu}_u^T \mathbf{P}_u \vec{\eta} + \vec{\mu}_u^T \mathbf{Q}_w^{us} \vec{\mu}_s\right) + \vec{\mu}_u^T \mathbf{Q}_w^u \vec{\mu}_u \end{aligned} \quad (6)$$

Since the last part of F_{reg} is irrelevant to $\vec{\mu}_s$ and $\vec{\eta}$, it is ignored during the computation. Given that $\mathbf{Q}_w^s$ and $\mathbf{Q}_w^{us}$ are of small size, quadratic programming techniques can be efficiently applied to (6).

3.2 A Probabilistic Approach

Another choice for $l(S_c, S_w)$ is to first convert the similarities based on terms and categories into probabilities and then compare the two probabilities. A logit function is used to convert a similarity score into a probability. We define $p_{i,j}^c$ and $p_{i,j}^w$, i.e., the probabilities for two documents to be similar based on their category and term information, as

$$\begin{aligned} p_{i,j}^c(\vec{\eta}) &= \frac{1}{1+\exp\left(-S_c(d_i, d_j; \vec{\eta}) + \eta_0\right)} \\ p_{i,j}^w(\vec{\mu}) &= \frac{1}{1+\exp\left(-S_w(d_i, d_j; \vec{\mu}) + \mu_0\right)} \end{aligned} \quad (7)$$

where η_0 and μ_0 are bias terms. Then, we define $l(S_c, S_w)$ as the cross entropy between $p_{i,j}^c$ and $p_{i,j}^w$:

$$\begin{aligned} &l\left(S_c(d_i, d_j; \vec{\eta}), S_w(d_i, d_j; \vec{\mu})\right) \\ &= -p_{i,j}^c(\vec{\eta}) \log p_{i,j}^w(\vec{\mu}) - (1 - p_{i,j}^c(\vec{\eta})) \log(1 - p_{i,j}^w(\vec{\mu})) \end{aligned} \quad (8)$$

The above expression is minimized when the two sets of probabilities $p_{i,j}^c$ and $p_{i,j}^w$ are of similar values and they are close to either 1 or 0. In other words, $l(S_c, S_w)$ is minimized when the two similarity measurements are consistent and confident in predicting if two documents are similar. Note that we did not use the KL divergence for $l(S_c, S_w)$. This is because a KL divergence based objective function can result in a trivial solution, in which case all weights and bias are assigned to be zeros. In contrast, the trivial solution will not be optimal solution to (8).

Using the definition in (7), now the objective function is written as:

$$\begin{aligned} F_{prob} &= \sum_{i \neq j}^{N} \left(p_{i,j}^c(\vec{\eta}) \log p_{i,j}^w(\vec{\mu}) + (1 - p_{i,j}^c(\vec{\eta})) \log(1 - p_{i,j}^w(\vec{\mu})) \right) \\ &= \sum_{i \neq j}^{N} \left\{ \begin{aligned} &\frac{1}{1+\exp\left(-\sum_k \eta_k c_{i,k} c_{j,k} + \eta_0\right)} \log \frac{1}{1+\exp\left(-\sum_k \mu_k w_{i,k} w_{j,k} + \mu_0\right)} \\ &+ \frac{1}{1+\exp\left(\sum_k \eta_k c_{i,k} c_{j,k} - \eta_0\right)} \log \frac{1}{1+\exp\left(\sum_k \mu_k w_{i,k} w_{j,k} - \mu_0\right)} \end{aligned} \right\} \end{aligned}$$

Similar to the consideration in the regression approach, we constrain all the weights to be non-negative, i.e., $\eta_i, \mu_j \geq 0$. Furthermore, similar to the logistic regression model used for text categorization (Zhang and Oles, 2001), a Laplacian prior is introduced into the objective function to prevent weights from being too large. Thus, the final optimization problem for the probabilistic approach becomes:

$$\left\{\vec{\eta}^*, \vec{\mu}^*, \eta_0^*, \mu_0^*\right\} = \arg\max_{\vec{\eta}, \vec{\mu}, \eta_0, \mu_0} F_{prob} - \alpha_w |\vec{\mu}|_1 - \alpha_c |\vec{\eta}|_1 \quad (9)$$

subject to $\vec{\eta}, \vec{\mu} \geq 0$ and $\eta_0, \mu_0 \geq 0$

Finding the optimal solution to the above expression is not easy. One essential difficulty is in evaluating the objective function in (9), which requires computing $p_{i,j}^c$ and $p_{i,j}^w$ for any two documents. Given a large number of documents (e.g., 40,000 in our experiment), the pair wise computation can be too expensive. Similar to the regression approach, we will apply an iterative procedure to the optimization. In each iteration, only a small number of terms are selected and only the weights associated with the selected terms are updated during the iteration. However, in the regression approach, the objective function is quadratic, and therefore weights for selected terms and for unselected terms can be easily separated in the objective function. In contrast, the objective function in (9) is rather complicated compared to the one in (3) and it is not so easy to separate weights for selected terms from weights for unselected ones. To this end, we use the bound optimization algorithm (Salakhutdinov and Roweis, 2003). The main idea is that, although it is difficult to separate weights in the original function, it may be easy to do so for a simple function that lower bounds the original function. Furthermore, since we need to find weights for both categories and terms, the bound algorithm will be applied twice in each iteration: one for finding the optimal term weights given fixed category

weights, and the other one for finding the optimal weights for category given fixed term weights. In the following, we will discuss the steps for optimizing term weights and category weights, separately.

Learn category weights $\vec{\eta}$ with fixed term weights $\vec{\mu}$. Let $\{\vec{\eta},\vec{\mu}\}$ be the weights of the previous iteration, and $\vec{\eta}' = \vec{\eta} + \vec{\delta}$ be the new category weights for the current iteration. Probabilities $p^c_{u,v}(\vec{\eta}')$ and $\overline{p}^c_{i,j}(\vec{\eta}) = 1 - p^c_{i,j}(\vec{\eta})$ can be upper bounded by the following expression:

$$p^c_{i,j}(\vec{\eta}') \le p^c_{i,j}(\vec{\eta}) + p^c_{i,j}(\vec{\eta})\overline{p}^c_{i,j}(\vec{\eta})\left\{\exp\left(\sum_k c_{i,}c_{j,k}\delta_k - \delta_0\right) - 1\right\}$$

$$\overline{p}^c_{i,j}(\vec{\eta}') \le \overline{p}^c_{i,j}(\vec{\eta}) + p^c_{i,j}(\vec{\eta})\overline{p}^c_{i,j}(\vec{\eta})\left\{\exp\left(-\sum_k c_{i,}c_{j,k}\delta_k + \delta_0\right) - 1\right\}$$

Now, the objective function in (9) can be lower bounded as:

$$\begin{aligned}
&F_{prob}(\vec{\eta}',\vec{\mu}) - \alpha_w|\vec{\mu}|_1 - \alpha_c|\vec{\eta}'|_1 \ge \\
&F_{prob}(\vec{\eta},\vec{\mu}) - \alpha_w|\vec{\mu}|_1 - \alpha_c|\vec{\eta}|_1 \\
&+\sum_{i\ne j} p^c_{i,j}(\vec{\eta})\overline{p}^c_{i,j}(\vec{\eta})\log p^w_{i,j}(\vec{\mu})\left\{\exp\left(\sum_k c_{i,k}c_{j,k}\delta_k - \delta_0\right) - 1\right\} \\
&+\sum_{i\ne j} p^c_{i,j}(\vec{\eta})\overline{p}^c_{i,j}(\vec{\eta})\log \overline{p}^w_{i,j}(\vec{\mu})\left\{\exp\left(-\sum_k c_{i,k}c_{j,k}\delta_k + \delta_0\right) - 1\right\}
\end{aligned} \tag{10}$$

Using the Jensen inequality, the exponential term in the above expression can be upper bounded as:

$$\begin{aligned}
&\exp\left(\sum_k c_{i,k}c_{j,k}\delta_k - \delta_0\right) - 1 \\
&\le \frac{\sum_k \mathrm{e}^{\delta_j L_c c_{i,k}c_{i,k}}}{L_c} + \frac{e^{L_c\delta_0}}{L_c} - \frac{1+\sum_k I\left(c_{i,k}c_{i,k}\right)}{L_c}
\end{aligned}$$

where $L_c = 2\max_i |\vec{c}_i|_0 + 1$, and $I(\cdot)$ is an indicator that outputs one for positive value and zero otherwise. Then, the upper bound in (10) can be further relaxed as:

$$\begin{aligned}
&F_{dis}(\vec{\eta}',\vec{\mu}) - \alpha_w - \alpha_w|\vec{\mu}|_1 - \alpha_c|\vec{\eta}'|_1 \ge \\
&\left\{F_{dis}(\vec{\eta},\vec{\mu}) - \alpha_w - \alpha_w|\vec{\mu}|_1 - \alpha_c|\vec{\eta}|_1\right\} \\
&-\sum_{i\ne j} p^c_{i,j}(\vec{\eta})\overline{p}^c_{i,j}(\vec{\eta})\frac{1+\sum_k I\left(c_{i,k}c_{j,k}\right)}{L_c}\left(\log p^w_{i,j}(\vec{\mu}) + \log \overline{p}^w_{i,j}(\vec{\mu})\right) \\
&+\sum_k \frac{\sum_{i\ne j} p^c_{i,j}(\vec{\eta})\overline{p}^c_{i,j}(\vec{\eta})\begin{pmatrix}\log p^w_{i,j}(\vec{\mu})\,\mathrm{e}^{\delta_k L_c c_{i,k}c_{j,k}} \\ +\log \overline{p}^w_{i,j}(\vec{\mu})\,\mathrm{e}^{-\delta_k L_c c_{i,k}c_{j,k}}\end{pmatrix}}{L_c+1} - \alpha_c\delta_k \\
&+\sum_{i\ne j} \frac{p^c_{i,j}(\vec{\eta})\overline{p}^c_{i,j}(\vec{\eta})\begin{pmatrix}e^{-L_c\delta_0}\log p^w_{i,j}(\vec{\mu}) \\ +e^{L_c\delta_0}\log \overline{p}^w_{i,j}(\vec{\mu})\end{pmatrix}}{L_c} - \alpha\delta_0
\end{aligned}$$

Notice that in the above expression, the interaction between weights for different categories is removed. Thus, each category weight can be updated independently. In fact, there is an analytical solution to the optimal weight change $\vec{\delta}$ that maximizes the above lower bounds. To avoid computing $p^c_{i,j}$ for all document pairs, in each iteration, only a subset of categories are selected in each iteration and their weights are updated. As a result, only the probabilities $p^c_{i,j}$ for the documents that belong to the selected categories are needed to be computed.

Learn term weights $\vec{\mu}$ with fixed category weights $\vec{\eta}$. Let $\vec{\mu}' = \vec{\mu} + \vec{\delta}$ be the new category weights for the current iteration. First, $\log \overline{p}^w_{i,j}(\vec{\mu}')$ can be lower bounded using the Jensen inequality:

$$\log p^w_{i,j}(\vec{\mu}') \ge \log p^w_{i,j}(\vec{\mu}) - \overline{p}^w_{i,j}(\vec{\mu})\left[\begin{array}{l}\dfrac{\sum_k \mathrm{e}^{-\delta_k L_w w_{i,k}w_{j,k}}}{L_w} + \dfrac{e^{L_w\delta_0}}{L_w} \\ -\dfrac{1+\sum_k I\left(w_{i,k}w_{j,k}\right)}{L_w}\end{array}\right]$$

where $L_c = 2\max_i |\vec{w}_i|_0 + 1$. By substituting the above expression for $\log \overline{p}^w_{u,v}(\vec{\mu}')$, we have a lower bound for the objective function in (9):

$$\begin{aligned}
&F_{prob}(\vec{\eta},\vec{\mu}') - \alpha_w - \alpha_w|\vec{\mu}'|_1 - \alpha_c|\vec{\eta}|_1 \\
&\ge F_{prob}(\vec{\eta},\vec{\mu}) - \alpha_w - \alpha_w|\vec{\mu}|_1 - \alpha_c|\vec{\eta}|_1 \\
&+\sum_{i\ne j}\frac{1+\sum_k I\left(w_{i,k}w_{j,k}\right)}{L_w}\left(p^c_{i,j}(\vec{\eta})\overline{p}^w_{i,j}(\vec{\mu}) + \overline{p}^c_{i,j}(\vec{\eta})p^w_{i,j}(\vec{\mu})\right) \\
&-\sum_k\left\{\begin{array}{l}\beta\delta_k + \sum_{i\ne j} p^c_{i,j}(\vec{\eta})\overline{p}^w_{i,j}(\vec{\mu})\dfrac{\mathrm{e}^{-\delta_k L_w w_{i,k}w_{j,k}}}{L_w} \\ +\sum_{i\ne j}\overline{p}^c_{i,j}(\vec{\eta})p^w_{i,j}(\vec{\mu})\dfrac{\mathrm{e}^{\delta_k L_w w_{i,k}w_{j,k}}}{L_w}\end{array}\right\} \\
&-\beta\delta_0 - \sum_{i\ne j}\left\{\begin{array}{l}p^c_{i,j}(\vec{\eta})\overline{p}^w_{i,j}(\vec{\mu})\dfrac{e^{L_w\delta_0}}{L_w} \\ +\overline{p}^c_{i,j}(\vec{\eta})p^w_{i,j}(\vec{\mu})\dfrac{e^{-L_w\delta_0}}{L_w}\end{array}\right\}
\end{aligned}$$

Again, in the above expression for the low bound, the correlation among weights for different term is removed, and thus each term weight can be updated independently. To avoid estimating $p^w_{i,j}$ for all document pairs, in each iteration, we can select a subset of term weights for updating.

4. Experiment

In this experiment, we examine the effectiveness of the proposed approaches that automatically learn term

	Using Category Information				No Category Information	
	Regression	**Probabilistic**	**ICF**	**CQE**	**Okapi**	**LM**
Avg. Prec.	0.45	0.48	0.38	0.42	0.41	0.41
Prec@						
5 doc	0.55	0.56	0.40	0.50	0.47	0.50
10 doc	0.48	0.51	0.40	0.48	0.45	0.48
20 doc	0.46	0.46	0.39	0.42	0.39	0.38
100 doc	0.21	0.21	0.19	0.19	0.20	0.20

Table 1: Retrieval results for a regression approach (Sec. 3.1), a probabilistic approach (Sec. 3.2), a inverse category frequency (ICF), a category-based query expansion (CQE), the Okapi method, and a language model (LM).

weighting by incorporating category information of documents. In particular, we will address the following research questions:

1) *Will category information be effective in improving the retrieval performance over the existing term weighting approaches*? We compare the proposed algorithms to the two state-of-the-art methods that do not use category information.

2) *How effective are our approaches in utilizing category information for information retrieval compared to other approaches*? We compare the proposed algorithms to other information retrieval methods that also exploit category information.

4.1 Experiment Design

We use the document collection from ImageCLEF (http://ir.shef.ac.uk/imageclef/). It consists of 28,133 documents and each document provides content description for a historical picture. All the documents in the collection are categorized into totally 933 different categories. Each document can be assigned to multiple categories and the average number of categories assigned to a document is about 4.5. In order to tune the parameters in the proposed method, five queries from ImageCLEF 2003 together with their relevance judgments are used as training data. 25 queries from ImageCLEF 2004 are used in our evaluation. Typical information retrieval metrics are employed in this empirical study, including average precision across 11 recall points and precision for top retrieved documents.

In order to see the effectiveness of the proposed algorithms in utilizing the category information, we compare it to two baseline models for comparison:

1) *Inverse category frequency for term importance.* For each word '*w*', we compute its inverse category frequency (i.e., icf) value, i.e., $icf(w) = \log m - \log m_c(w)$, where $m_c(w)$ is the number of categories that contain word '*w*'. This factor is then used to replace the idf factor in the Okapi formula. This approach is based on the assumption that a word tends to be less informative when it appears across a large number of categories. We hypothesize that the category frequency could be a better indicator for term importance than the document frequency. Consider an extreme case when a document is repeated thousands of times in a collection. A document frequency based approach would assign low weights to any words in the document while a category frequency based approach will not.

2) *Category-based query expansion.* A typical query expansion approach expands original queries with the terms that frequently appear in the top retrieved documents. With the category information of documents, we can expand original queries with the common categories for the top retrieved documents. With the expanded queries, documents are required to match not only the query words but also the expanded categories. Let the expanded query denoted by $q = \{f(w_1, q), ..., f(w_n, q); f(c_1, q), ..., f(c_m, q)\}$ where $f(w_i \mid q)$ and $f(c_i \mid q)$ is term frequency and category frequency for the expanded query *q*. Then, the likelihood $p(q \mid d)$ is computed as:

$$\log p(q \mid d) = \frac{\beta \sum_{i=1}^{n} f(w_i, q) \log p(w_i \mid d)}{\sum_{i=1}^{n} f(w_i, q)} + \frac{(1-\beta) \sum_{i=1}^{m} f(c_i, q) \log p(c_i \mid d)}{\sum_{i=1}^{m} f(c_i, q)}$$

where β is the smoothing parameter. $p(c_i \mid d)$ is $\gamma \frac{f(c_i, d)}{\sum_{i=1}^{m} f(c_i, d)} + (1-\gamma) \frac{\sum_{k=1}^{N} f(c_i, d_k)}{\sum_{k=1}^{N} \sum_{i=1}^{n} f(c_i, d_k)}$, where $f(c_i, d)$ is one when document *d* belongs to the category c_i and zero otherwise. γ is another smoothing parameter. Both parameters are tuned using the training queries.

4.2 Experiment (I): The Effectiveness of Category Information on Term Weighting

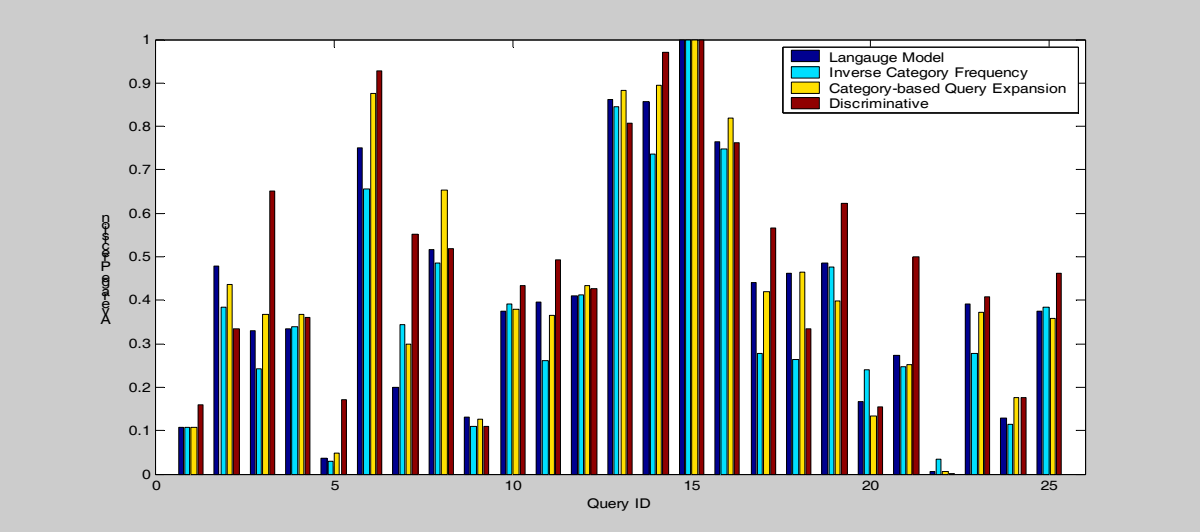

Figure 2: Average precision of individual queries for a language model, the inverse category frequency, and category-based query expansion, and the proposed probabilistic approach (titled as 'discriminative').

The retrieval performance of both the regression approach and the probabilistic approach is summarized in Table 1. We also included the performance of both the Okapi method and the language modeling approach in Table1. The precision-recall curves for these four methods are plotted in Figure 3.

First, both the regression approach and the probabilistic approach, which utilizes category information, outperform the Okapi method and the language modeling approach, which do not utilize the category information. This is further confirmed by the precision-recall curves in Figure 3. Furthermore, the precision result shown in Table 1 indicates that the largest gain of the proposed algorithms is achieved for the top ranked documents. When only five documents are retrieved for each query, the two proposed algorithms are able to achieve precision around 0.56, while the precision for the Okapi method and the language modeling approach is no more than 50%. The advantage disappears when 100 documents are retrieved for each query. Finally, comparing the two proposed algorithms, we see that the probabilistic approach performs slightly better than the regression approach in all evaluation metrics.

Third, to further examine the difference between the proposed probabilistic approach and the state-of-the-art term weighting methods, we computed average precision of individual queries for the probabilistic approach and the language modeling approach. The results are shown in Figure 2. We see that the probabilistic approach outperforms the language modeling approach substantially over 16 out of 25 queries. Only for 5 queries, the language modeling approach achieves better accuracy than the probabilistic approach.

Based on the above observation, we conclude that category information is effective for determining term weights for information retrieval.

4.3 Experiment (II): Effectiveness in Exploiting Category Information for IR

In order to see how effective the proposed algorithms are in utilizing category information, we compare the proposed algorithms to two baseline models that also exploit category information for information retrieval. The two baselines are: the inverse category frequency approach (ICF) and the category-based query expansion (CQE). The retrieval results of these two baseline models are included in Table 1. The precision-recall curves of the two baseline models are shown in Figure 3 and their average precision for individual queries are shown in Figure 4.

First, according to Table 1, the inverse category frequency approach performs slightly worse than the language modeling approach. A detailed examination of precision recall curves indicate that, compared to the language modeling approach, the disadvantage of the inverse category frequency approach is mainly at the low recall points. This is further confirmed by the results of precision for top ranked documents in Table 1. The poor performance of ICF can be attributed to the fact that the ICF method treats each category equally when it computes the 'importance' of terms. Second, the category-based query expansion performs slightly better than the language modeling approach. However, the overall improvement is rather marginal. Furthermore, the CQE method does not acquire any noticeable improvement for the top ranked documents, which is usually more important than the document ranked at

bottom. Based on the above observation, we conclude that although category information can be useful for information retrieval, how to incorporate the category information into the framework of information retrieval is also critical.

5. Conclusion and Future Work

In this paper, we present novel algorithms for automatic term weighting that exploit document category information. These algorithms learn term weights based on the correlation between term frequency and category information of documents. Two strategies have been examined: a regression approach and a probabilistic approach. Another important contribution of this paper is that we propose a bounding algorithm for efficiently learning both category weights and term weights. As a future work, we plan to introduce nonlinearity into the current work by replacing the dot product in the similarity function with a predefined kernel function. We also plan to apply the current framework to learn weights for image features through a collection of annotated images.

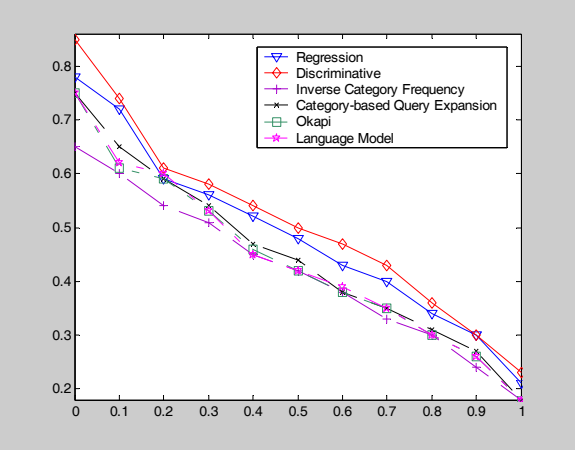

Figure 3: Precisions for 11 recall points for different term weightings.

References

Croft, W. B. and D. J. Harper (1979). "Using probabilistic models of document retrieval without relevance information." Journal of Documentation **35**: 285-295.

Fuhr, N. (1992). "Probabilistic models in information retrieval." The Computer Journal **35**(3): 243-255.

Gill, P. E., W. Murray and M. H. Wright (1981). Practical Optimization. London, Academic Press.

Jin, R., C. X. Zhai and A. G. Hauptmann (2002). Title Language Model for Information Retrieval. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002).

Lafferty, J. and C. X. Zhai (2001). Document language models, query models, and risk minimization for information retrieval. The 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Miller, D. H., T. Leek and R. Schwartz (1999). A hidden markov model information retrieval system. Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval.

Miller, D. R., T. Leek and R. M. Schwartz (1999). A hidden Markov model information retrieval system. Proceedings of 22nd ACM International Conference on Research and Development in Information Retrieva.

Ponte, J. (1998). A Language Modeling Approach to Information Retrieval. Department of Computer Science, Univ. of Massachusetts at Amherst

Ponte, J. M. and W. B. Croft (1998). A Language Modeling Approach to Information Retrieval. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.

Robertson, S. E., C. J. van Rijsbergen and M. F. Porter (1981). Probabilistic models of indexing and searching. Information Retrieval Research. P. W. Williams. London, Butterworth.

Robertson, S. E., S. Walker and M. Beaulieu (2000). "Experimentation as a way of life: Okapi at TREC." Information Processing and Management **36**: 95-108.

Robertson, S. E., S. Walker, M. M. HancockBeaulieu, M. Gatford and A. Payne (1996). Okapi at TREC-4. The Fourth Text REtrieval Conference (TREC-4).

Salakhutdinov, R. and S. T. Roweis (2003). Adaptive Overrelaxed Bound Optimization Methods. Proceedings of the Twentieth International Conference (ICML 2003).

Salton, G. and C. Buckley (1988). "Term-weighting approaches in automatic text retrieval." Information Processing and Management: an International Journal **24**(5): 513-523.

Singhal, A., C. Buckley and M. Mitra (1996). Pivoted Document Length Normalization. Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval.

Zhai, C. and J. Lafferty (2002). Two-stage language models for information retrieval. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.

Zhai, C. X. and J. Lafferty (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. Proceedings of The 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Zhang, T. and F. J. Oles (2001). "Text Categorization Based on Regularized Linear Classification Methods." Information Retrieval **4**(1): 5-31.