A Learning-based Term-Weighting Approach for Information Retrieval

GuangCan Liu Yong Yu Xing Zhu
Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China, 200030
{roth,yyu,redstar}@apex.sjtu.edu.cn

Abstract

One of the core components in information retrieval(IR) is the document-term-weighting scheme. In this paper,we will propose a novel learning-based term-weighting approach to improve the retrieval performance of vector space model in homogeneous collections. We first introduce a simple learning system to weighting the index terms of documents. Then, we deduce a formal computational approach according to some theories of *matrix computation* and *statistical inference*. Our experiments on 8 collections will show that our approach outperforms classic $tfidf$ weighting, about 20%∼45%.

Introduction

An IR model is to provide a ranking algorithm which defines an ordering on a set of documents in terms of the degree of relevance between each document and a query. According to (Sager & Jackson 1976), a ranking algorithm has three major components: (1)weighting of terms in query, (2)weighting of terms in the document, (3)relevance(or similarity) measure between query and document. In this paper, we will focus on the second one.

The current term-weighting approaches have two major categories. One is $tfidf$-based, the other is probabilistic-based.

$tfidf$ **weighing schemes** These schemes are based on the hypotheses that the *term frequency*(tf) provides one measure of the intra-document characterization, and the *inverse document frequency*(idf) provides one measure of inter-cluster dissimilarity(Baeza-Yates, Baeza-Yates, & Ribeiro-Neto 1999). Several $tfidf$ based term-weighting approaches were reviewed in (Salton & Buckley 1988), the most classic one is

$$w_{ij} = tf(t_i, d_j) \times \log \frac{m}{df(t_i)}$$

where w_{ij} is a weight associated to the term-document pair $[t_i, d_j]$, $tf(t_i, d_j)$ is the *term frequency* of term t_i in the document d_j,$df(t_i)$ is the *document frequency* of t_i, m is the total number of documents in a collection.

Probabilistic-based schemes These schemes aim at estimating the probability that a term t_i is relevant to a document d_j, i.e., $P_r(t_i \wedge d_j)$. The most famous one is the *language model*(Ponte & Croft 1998), which uses the probability of term t_i under the term distribution for document d_j to estimate $P_r(t_i \wedge d_j)$

$$\hat{P}_r(t_i \wedge d_j) = \begin{cases} \hat{P}_r(t_i|d_j) & \text{if } t_i \in d_j \\ \alpha \hat{P}_r(t_i|DC) & \text{else} \end{cases}$$

However, the above two classic term-weighting approaches are debatable when applied into homogeneous collection, where all documents are about the same topic, and it is more difficult to distinguish them from traditional weighting schemes. Because:

- Some terms, which describe the core concepts of a topic, tend to have relatively higher document frequencies. For example, "oil" and "fry" may have high document frequencies if the collection is about cooking. However, according to $tfidf$ weighting schemes, these important index terms be assigned with low weights. All extensions of $tfidf$ suffer this problem. We call it "idf-problem".
- The challenge of language models is how to estimate the collection probability $P_r(t_i|DC)$, which educes *smoothing* techniques. In (Ponte & Croft 1998), Ponte and Croft remind us that their smoothing technique performs poorly in homogeneous collections. However, the succeeding smoothing techniques, such as what were reviewed in(Zhai & Lafferty 2001), don't consider homogeneous collections.

In this paper we devote to make use of the relations between documents and the co-occurrence of terms. We will propose a novel learning-based term-weighting algorithm and deduce a new term-weighting computational approach as follows

$$w_{ij} = \sum_{k=1}^{n} MI(t_i, t_k) RDF(t_k, d_j) \quad (1)$$

$$MI(t_i, t_k) = \log_2(1 + \frac{df(t_i, t_k)}{df(t_i) df(t_k)})$$

where $df(t_i)$ is the document frequency of t_i. $df(t_i, t_k)$ is the number of documents in which t_i and t_k co-occur. If i =

k, $df(t_i, t_k) = df(t_i) = df(t_k)$. $MI(t_i, t_k)$ is an estimation of the *mutual information*(Church & Hanks 1990) between t_i with t_k. $t_k \in TS$, TS denotes the index term space of the whole collection. n is the total number of terms in the whole collection.

$RDF(t_k, d_j)$ denotes the number of the documents not only relevant to d_j but also containing the term t_k, i.e., the relevant document frequency (RDF) of t_k with regard to d_j. For example, there are d_1, d_2, d_3 and d_4 in the collection which are relevant to d_j . t_k appears in documents d_1, d_2, then $RDF(t_k, d_j) = 2$. It is a measure of the relevance degree of t_k and d_j.

The production of $MI(t_i, t_k)$ and $RDF(t_k, d_j)$ can be considered as a measure of the relevance degree of t_i and d_j with considering a "medi-term" t_k, which forms our approach basis. A term which does not appear in a documents may still be related to the document. Our approach can find the relevance relationships through another medi-term. But only considering few medi-terms is not enough, which is supported by our experimental results (See table 4).

Example 1 simply demonstrates the computational process of our approach.

Example 1. Assume there are a 3-document collection DC = $\{d_1, d_2, d_3\}$. d_1 = $\{satellite\}$, i.e., d_1 has only one term "satellite". d_2 = $\{launch\}$. d_3 = $\{satellite, launch\}$. Let t_1 = "satellite", t_2 = "launch", then

$$df(t_1) = 2, df(t_2) = 2, df(t_1, t_2) = 1$$
$$MI(t_1, t_1) = MI(t_2, t_2) = \log_2(1 + \frac{1}{2}) = 0.5850$$
$$MI(t_1, t_2) = \log_2(1 + \frac{1}{4}) = 0.3219$$

Assume document d_1 is relevant to d_3 [1], but irrelevant to d_2, then

$$RDF(t_1, d_1) = 2, RDF(t_2, d_1) = 1$$

For document d_1

$$w_{11} = MI(t_1, t_1)RDF(t_1, d_1) + MI(t_1, t_2)RDF(t_2, d_1)$$
$$= 0.5850 \times 2 + 0.3219 \times 1 = 1.4919$$
$$w_{21} = MI(t_2, t_1)RDF(t_1, d_1) + MI(t_2, t_2)RDF(t_2, d_1)$$
$$= 0.3219 \times 2 + 0.5850 \times 1 = 1.2288$$

Each term-document pair is assigned a weight. Although "launch" does not appear in d_1, it is still assigned a weight 1.2288. These weights act as "smoothing" technique to improve the performance of our approach.

In the following discussion, we will focus on the deduct of this computational approach. The remains of this paper are organized as follows: Firstly, we introduce a simple learning system to weighting and the definition of our target. Then, we deduce a formal computational approach according to some theories of *matrix computation*(Modi 1989) and *statistical inference*. At last, we apply our approach to improve the retrieval performance of vector space model(VSM). Our experiments will show that our approach outperforms classic $tfidf$ weighting (about $20\% \sim 45\%$) on 8 different homogeneous collections.

[1] We will discuss how to discover relevance relationships between documents later. Obviously, a document is also relevant to itself.

Problem Definition

In homogeneous collections, where exist strong relations between documents. If a term t is important to a document d, it may appear frequently in the relevant documents of d and appear seldom in the irrelevant documents. We target to assign high weights to these terms and low weights to the others. In our work, document d_l is relevant to document d_m if and only if

$$sim(d_l, d_m) = \frac{\overrightarrow{d_l}.\overrightarrow{d_m}}{\sqrt{|\overrightarrow{d_l}||\overrightarrow{d_m}|}} > simTH \quad (2)$$

where $\overrightarrow{d_l} = (w_{1l}, ..., w_{nl})$, w_{il} is weighted according to $tfidf$ scheme. $simTH$ is the threshold.

Suppose we have known: (1)The term set (or contents) of each document, i.e., we know whether a term appear in a document or not. (2)The relevance relationships between documents, i.e., we know whether two documents are relevant or not. The problem will be to determine a proper weight w_{ij} associated with each term-document pair $[t_i, d_j]$ so that the retrieval performance can be improved. This mechanism is somewhat similar with *relevance-feedback*(Harman 1992) which can improve the effectiveness of IR systems. The difference is that we use the relevance relationships between documents.

Weighting Through a Learning Process

We define an *output* function based on *linear unit* (Mitchell 1999) for each document d_j

$$f_{d_j}(x_l) = \overrightarrow{d_j}\overrightarrow{x_l} = w_{1j}x_{1l} + \ldots + w_{nj}x_{nl}$$

where $\overrightarrow{x_l}$ is a n-dimensional vector representing an *example* x_l, $\overrightarrow{x_l}$ = $(x_{1l}, ..., x_{nl})$.

We use some documents as examples to train over each document d_j. Each example-document[2] is marked as x_l, represented by a binary weight vector $\overrightarrow{x_l}$ = $(x_{1l}, ..., x_{nl})$, where

$$x_{il} = \begin{cases} 1 & \text{if } t_i \in x_l \\ 0 & \text{else} \end{cases} \quad (3)$$

We define *target output* for each example x_l as

$$y_{d_j}(x_l) = \begin{cases} 1 & \text{if } x_l \text{ is relevant to } d_j \\ 0 & \text{else} \end{cases} \quad (4)$$

To determine the weight vector $\overrightarrow{d_j}$, the learning process should make the outputs fit the target outputs so as to solve the following *inconsistent equation* (Manna 2003)

$$\begin{bmatrix} x_{11} & \ldots & x_{n1} \\ x_{12} & \ldots & x_{n2} \\ \vdots & \vdots & \vdots \\ x_{1m} & \ldots & x_{nm} \end{bmatrix} \begin{bmatrix} w_{1j} \\ w_{2j} \\ \vdots \\ w_{nj} \end{bmatrix} = \begin{bmatrix} y_{d_j}(x_1) \\ y_{d_j}(x_2) \\ \vdots \\ y_{d_j}(x_m) \end{bmatrix}$$

[2] A document which is used as a training example.

where n is size of index term space. m is the training examples amount. Let X=$[x_{uv}]_{m\times n}$, $\overrightarrow{y}$=$(y_{d_j}(x_1), ..., y_{d_j}(x_m))$, the equation above is equal to

$$X\overrightarrow{d_j}^T = \overrightarrow{y}^T \tag{5}$$

Example 2 The same settings with example 1. We select the whole collection as a sample [3]: $S = \{x_1, x_2, x_3\}$, $x_1 = d_1$, $x_2 = d_2$, $x_3 = d_3$. The training examples are presented as: $\overrightarrow{x_1} = (1, 0)$, $\overrightarrow{x_2} = (0, 1)$, $\overrightarrow{x_3} = (1, 1)$.
For document d_1: $y_{d_1}(x_1) = 1$, $y_{d_1}(x_2) = 0$, $y_{d_1}(x_3) = 1$. Then, to determine the weight vector $\overrightarrow{d_1} = (w_{11}, w_{21})$ is equal to solve the following inconsistent equation

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

To solve the inconsistent equation 5, the RGDS (Random Gradient Descent Search) algorithm devotes to minimize the following *training error*(Mitchell 1999)

$$E(\overrightarrow{d_j}) = \frac{1}{2}\sum_{\forall x_l}(y_{d_j}(x_l) - f_{d_j}(x_l))^2$$

i,e., to solve the following *optimization equation*

$$\overrightarrow{d_j}^T = \arg\min_{\overrightarrow{\beta}^T} \|X\overrightarrow{\beta}^T - \overrightarrow{y}^T\| \tag{6}$$

The RGDS algorithm works as(Mitchell 1999)

$$w_{ij} \leftarrow w_{ij} + \eta(y_{d_j}(x_l) - f_{d_j}(x_l))x_{il}$$

where η is the *learning speed*. This process continues until the *minimum error* is reached. Obviously, the terms which appear frequently in the relevant documents (the target output is 1) will be assigned higher weights, and vice versa.

However, the computational complexity of the process is too high. So we deduce another more efficient computational approach as discussed in the next section.

Deduce a Computational Approach

According to Matrix Theory

As shown in(T. Hastie & Friedman 2001), the solution for the equation 6 is

$$\overrightarrow{d_j}^T = (X^TX)^{-1}X^T\overrightarrow{y}^T \tag{7}$$

To facility our discussion, we assume select the whole collection as the training sample.

(1)Consider the detail of $X^T\overrightarrow{y}^T$ which is a n-dimensional vector. Let $X^T\overrightarrow{y}^T = (r_{1j}, ..., r_{nj})^T$, where r_{ij} is associated with a term-document pair $[t_i, d_j]$

$$r_{ij} = \sum_{l=1}^{m} x_{il} \times y_{d_j}(x_l)$$

The production of x_{il} and $y_{d_j}(x_l)$ is non-zero if and only if $x_{il} = 1$ and $y_{d_j}(x_l) = 1$. $x_{il} = 1$ denotes that $t_i \in d_l$. $y_{d_j}(x_l) = 1$ denotes that d_l is relevant to d_j. Therefore, r_{ij} denotes the number of documents which not only relevant to d_j but also contain t_i. i.e.,

$$r_{ij} = RDF(t_i, d_j) \tag{8}$$

(2)Consider the detail of the matrix X^TX, Let $X^TX = [m_{ij}]_{n\times n}$

$$m_{ij} = \sum_{l=1}^{m} x_{il}x_{jl} = df(t_i, t_j)$$

Let $X^TX = E + D + E^T$ be the splitting of X^TX into its strictly upper triangular, diagonal, and strictly lower triangular part. According to(Fischer *et al.* 1996)(Bietenholz)(Benzi & Tuma 1999)

$$(X^TX)^{-1} \approx (I - L)^{-T}\widehat{D}^{-1}(I - L)^{-1} \tag{9}$$

where I is the *unit matrix*, $L = \omega ED^{-1}$, $\widehat{D} = \omega^{-1}D$, and $0 < \omega < 2$ is a *relaxation parameter*(Modi 1989). Let ω=1, then the element l_{ij} of the matrix L is

$$l_{ij} = \begin{cases} \frac{df(t_i,t_j)}{df(t_j,t_j)} = \frac{df(t_i,t_j)}{df(t_j)} & \text{if } i < j \\ 0 & \text{else} \end{cases}$$

Generally speaking, $df(t_i, t_j) \ll df(t_j)$. Therefore, according to(Benzi & Tuma 1999)

$$(I - L)^{-1} = \sum_{k=0}^{+\infty} L^k = \sum_{k=0}^{n-1} L^k \approx I + L$$

Then

$$(X^TX)^{-1} \approx (I + L^T)\widehat{D}^{-1}(I + L)$$

Let $C = (I+L^T)\widehat{D}^{-1}(I+L) = [c_{ij}]_{n\times n}$, expand this matrix

$$c_{11} = \frac{1}{df(t_1)}, c_{12} = \frac{df(t_1, t_2)}{df(t_1)df(t_2)}, c_{1j} = \frac{df(t_1, t_j)}{df(t_1)df(t_j)}$$

for $i \le j$

$$c_{ij} = \frac{df(t_1, t_i)df(t_1, t_j)}{df(t_1)df(t_i)df(t_j)} + \frac{df(t_2, t_i)df(t_2, t_j)}{df(t_2)df(t_i)df(t_j)} + \ldots + \frac{df(t_{i-1}, t_i)df(t_{i-1}, t_j)}{df(t_{i-1})df(t_i)df(t_j)} + \frac{df(t_i, t_j)}{df(t_i)df(t_j)}$$

According to(Church & Hanks 1990), and C is a symmetrical matrix. Then, for $\forall i, j$

$$c_{ij} \approx \log_2(1 + \frac{df(t_i, t_j)}{df(t_i)df(t_j)}) = MI(t_i, t_j)$$

i.e.,

$$(X^TX)^{-1} \approx [MI(t_i, t_j)]_{n\times n} \tag{10}$$

(3)Integrate equation 7, 8 and 10. For document d_j, its weight vector $\overrightarrow{d_j} = (w_1j, ..., w_nj)$ is

$$\begin{bmatrix} w_{1j} \\ w_{2j} \\ \vdots \\ w_{nj} \end{bmatrix} = \begin{bmatrix} MI(t_1,t_1) & \ldots & MI(t_1,t_n) \\ MI(t_2,t_1) & \ldots & MI(t_2,t_n) \\ \vdots & \vdots & \vdots \\ MI(t_n,t_1) & \ldots & MI(t_n,t_n) \end{bmatrix} \times \begin{bmatrix} RDF(t_1,d_j) \\ RDF(t_2,d_j) \\ \vdots \\ RDF(t_n,d_j) \end{bmatrix}$$

[3] A set of examples.

Then, the weight w_{ij} associated to term-document pair $[t_i, d_j]$ is

$$w_{ij} = \sum_{k=1}^{n} MI(t_i, t_k) RDF(t_k, d_j) \quad (11)$$

In the following discussion we present another deduct of this approach according to *statistical inference*.

According to Statistical Inference

For a document d_j, according to the *inference network*(Turtle & Croft 1991)(Turtle & Croft 1990), the probability of a query q is relevant to d_j can be measured by

$$P_r(q \wedge d_j) = \sum_{k=1}^{n} Pr(q|t_k) Pr(t_k|d_j) Pr(d_j)$$

Let q be a single term t_i

$$P_r(t_i \wedge d_j) = \sum_{k=1}^{n} Pr(t_i|t_k) Pr(t_k|d_j) Pr(d_j) \quad (12)$$

Generally speaking, $\hat{Pr}(d_j) = \frac{1}{|DC|}$, where $|DC|$ is the size of the collection. In the following discussion, we devoted to estimate $Pr(t_i|t_k)$ and $Pr(t_k|d_j)$.

For a document d_j, let p be "the probability of a term t_k appears in the relevant documents of d_j", then

$$P_r(RDF(t_k, d_j) = r) = C_n^r p^r (1-p)^{n-r}$$

where n is the total number of document where t_k appears, i.e., $n = df(t_k)$. This is based on the assumption that all documents are either relevant to d_j or not. We use the estimation of p to estimate $Pr(t_k|d_j)$

$$\hat{P}_r(t_k|d_j) = \hat{p} = \frac{RDF(t_k, d_j)}{df(t_k)} \quad (13)$$

The same to above, let q be the "probability of the term t_i co-occur with t_k inside documents ". We use the estimation of q to estimate the $Pr(t_i|t_k)$

$$\hat{P}_r(t_i|t_k) = \hat{q} = \frac{df(t_i, t_k)}{df(t_i)} \quad (14)$$

Integrate the equation 12, 13 and 14

$$\frac{df(t_i, t_k)}{df(t_i) df(t_k)} \approx \log_2(1 + \frac{df(t_i, t_k)}{df(t_i) df(t_k)})$$

$$\hat{P}_r(t_i \wedge d_j) \approx \frac{1}{|DC|} \sum_{k=1}^{n} MI(t_i, t_k) RDF(t_k, d_j) \quad (15)$$

This is the same to equation 12 except a constant $\frac{1}{|DC|}$. So the w_{ij} computed with equation 1 is a measure of relevance degree of t_i and d_j. This support the analysis about MI and RDF at the beginning of this paper.

We deduce the same computational approach according to two different theories. One is based on machine learning, the other is based on statistical inference. This is not just coincidence. In fact, it has been proved that probabilistic-based IR models can be considered as an application of learning process(Chen 1995).

Efficiency Improvement

As shown in equation 1, we need to consider each term as the medi-term when weighting for a term-document pair. However, this is not necessary. Some terms which appear seldom are negligible when used as medi-terms. Because they co-occur seldom with other terms, i.e., low MI value. The RDF value can not be high if a term appear seldom. These seldom appear terms are useless in statistics which form the basis of our approach.

We just need to select some core-terms as the medi-terms. As mentioned above, these core terms with relatively high document frequency describe the core concepts of a topic. Based on this, we select N terms which with highest document frequency as the "core space": $CS = \{c_1, ..., c_N\}$. The weighting algorithm needs a small change

$$w_{ij} = \sum_{k=1}^{N} MI(t_i, c_k) RDF(c_k, d_j) \quad (16)$$

Apply to Information Retrieval

We apply our approach to VSM. The degree of relevance between the document d_j and the query q is measured by

$$rel(d_j, q) = \sum_{k_i \in q \cap d_j} w_{ij} \quad (17)$$

where w_{ij} is calculated with equation 16.This ranking algorithm is called *inner product measure*(Salton 1971). $k_i \in q \cap d_j$ denotes that, we just need to weighting the terms which appears in d_j, i.e., remove "non-appear" terms.

There also exist other reasons of that we remove non-appear terms: (1) Add new terms into a document can help the users to find this document(improve "recall"), but the inter-cluster dissimilarity will be reduced(decline "precision"). (2)Retrieval over a collection with equation 17, the returned-documents of each test query will be exactly the same with $tfidf$ approach except a different ordering. Our aim is to manifest that our weighting scheme is more accurate than traditional $tfidf$ scheme.

About Normalization

Another popular ranking algorithm of VSM is the *cosine measure*(Salton 1971), which normalizes the weight vector before using the inner product measure. The normalization can improve the retrieval effectiveness. However, normalization in our approach must be executed carefully. Because all terms can be assigned weights to a document, non-appear terms of the document must be took into account. We do as follows

(1)For a document d_j, let $B_{d_j} = CS \cup d_j$, i.e., combine d_j and CS into a new term set.

(2)For a term $t_i \notin B_{d_j}$, $w_{ij} = 0$, otherwise, compute w_{ij} with equation 16.

(3)Normalize the weight vector as follows

$$\hat{w}_{ij} = \begin{cases} \frac{w_{ij}}{\sqrt{\sum_{t_k \in B_{d_j}} w_{kj}^2}}, & \text{if } t_i \in d_j \\ 0, & \text{else} \end{cases}$$

Topic	Docs Num	Terms Num	N	$simTH$
EN	2289	52563	6000	0.12
IE	5315	63132	8000	0.12
IF	1449	29396	4000	0.12
IP	2764	46253	5000	0.12
IR	3127	47318	5000	0.12
LG	2495	56152	6000	0.12
MB	2575	63821	7000	0.12
ST	4852	73659	8000	0.12

Table 1: Statistics of the 8 collections and corresponding parameters setting. where N denotes the size of the concept space, $simTH$ is the relevance-document decision threshold in equation 2

where w_{ij} is computed with equation 16. We label this with "smoothing".

Example 3 In example 1. If we select the core space CS = $\{launch\}$, then $B_{d_1} = \{satellite, launch\}$.
$w_{11} = 0.3219 \times 1 = 0.3219, w_{21} = 0.5850 \times 1 = 0.5850$
After normalization: w_{11} = 0.4821, $w_{21} = 0.8761$.
After remove non-appear term: $\hat{w}_{11}$ = 0.4821, $\hat{w}_{21} = 0$.
If normalize the weight vector without considering the non-appear term "launch". Then $\hat{w}_{11} = 1$, $\hat{w}_{21} = 0$. Because d_1 contain only one term.

Our experiment will show that this smoothing technique can improve the performance of our approach.

Experiments and Results

Test Data

TREC5 We use the 30 queries and related documents in routing task of TREC5 to construct test data. These queries have "domain" information and there are altogether 8 domains. The "domain" is used as topic in our experiments. About 17000 documents belong to these 8 topics according to query-result list. We construct 8 homogeneous collections by using documents with the same topic.

MEDLINE There are 1033 documents and 30 test queries. Each document is an abstract article on medicine.

Experiments Setting

Compare with $tfidf$ on TREC5 Compare our approach with classic $tfidf$ (which is presented at the beginning of this paper) on 8 collections with average 11-point average evaluation. Statistics of these collections and corresponding parameters setting are presented in table 1. The 8 collections are labelled with the abbreviation of their topic name. For example,"IE" is the short for "International Economics". N denotes the size of core space, i.e., we select top N terms from a term list sorted descending according to document frequency.

In addition, we also present the performance of tf-weighting scheme which has no idf-problem on homogeneous collections. But it is too simple to achieve good performance.

Rec	$tfidf$	NoSmo	$MIRDF$	Imp1	Imp2
0.0	0.9095	0.9591	0.9411	+03.48	-01.88
0.1	0.4846	0.6042	0.6397	+32.00	+05.87
0.2	0.3797	0.4596	0.5140	+35.37	+11.84
0.3	0.3207	0.3804	0.4393	+37.02	+15.51
0.4	0.2730	0.3169	0.3748	+37.26	+18.26
0.5	0.2349	0.2705	0.3174	+35.09	+17.35
0.6	0.1879	0.2178	0.2539	+35.09	+16.58
0.7	0.1507	0.1735	0.2006	+33.10	+15.62
0.8	0.1120	0.1281	0.1436	+28.22	+12.13
0.9	0.0706	0.0794	0.0874	+21.29	+10.15
1.0	0.0053	0.0053	0.0058	+09.43	+10.31
avg	0.2846	0.3268	0.3561	+25.15	+08.99

Table 2: Test the impact of smoothing on TREC5-IE collection. "NoSmo" denotes without smoothing

Topic	EN	IE	IF	IP
tf	0.2893	0.2650	0.2600	0.2533
$tfidf$	0.3103	0.2846	0.2756	0.2708
$MIRDF$	0.3922	0.3561	0.4001	0.3590
Imp	+26.40	+25.15	+45.21	+32.55
Topic	IR	LG	MB	ST
tf	0.2478	0.2759	0.2873	0.2734
$tfidf$	0.2601	0.2999	0.3075	0.2989
$MIRDF$	0.3732	0.3834	0.3690	0.3603
Imp	+43.49	+27.86	+19.99	+20.53

Table 3: Compare with $tfidf$ on 8 collections with average 11-point average evaluation.

Test the impact of smoothing on TREC5-IE IE denotes the "International Economics" topic. We test the impact of smoothing on this collection with 11-point average evaluation. In addition, we present the performance of $tfidf$ with 11-point average evaluation.

Test the impact of core space on MEDLINE The total number of terms in MEDLINE is 8702. The size of core space is set as N = {500, 1000, 3000, 5000, 7000}. We test the impact of the selection of core space CS with average 11-point average evaluation.

Results

(1)In table 2, we show the detail recall-precision information on IE collection. The results show a comprehensive improvement in precision compared with $tfidf$ (labelled with "Imp1"). At recall 0.1, improve about 32%. Improve about 25% in average. It is noticeable that, our approach with the ranking algorithm (equation 17) don't put any change to the contents of returned-documents compared with $tfidf$ except their ordering.

The impact of smoothing is positive in this collection (labelled with "imp2"). It improve the precision at every levels of recall except recall 0(decline about 2%) and with 8.99% improvement in average.

(2)In table 3, we compare our approach with $tfidf$ on 8

N	$tfidf$	500	1000	3000	5000	7000
Avg	0.504	0.541	0.564	0.572	0.573	0.574
Imp		+7.4	+11.9	+13.6	+13.7	+13.9

Table 4: Test the impact of the selection of concept space on MEDLINE collection with average 11-point average evaluation(baseline: $tfidf$), where N is the size of core space

homogeneous collections. Our approach outperform $tfidf$ with ranges of 19.99%∼45.21% with average 11-point average evaluation. tf weighting also perform poorly on these homogenous collections although it has no idf-problem.

(3)Table 4 shows that the retrieval performance do not ascend along with the increasing of the size of core space. When the core size $N \geq 1000$, the retrieval performance tend to be stable. $Mean = 0.5706$, $Std.Dev = 0.0047$. However, use a too small core space will reduce the performance. When $N = 500$, the performance declines obviously compared with $N = 1000$. This offers support to that we can not use too few medi-terms.

In MELINE collection, $MIRDF$ also improve the performance, about 13% at most. The performance of $tfidf$ is also considerable in this collection.

In our experiments, the relevance decision threshold $simTH$ is set to 0.12 on all collections. We found the performance of our approach is not very sensitive to this parameter when it locates in a space($0.08 \sim 0.14$).

Conclusion and Future Work

In this paper, we devote to solve the term-weighting challenge in homogeneous collections. By using relations between documents and terms, we could weighting terms through a simple learning process. Then we deduce a computational approach accordingly. We also get the same calculation mechanism according to inference network, which provides evidence of that probabilistic-based IR models can be considered as an application of machine learning. One advantage of our approach is that non-appear terms can be assigned reasonable none-zero weights, while most traditional approaches assign zero or random weights. In order to test the effectiveness of our mechanism, we apply it to 8 collections constructed by TREC5 data. The experimental results prove that it outperforms classic $tfidf$ significantly.

There is still some work left. First, the estimation of $P(t_i|d_j)$ is not proved strictly in this paper. Second, our approach is not appropriate for heterogeneous collections in which RDF values tend to be close for term-document pairs, so that different terms can not be distinguished clearly.

References

Baeza-Yates, R. A.; Baeza-Yates, R.; and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc.

Benzi, M., and Tuma, M. 1999. A comparative study of sparse approximate inverse preconditioners. In *IMACS'97: Proceedings on the on Iterative methods and preconditioners*, 305–340. Elsevier North-Holland, Inc.

Bietenholz, E. Ssor preconditioning of improved actions.

Chen, H. 1995. Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms. *J. Am. Soc. Inf. Sci.* 46(3):194–216.

Church, K. W., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.* 16(1):22–29.

Fischer, S.; Frommer, A.; Glässner, U.; Lippert, T.; Ritzenhöfer, G.; and Schilling, K. 1996. A parallel SSOR preconditioner for lattice QCD. *Computer Physics Communications* 98:20–34.

Harman, D. 1992. Relevance feedback and other query modification techniques. 241–263.

Manna, Z. 2003. *Mathematical Theory of Computation*. Dover Publications, Incorporated.

Mitchell, T. M. 1999. Machine learning and data mining. *Commun. ACM* 42(11):30–36.

Modi, J. J. 1989. *Parallel Algorithms and Matrix Computation*. Oxford University Press, Inc.

Ponte, J. M., and Croft, W. B. 1998. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 275–281. ACM Press.

Sager, W., and Jackson, D. 1976. Classification of ranking algorithms. *International Forum on Information and Documentation*. 1:41–46.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* 24(5):513–523.

Salton, G. 1971. The smart retrieval system-experiments in automatic document processing.

T. Hastie, R. T., and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Verlag.

Turtle, H., and Croft, W. B. 1990. Inference networks for document retrieval. In *SIGIR '90: Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, 1–24. ACM Press.

Turtle, H., and Croft, W. B. 1991. Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.* 9(3):187–222.

Zhai, C., and Lafferty, J. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 334–342. ACM Press.