Expert Systems with Applications 36 (2009) 7985–7991

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

HQE: A hybrid method for query expansion

Lixin Han [a,b,c,*], Guihai Chen [b]

[a] College of Computer and Information Engineering, Hohai University, Nanjing, China
[b] State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China
[c] Department of Mathematics, Nanjing University, Nanjing, China

ARTICLE INFO

Keywords:
Information retrieval
Collaborative filtering
Data mining
Optimization

ABSTRACT

Query expansion methods have been extensively studied in information retrieval. This paper proposes a query expansion method. The HQE method employs a combination of ontology-based collaborative filtering and neural networks to improve query expansion. In the HQE method, ontology-based collaborative filtering is used to analyze semantic relationships in order to find the similar users, and the radial basis function (RBF) networks are used to acquire the most relevant web documents and their corresponding terms from these similar users' queries. The method can improve the precision and only requires users to provide less query information at the beginning than traditional collaborative filtering methods.

1. Introduction

Query expansion methods have a long history in traditional information retrieval. Its study may date back to, at least, the studies of Sparck Jones and Needham (1968), Sparck Jones (1971) in which the collection is analyzed to provide a similarity thesaurus of word relationships. A number of query expansion methods have been proposed. Generally the existing query expansion methods can be classified into the two classes of global analysis and local analysis (Xu & Croft, 2000).

(1) Global analysis is one of the first techniques to produce consistent and effective improvements through query expansion. Global analysis analyzes the entire document corpus to discover word relationships. A known global technique is Latent Semantic Indexing (LSI) (Xu & Croft, 2000). Latent Semantic Indexing (LSI) (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990), which is a way to represent data, has explored the use of semantics for information retrieval and reduces the retrieval and ranking problem to one of significantly lower dimensions. Other global techniques are term clustering, similarity thesauri and Phrasefinder (Xu & Croft, 2000). Generally, global analysis needs to compute term correlation only once at system development time. However, term ambiguity may introduce irrelevant correlated terms.

(2) In contrast to global analysis, local feedback methods (Salton & Buckley, 1990) are generally more effective. However, these local feedback methods are not relatively robust. Local analysis uses only top-ranked retrieved documents for further query expansion. Local analysis needs to compute term correlation for every query at run time.

The above query expansion methods require the systems to analyze a large amount of information that the users provide. In addition, it is too difficult to ensure the completeness and correctness of the existing documents. Collaborative filtering goes some way to addressing these issues. Collaborative filtering (CF) methods (Resnik, Iacovou, Suchak, Bergstrom, & Riedl, 1994) work by assessing similarities among users, then recommending given users more useful documents that like-minded users accessed previously. Therefore users' query expansion information can be acquired from other similar users. However, if the users provide little information at the beginning, collaborative filtering cannot discover any correlations between users and their similar users and further collaborative filtering cannot find their similar users. If external ontological sources including users' personal information can be employed, some similar users may be found. Initial knowledge about all users and their interests can be provided from these external ontologies. Accordingly, based on the idea above, we propose a novel query expansion method called HQE (hybrid query expansion). The HQE method employs collaborative filtering combined with the Semantic Web and neural networks to improve query expansion. The HQE method derives ontology-based user similarity and then finds the similar users, and further constructs the training data of relevant documents retrieved by similar users, at last predict document relevancy and discover the most relevant web documents and their corresponding terms.

The HQE method is more convenient for users to expand new query than the above traditional query expansion methods. Be-

* Corresponding author. Address: College of Computer and Information Engineering, Hohai University, Nanjing, Jiangsu 210098, China.
E-mail address: lixinhan2002@hotmail.com (L. Han).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.10.060

cause of using ontology-based collaborative filtering, the HQE method only requires users to enter some keywords and these users do not require provide a large amount of information that is acquired by analyzing these retrieved documents at the beginning.

2. Related work

Various methods of query expansion have been explored in previous work. Xu and Croft (2000) propose a local context analysis method, which combines both local analysis and global analysis. Expansion terms employed are not based on frequencies in the top-ranked documents, similarly to local feedback, but on co-occurrences with the query terms. Therefore, the local context analysis method can overcome the difficulty of local analysis to some extent. The method is based on the hypothesis that a frequent term from the top-ranked relevant documents will tend to co-occurrence with all query terms within the top-ranked documents. However, the hypothesis is not always true. Cui, Wen, Nie, and Ma (2002) propose a new method for query expansion based on query logs. In the method, probabilistic correlations between query terms and document terms are extracted by analyzing query logs. These correlations are then used to select high-quality expansion terms for new queries. Kamps (2004) explores a new feedback technique that re-ranks the set of initially retrieved documents based on the controlled vocabulary terms assigned to the documents. The approach uses the combination of global and local feedback techniques. On the one hand, the approach uses a global feedback technique to analyze the usage of controlled vocabulary in the collections. The rationale for this is that the approach does not want to rely on the availability of special dictionaries or thesauri. The approach is similar to latent semantic indexing. On the other hand, the approach uses a local feedback technique for re-ranking the set of initially retrieved documents. Han and Chen (2006) use ontology information to extend the keywords and then employ an algorithm for mining association rules to find the terms that are closely related to these keywords and their synonyms. These keywords and their synonyms are restricted by these relevant words to reduce ambiguity and improve precision. Pôssas, Ziviani, Wagner, and Ribeiro-Neto (2002) propose a model referred to as set-based model for computing index term weights and for ranking documents. The components in set-based model are no longer terms, but termsets. The model computes term weights using association rules technique. El-Kahlout and Oflazer (2004) present the design and implementation of a system called Meaning to Word (MTW) to find the appropriate Turkish words that closely match the definition entered by the user. The approach for extracting words based on meanings checks the similarity between the user's definition and each entry of the Turkish database that does not consider the semantics or the context.

Various methods of collaborative filtering have been proposed in previous work. Sarwar, Karypis, Konstan, and Riedl (2001) propose an effective item-based approach to dealing with both the scalability and sparsity problems in the context of collaborative filtering. In this approach, the user-item representation matrix is analyzed to identify the relations between items, rather than relations between users, and then to use these relations to indirectly compute recommendations for users. Because the relations between items are relatively static, item-based algorithms may have less online computation than the user-based algorithms. Huang, Chen, and Zeng (2004) present an effective collaborative filtering approach to exploring relational user and item similarities for dealing with the sparsity problem. They apply associative retrieval techniques and three related spreading activation algorithms including a constrained spreading activation algorithm based on the Leaky Capacitor Model (LCM), a branch and bound serial symbolic search algorithm (BNB), and a Hopfield net parallel relaxation search algorithm (Hopfield) to explicitly generate associations among consumers and products in collaborative filtering. Rashid, Karypis, and Riedl (2005) present a measure of influence that is algorithm-independent, in order to ensure that it can be applied to any ratings-based recommender system. Sarwar, Karypis, Konstan, and Reidl (2000) present and experimentally evaluate an approach to applying the combination of the classical association rule algorithms, nearest-neighbor collaborative filtering algorithms, and algorithms based on dimensionality reduction to CF-based recommender systems. Hofmann (2004) presents a model-based approach to collaborative filtering. The approach introduces a statistical latent class model to discover user communities and build user profiles. In addition, the approach employs overlapping user communities to decompose user preferences. Lemire (2005) presents a scale and translation invariant as being a desirable property for collaborative filtering. The paper takes into consideration such criteria as the amplitude and the mean, and the number of ratings to normalize users, in order to improve accuracy.

Here, we only have space to contrast our work with the most directly related to Middleton, Shadbolt, and De Roure (2004); Hust (2004); Robin and Ramalho (2003); Ciaramita, Hofmann, and Johnson (2003) Akrivas, Wallace, Andreou, Stamou, and Kollias (2002). Middleton et al. (2004) present a general ontology-based recommendation system approach. Their two experimental systems, Quickstep and Foxtrot, are built. Quickstep improves user profiles' accuracy via inference and Quickstep is integrated with an external ontology built from a publication database and personnel database in order to bootstrap a recommender system for easing the cold-start problem. Foxtrot enhances Quickstep by visualizing user profiles in order to acquire direct user profiles' feedback. In contrast to Middleton et al. (2004), the HQE method applies the combination of using an ontology and collaborative filtering to query expansion instead of user profiling within recommender systems. In addition, the HQE method presents some algorithms to improve query expansion. Hust (2004) introduces collaborative filtering techniques to query expansion in a restricted collaborative information retrieval environment. The information retrieval system acquires the relevant documents from previous search processes carried out by one or several users to improve retrieval performance for the current query. The method does not acquire users' knowledge from ontologies. In contrast to Hust (2004), HQE combines Semantic Web with collaborative filtering to perform the analysis of semantic relationships for finding the similar users, and acquires some relevant web documents from these similar users' queries using the Radial Basis Function (RBF) networks. Various uses of ontology as a linguistic knowledge resource to improve IR effectiveness have been tried in previous work. Robin and Ramalho (2003) argue search engines access to wide-coverage linguistic ontologies such as WordNet, can improve web search effectiveness in retrieving a set of purely textual documents with no semantic annotation. They present an experiment that empirically measured the impact on retrieval effectiveness of automatically expanding the query keywords with their synonyms or immediately neighboring concepts. In the experiment, the individual impact of such expansion under environment typical web searches is evaluated. Ciaramita et al. (2003) introduce a hierarchical learning architecture based on the multiclass perceptron for lexical semantic classification problems that integrates task-specific and general information. They propose to generate additional training data by extracting training sentences from a dictionary ontology-WordNet. Akrivas et al. (2002) propose a query expansion method, which takes into account the query context based on the Inclusion relation. The query context is defined as a fuzzy set of semantic entities. They integrate the method with the user's profile. The user's preferences are used to change this context to provide the capability for query personalization. In contrast to Robin and Ramalho (2003), Ciaramita et al. (2003), and Akrivas et al. (2002), the HQE method uses ontology information to find the similar users, acquires the relevant web documents from these similar users' queries, and the relevant web documents are used by the RBF networks in order to expand new query.

In contrast to the above work, the HQE method employs ontology-based collaborative filtering combined with neural networks to improve query expansion. In addition, the HQE method presents some algorithms to improve query expansion. In the HQE method, collaborative filtering is used to analyze semantic relationships that are acquired from the constructed ontologies in order to find the similar users, and the radial basis function (RBF) networks are used to acquire the most relevant web documents and their corresponding terms from these similar users' queries.

3. The HQE method of improving query expansion

In this section, we propose a new query expansion method called the HQE method. The HQE method consists mainly of the CCIO (combines complete-link clustering with inference for constructing of ontologies) algorithm, the BBFU (branch and bound for finding similar users) algorithm and the RENQ (RBF to expand new query) algorithm. The CCIO algorithm is described in Section 3.1, The BBFU algorithm is described in Section 3.2, and the RENQ algorithm is described in Section 3.3.

The HQE Method can be described as the following steps:

{the CCIO algorithm is used to construct ontologies;
 based on the above ontologies, the BBFU algorithm is used to perform the analysis of semantic relationships for finding the similar users;
 some relevant web documents are acquired from the above similar users' queries;
 the RENQ algorithm is used to rank these web documents and discover the most relevant web documents and their corresponding terms;
}

3.1. The CCIO algorithm for constructing ontologies

An ontology is an explicit specification of a conceptualization (Gruber, 1993). Form a human standpoint, an ontology is a conceptualisation of a domain; Form a machine standpoint, it consists of entities, attributes, relationships, and axioms (Guarino & Giaretta, 1995).

The ontologies are constructed to identify the association relationships between concepts that can be extracted by users' personal information, and to be shared among all users. In addition, the existing relationships in the knowledge base provide a scope for discovering new relationships. For example, if such entities as teaching, institution, staff, project, student, and paper are constructed, we can expect to find all users' university, department, research interests, research projects, publications, course, graduate student etc. Accordingly, based on the idea above, we propose the CCIO algorithm for automatically constructing ontologies. The CCIO algorithm can exploit the desirable properties of both complete-link clustering algorithms and inference mechanism to construct the hierarchically structured ontologies, that is, based on the existing association entities acquired from the Complete-link Clustering algorithm, new meaningful relationships that cannot directly observed can be inferred through Jena2's inference mechanism.

Formula (1) denotes the similarity between two entities represented as sets A and B.

$$S(A,B) = |A \cap B| / |A \cup B| \tag{1}$$

where a^x is an interest of user x, b^y is an interest of user y, $||$ is the cardinality of a set, and $A = \{u_{a1}, \ldots, u_{an}\}$ and $B = \{u_{b1}, \ldots, u_{bn}\}$, where $u_{a1}, \ldots, u_{an}$ are the terms of a^x, and $u_{b1}, \ldots, u_{bn}$ are the terms of b^y. The larger $S(A,B)$ is, the larger the similarity between A and B is.

The CCIO algorithm can be described as follows:

{ $i = 0$;
 using formula (1), calculate the proximity matrix D_i containing the distance between a^x and b^y; treat each interest as a cluster, that is $D_{pq} = d_{pq}$; // D_{pq} is the distance between each pair of clusters, d_{pq} is the distance between each pair of terms
 while all terms are not in one cluster
 { $i = i + 1$;
  find the most similar pair of D_{pq} in the proximity matrix and merge these two clusters C_p and C_q into one cluster C_r, that is $C_r = \{C_p, C_q\}$; //C_r, C_p, C_q is cluster
  using formula (1) and the formula $D_{rk} = \max\{D_{pk}, D_{qk}\}$, calculate the proximity matrix D_i to perform a merge operation;
  a class hierarchy can be constructed;
  generate some rules from the class hierarchy;
  store these rules into the knowledge base;
  the widely used Jena2's inference mechanism (Wilkinson, Sayers, Kuno, & Reynolds, 2003 & Carroll et al., 2003) is used to infer semantic associations from the existing rules within the knowledge base in order to discover more meaningful association entities;
  a class hierarchy can be reconstructed;
 }
}

In contrast to the widely used complete-link clustering algorithm (Jain, Murty, & Flynn, 1999), the CCIO algorithm provides the inference mechanism to find more useful association entities that are not directly observed in the users' behaviour.

3.2. The BBFU algorithm for finding the similar users

Branch and bound algorithms are methods for global optimization in non-convex problems (Lawler & Wood, 1966 & Moore, 1991). They are non-heuristic, in the sense that they maintain a provable upper and lower bound on the globally optimal objective value; they terminate with a guarantee proving that the indeed suboptimal point found is suboptimal. In the worst case they require effort that grows exponentially with problem size, but in some cases we are lucky, and the methods converge with much less effort. The methods are useful to solve small instances of hard problems.

We propose an efficient branch and bound search algorithm BBFU that introduces a branch and bound method to find the similar users. The BBFU algorithm employs optimization techniques to evaluate the nodes close enough to the specified node. Specifically, the BBFU algorithm examines the connectivity of entities in ontologies, traverses the semantic relationships between entities before a given threshold is reached, according to their semantic distance, identifies a set of close entities, and finds the similar users.

In the following description, k stands for the iteration index. $List_k$ denotes the list of nodes. U_k denotes the upper bound, at the end of k iterations. x^* denotes the optimal solution. The bounding function $F(x) = \max(P)$. The Pearson correlation $P = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{(n-1)S_X S_Y}$, where X and Y are two entities vectors, $\bar{X}$ and $\bar{Y}$ are their means, respectively, S_X and S_Y are their standard deviations, respectively.

The Pearson correlation is often used to the similar entities whose features are known.

The BBFU algorithm can be described as the following steps:

{ given ontologies and a user;
$x^* = \emptyset$;
$k = 0$;
$List_0$ = {root node};
$U_0 = +\infty$;
do {maximizing choose the class node x with bounding function $F(x)$, where $x \in List_k$;
$List_{k+1} = List_k - \{x\}$;
if $f(x) \geqslant U_k$
then prune the node;
else {$x^* = x$;
$U_k = f(x)$;
split x into x_1 and x_2;
$List_{k+1} = List_k \cup \{x_1, x_2\}$;
}
$k = k + 1$;
}
while $List_k \neq \emptyset$ and $U_k - f(x) > \varepsilon$
determine optimal solution x^*, that is find the most similar user;
a set of the similar users are identified from the most similar user' subclass and superclass in ontologies;
}

3.3. The RENQ algorithm for expanding new query

The RENQ algorithm introduces RBF networks to rank web documents and discover the most relevant web documents and their corresponding terms. In the RBF networks, the relevant web pages are obtained from a set of the similar users. The algorithm is based on the hypothesis that two web pages are more similar in content if there are more common keywords in their web pages. The algorithm begins by training known relevant and irrelevant web documents. Relevant web documents are obtained from a set of the similar users and irrelevant web documents are obtained from a set of the dissimilar users. During training, the connection weights of the RBF networks are initialized with some random values. The training samples in the training set are then presented to the RBF networks in random order and the connection weights are adjusted using the SC method. This process is repeated until the least-squares error between the predicted outputs from the network and the observed output is small or the predefined maximum iterative number is achieved. After network training has finished, the neural network is built using keywords from these web pages. Each keyword maps into a 0 or 1, so the length of the input vector equals the number of the keywords chosen. The neural network can assign all the web pages to a relevancy value that is a fraction between 1 and 0. These relevant values may be used to rank these relevant web pages.

The RENQ algorithm can be described as follows:

{the training set is obtained from a set of the similar users and the dissimilar users;
cstep = 0; // *cstep* is the current iterative step;
While (the least-squares error > *tnumber*) and *step* < *maxstep*
// *tnumber* is a predefined tolerance number, *maxstep* is a iterative maximum number;
{using the following the SC method, the training samples are trained;
step = *step* + 1;
}
using the trained neural networks, rank a set of web documents acquired from these existing similar users' queries, and discover the most relevant web documents and their corresponding terms;
}

In contrast to the widely used RBF Networks (Yang & Zheng, 2003 & Ibnkahla, 2000), a SC method has been proposed to update RBF networks.

3.3.1. The SC method for updating RBF networks

The SC (Combine SACU with CTSB) method has been proposed to update RBF networks. The method is composed of the SACU algorithm for centers update and the CTSB algorithm for the output weights update.

For clarity of presentation, the basic concepts about RBF networks are briefly recalled that are relevant to this paper.

The RBF network (Ibnkahla, 2000) is a two-layer feedforward network, whose input-layer activation function can be described as follows: $f(x) = \phi(\|x\|)$, where ϕ is a radial basis function. $\|\cdot\|$ is the Euclidean norm on R^n. In contrast to MLNN (multi-layer neural network), the activation function is a compact one, instead of the usual sigmoid function. The activation function can be described as follows: $\phi(T) = \frac{1}{\sqrt{2\pi}\delta}\exp(-T)$, $T = \frac{(\|x-c_k\|)^2}{2\delta^2}$. The outputs of the first layer neurons are described as follows: $x_{1k} = \frac{1}{\sqrt{2\pi}\delta}\exp\left(-\frac{\|x-c_k\|^2}{2\delta^2}\right)$, where x is the input vector and c is the weight associated to neuron k. The network outputs are described as follows: $y_j = \sum_{k=1}^{N_1} w_{kj}\frac{1}{\sqrt{2\pi}\delta}\exp\left(-\frac{\|x-c_k\|^2}{2\delta^2}\right)$, where w_{kj} is the weights associated with the output layer.

3.3.1.1. The SACU algorithm for centers update. Simulated annealing is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953). The concept is based on the manner in which liquids freeze or metals recrystalize in the process of annealing. By analogy the generalization of this Monte Carlo approach to combinatorial problems is straight forward (Cerny, 1985; Kirpatrick, Gellat, & Vecchi, 1983). The major difficulty in implementation of the algorithm is that there is no obvious analogy for the temperature T with respect to a free parameter in the combinatorial problem. Furthermore, avoidance of entrainment in local minima is dependent on the "annealing schedule", the choice of initial temperature, how much iterations are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds (Gray et al., 1997).

We propose the SACU (Simulated Annealing to Centers Update) algorithm to update centers. The SACU algorithm introduces simulated annealing for centers update. The simulated annealing algorithm can be viewed as globally optimizing an objective function. The SACU algorithm can ensure to break away current local optimal solution and to reduce the computing workload.

The SACU algorithm is based on the following solution form:

1. The solution space is a training set;
2. The objective function is $f = \min\sum_{r=1}^{k} S_r = \min\sum_{r=1}^{k}\sum_{i=1}^{n_r}(x_{ri} - \bar{x}_r)'(x_{ri} - \bar{x}_r)$, where k is the number of clusters, n_r is the number of training samples in cluster r, S_r is the squared error, x_{ri} is training sample i in cluster r, and x_r^- is the centroid of cluster r;
3. A new solution is produced by randomly choosing another training sample from neighboring region $N(x)$;

4. The objective function difference is $\Delta f = f$ (new training sample)-f (local optimal solution);
5. The accept criterion is $P = 1$ for $\Delta f > 0$ or $P = \exp(-\Delta f/t)$ for $\Delta f \leqslant 0$.

The SACU algorithm can be described below:

{given a set of the training samples, initial solution x_0, initial optimal objective function value f^*, initial temperature t_0, initial step q_0, monotone increment function $I(x)$;
  $x^* = x_0$; // initialize optimal solution x^*
  $q = q_0$; // initialize step q
  $t = t_0$; // initialize temperature t, $t_0 = 100$
  $i = 1$; // initialization
  $at = 1$; // initialization
  while $|R_i - C| > \varepsilon$
  // ε is a given enough small positive number, where $\varepsilon = 2 \times 10^{-3}$. C is an threshold, where $C = 0.75$
  { $R_{i-1} = at/as$; // at is the acceptable times of new solution, as is iterative steps number, R_{i-1} is acceptable rate
    $at = at + 1$;
    $as = 0$; // initialization
    while $J <> U$//U is the upper limit of step length
    { $as = as + 1$;
     while $x \notin N(x)$//$N(x)$ is neighboring region
      {step length is q and the BFGS optimizer in the quasi-Newton method is used to compute local optimal solution;
      }
    x_i is produced by randomly choosing another training sample from neighboring region $N(x)$ and objective function difference $\Delta f = f(x_i) - f(x)$ is computed;
   if $\Delta f \leqslant 0$ or $\exp(-\Delta f/t) > \text{random}(0,1)$
   then {$x = x_i$;
        if $f < f^*$
      then {$x = x^*$;
          $f = f^*$;
          }
      if at is equal to the given acceptable times
      then break;
      }
   $q = I(R_i) * q$;
   // adjustment function $I(x)$ is monofonic increasing function, where $I(x) = (x\text{-}0.5)^2 + 1$
   }
  if $f < f^*$
  then {$x = x^*$;
      $f = f^*$;
      }
  $R_i = at/as$;
  if $i > 2$
  then // compute self-adjusting temperature
    {**if** $R_{i-1} < C$ and $R_i < C$
    then $t = t$+TC;// TC is a given constant
    if $R_{i-1} \geqslant C$ and $R_i \geqslant C$
    then $t = t$-TC;
    if $R_{i-1} \leqslant C$ and $R_i \geqslant C$
    then$t = t\text{-}TC/2$;
    if $R_{i-1} > C$ and $R_i < C$
    then $t = t + TC/2$;
    }
  $i = i + 1$;
  }
}

In contrast to the widely used simulated annealing algorithm (Jonhson, Aragon, Mcgeoch, & Schevon, 1991; Kirpatrick et al., 1983), there are three features in the SACU algorithm. Firstly, the SACU algorithm can self-adaptingly adjust temperature. Therefore it is regarded as time after time optional process. On the one hand, the Boltzmann factor $\exp(-\Delta f/t)$ increases with t. This is useful to break away from the current local optimal solution and to increase step length in order to look for a better new solution. On the other hand, when acceptable rate meets the given condition, the program can end so as to reduce the computing workload. Secondly, local optimization techniques instead of choosing randomly a new solution are employed in order to compute a local optimal solution. Thirdly, the local optimal solution is recorded during annealing process, in order to protect the current optimal solution from being thrown away.

3.3.1.2. The CTSB algorithm for updating the output layer weights. We propose the CTSB (combine TS with BFGS) algorithm to update the output layer weights. The CTSB algorithm is a hybrid algorithm of exploiting the desirable properties of both the widely used globally optimal algorithms Tabu Search (Glover, 1986; Gray et al., 1997) and locally optimal algorithms BFGS (Xie, Han, & Lin, 1997).

The CTSB algorithm can be described below:

{x_j is normalized, that is, $x'_j = a_j x_j + b_j$, where $a_j = \frac{2}{x_{j\max} - x_{j\min}}$, $b_j = 1 - a_j x_{j\max}$, and $x_j \in (x_{j\min}, x_{j\max})$;
  $count = 1$;
  while $count < maxno$ // $maxno$ is the iterative maximum $number$
   {$\Delta W_{ijk}(n-1) = W_{ijk}(n) - W_{ijk}(n-1)$, $n = 0,1,\ldots$;// $W_{ijk}(n)$ is a weight vector
  $\Delta W_{ijk}(n) = -\mu \frac{\partial E(W_{ijk}(n))}{\partial W_{ijk}(n)} + v\Delta W_{ijk}(n-1), n = 0,1,\ldots$; // μ is the learning rate, v is a law of inertia coefficient
  if $\sum_{n=0}^{k_2-1}(E(W_{ijk}(n+1)) - E(W_{ijk}(n))) > \lambda E(W_{ijk}(n_1))$// move a local optimal solution to a better local optimal solution, where λ is a given appropriate small positive $number$ between 0.15 and 0.25, k_2 is a current value, and $W_{ijk}(n_1)$ is a initialized weight vector, the squared error energy function $E(n)$ is the error power between the network output vector at time n and the desired output
  then the BFGS algorithm is used to determine a local optimal solution and the weights are updated;
  if $\frac{|\Delta E(W_{ijk}(n))|}{E(W_{ijk}(n))} < \varepsilon$ and $E(W_{ijk}(n)) > L$ // indulge in a local optimal solution, where L is a given appropriate large number, and ε is a given enough small positive number
  then the Tabu Search algorithm is used to break away current local optimal solution and the weights is updated;
  $count = count + 1$;
  }
}

In contrast to the supervised learning rules BP (Ibnkahla, 2000; Yang & Zheng, 2003) for the output weights update in RBF networks, the SACU algorithm exploit the desirable properties of both Tabu Search and BFGS. Therefore the SACU algorithm is easy to find global optimal solution and to bring a less the squared error energy. In contrast to the BP algorithm (Ibnkahla, 2000), $\Delta W_{ijk}(n)$ in the SACU algorithm consider not only learning rate but also a law of inertia coefficient in order to have faster convergence at iterative steps and avoid fluctuating fiercely around convergence point.

4. Experimental results and discussion

In this section, we present the experiments results of the HQE method. The HQE method employs OWL (Dean & Schreiber, 2004; McGuinness & Harmelen, 2004) to describe resources and their inter-relations. Jena2's inference mechanism is used to infer semantic associations from the existing rules. We select 251 Web pages to acquire ontology information. These selected Web pages are related to personal information. Some entities are extracted from these Web pages and stored in the OWL files. Jena2's inference mechanism allows additional facts to be inferred from these entities. Inference mechanism uses forward chaining, backward chaining and a hybrid execution model provided by the Jena2 reasoner. The connectivity of entities in ontologies is examined, and the semantic relationships between entities are traversed before a given threshold is reached, and according to their semantic distance, a set of close entities are identified, and eighteen similar users are found. Some relevant web documents are acquired from these similar users' research interests. After training these relevant web documents, a set of term for query expansion are discovered. We submit the user's queries that are related to our research to the existing third party search engine Google. The search engine results are the ten top-ranked retrieved documents. These queries belong to such topics as information retrieval, data mining, pattern recognition, semantic web, optimization and network computing. The query results are divided into relevant and irrelevant documents. The ranked results are analyzed to check whether the relevant results are contained in the answer, in order to meet the need of a quantitative measure. We adopt mean average precision (MAP) to evaluate the HQE retrieval performance. Mean average precision, which is a standard IR evaluation measure, is used to find the mean of the average precisions over a set of queries, where average precision is the mean of the precision scores after each relevant document is retrieved. Hust's method is the most relevant to our work. We compare the results of the HQE method with the results of Hust's method (Hust, 2004) and entering a set of keywords without using the HQE method for six topics. Six topics are closely related to these similar users. Six topics are described below: topic 1 is information retrieval, topic 2 is data mining, topic 3 is pattern recognition, topic 4 is semantic web, topic 5 is optimization and topic 6 is network computing. Mean average precision of these methods for six topics are shown, respectively, in Fig. 1. The experiment result in Fig. 1 shows that the HQE method can perform better in topic 1, topic 2, topic 4, and topic 5. The experiment result shows that if a set of term for query is scientific terms without ambiguity, these methods led to similar higher mean average precision, and if a set of term for query suffer the usual problem of synonymy and ambiguity, that HQE can perform better. Hust's method does not take into consideration acquiring relevant knowledge from ontologies. Hust's method is a method of introducing collaborative filtering techniques to query expansion. The method can help to improve query expansion. However, if similar users provide too many terms for a query and this causes too many expansion terms, the method can not improve query expansion. In contrast to Hust's method, the HQE method can acquire the most relevant terms from these similar users' queries and reduce the number of expansion terms. For example, we look for ranking algorithm in information retrieval. When keyword ranking is entered, such useless information as best college rankings and best graduate schools rankings could be found from query results. Thus, its average precision is only 0.297. After Hust's method is used, its average precision is 0.757. After the HQE method is used, its average precision is 0.953. It is because ranking algorithm can be acquired from ontology information. Some similar users in ontology user do research in information retrieval area. Thus, ranking may be regarded as a kind of ranking algorithm in ontology information retrieval. Mean average precision of these methods for all tops are shown, respectively, in Fig. 2. The experiment result in Fig. 2 shows that HQE can perform better.

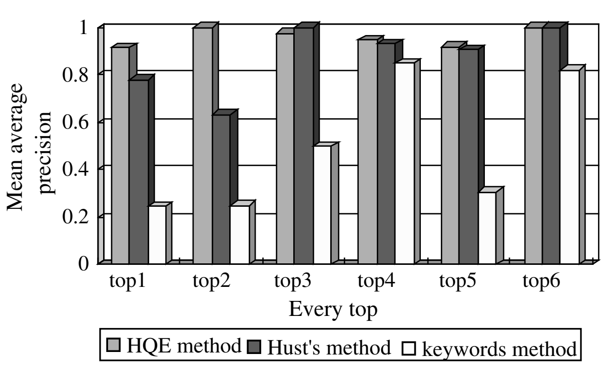

Fig. 1. MAP f or every top.

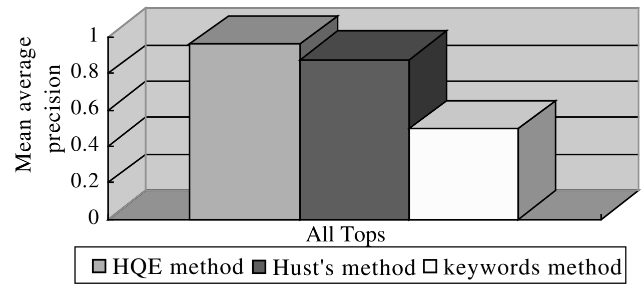

Fig. 2. MAP f or all tops.

5. Conclusion

Nowadays, the amount of information in the Internet is increasing dramatically. Facilitating users to get useful information has become more and more important to information retrieval systems. The traditional query expansion methods require users to analyze a large amount of information. In addition, it is too difficult to ensure the completeness and correctness of the existing documents. To solve these problems, in this paper, we propose a query expansion method called HQE. In the HQE method, the CCIO algorithm is proposed to construct ontologies. The BBFU algorithm is proposed to analyze semantic relationships for finding the similar users. The RENQ algorithm is proposed to acquire the most relevant web documents and their corresponding terms from these similar users' queries. The method can improve the precision and only requires users to provide little query information at the beginning than traditional collaborative filtering methods.

Acknowledgements

The authors thank Professor Linping Sun for technical help. This work is supported by the National Natural Science Foundation of China under Grants 60673186, 60571048 and 60736015, the national 863 program under Grants 2007AA01Z178, and the State Key Laboratory Foundation of Novel Software Technology at Nanjing University under Grant A200604. The work is partly supported by China NSF grants (60573131, 60673154, 60721002, 60825205), Jiangsu High-Tech Research Project of China (BG2007039), and China 973 project (2006CB303000).

References

Akrivas, G., Wallace, M., Andreou, G., Stamou, G., & Kollias, S. (2002). Context-sensitive semantic query expansion. In *2002 IEEE international conference on artificial intelligence systems (ICAIS'02)*.

Carroll, J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., & Wilkinson, K. (2003). *The Jena semantic Web platform: Architecture and design*. HP Laboratories Technical Report HPL-2003-146.

Cerny, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications, 45*(1), 41–51.

Ciaramita, M., Hofmann, T., Johnson, M., 2003. Hierarchical semantic classification: Word sense disambiguation with world knowledge. In *IJCAI 2003* (pp. 817–822).

Cui, H., Wen, J.-R., Nie, J.-Y., & Ma, W.-Y.(2002). *Probabilistic query expansion using query logs www.2002*. (pp. 325–332).

Dean, M., & Schreiber, G. (2004). *OWL Web ontology language reference. W3C recommendation* 10 February 2004. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS, 41*(6), 391–407.

El-Kahlout, I. D. & Oflazer, K. (2004). Use of Wordnet for retrieving words from their meanings. In *Proceedings of the global Wordnet conference (GWC2004)*, (pp. 118–123).

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 13*(5), 533–549.

Gray, P., Hart, W., Painton, L., Phillips, C., Trahan, M., & Wagner, J. (1997). *A survey of global optimization methods*. <http://www.cs.sandia.gov/opt/survey/main.html>.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199–220.

Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. In N. Mars (Ed.), *Towards very large knowledge bases: Knowledge building and knowledge sharing* (pp. 25–32). IOS Press.

Han, L., & Chen, G. (2006). The HWS hybrid web search. *Information and Software Technology, 48*(8), 687–695.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS), 22*(1), 89–115.

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems, 22*(1), 116–142.

Hust, A. (2004). Introducing query expansion methods for collaborative information retrieval. *Reading and Learning*, 252–280.

Ibnkahla, M. (2000). Applications of neural networks to digital communications – A survey. *Signal Processing, 80*(2000), 1185–1215.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.

Jonhson, D. S., Aragon, C. A., Mcgeoch, L. A., & Schevon, C. (1991). Optimization by simulated annealing: an experimental evaluation – Part II (graph coloring and number partition). *Operations research, 39*(3), 378–406.

Kamps, J. (2004). Improving retrieval effectiveness by reranking documents based on controlled vocabulary. *ECIR 2004*, 283–295.

Kirpatrick, S., Gellat, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 4598*, 671–680.

Lawler, E. L., & Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations Research, 14*, 699–719.

Lemire, Daniel (2005). Scale and translation invariant collaborative filtering systems. *Information Retrieval, 8*(1), 129–150.

McGuinness, D. L., & Harmelen, F. v. (2004). *OWL Web ontology language overview. W3C recommendation* 10 February 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics, 21*, 1087–1091.

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS), 22*(1), 54–88.

Moore, R. E. (1991). Global optimization to prescribed accuracy. *Computers and Mathematics with Applications, 21*(6/7), 25–39.

Pôssas, B., Ziviani, N., Jr., Wagner, M., & Ribeiro-Neto, B. A. (2002). Set-based model: A new approach for information retrieval. *SIGIR 2002*, 230–237.

Rashid, Al., Karypis, G., & Riedl, J. (2005). Influence in ratings-based recommender systems: An algorithm-independent approach. In *Proceedings of SIAM international conference on data mining, 2005*.

Resnik, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 conference computer supported cooperative work* (pp. 175–186). ACM Press.

Robin, J., Ramalho, F., 2003. Can ontologies improve web search engine effectiveness before the advent of the semantic web? In *SBBD 2003* (pp. 157–169).

Salton, Gerard, & Buckley, Chris (1990). Improving retrieval performance by relevance feedback. *JASIS, 41*(4), 288–297.

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the ACM conference on electronic commerce* (pp.158–167). New York: ACM.

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. T. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international World Wide Web conference* (pp. 285–295).

Sparck Jones, K. (1971). *Automatic keyword classification for information retrieval*. London: Butterworths.

Sparck Jones, K., & Needham, R. M. (1968). Automatic term classification and retrieval. *Information Processing & Management, 4*(1), 91–100.

Wilkinson, K., Sayers, C., Kuno, H. A., & Reynolds, D. (2003). Efficient RDF storage and retrieval in Jena2. In *The first International Workshop on Semantic Web and Databases, Co-located with VLDB 2003 (SWDB'03), Humboldt-Universität, Berlin, Germany, September 7–8, 2003* (pp. 131–150).

Xie, K., Han, L., & Lin, Y. (1997). *Optimization method*. Tientsin: University Press.

Xu, J., & Croft, B. W. (2000). Improving the effectiveness of informational retrieval with local context analysis. *ACM Transactions on Information Systems, 18*(1), 79–112.

Yang, X., & Zheng, J. (2003). *Artificial neural network and blind signal processing*. Tsinghua University Press.