Information Processing and Management 49 (2013) 441–453

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

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

High performance query expansion using adaptive co-training

Jimmy Xiangji Huang *, Jun Miao, Ben He

Information Retrieval and Knowledge Management Research Lab, School of Information Technology, York University, Toronto, Canada

ARTICLE INFO

Article history:
Received 11 April 2010
Received in revised form 6 August 2012
Accepted 27 August 2012
Available online 21 December 2012

Keywords:
Co-training
Query expansion
Relevance feedback

ABSTRACT

The quality of feedback documents is crucial to the effectiveness of query expansion (QE) in ad hoc retrieval. Recently, machine learning methods have been adopted to tackle this issue by training classifiers from feedback documents. However, the lack of proper training data has prevented these methods from selecting good feedback documents. In this paper, we propose a new method, called AdapCOT, which applies co-training in an adaptive manner to select feedback documents for boosting QE's effectiveness. Co-training is an effective technique for classification over limited training data, which is particularly suitable for selecting feedback documents. The proposed AdapCOT method makes use of a small set of training documents, and labels the feedback documents according to their quality through an iterative process. Two exclusive sets of term-based features are selected to train the classifiers. Finally, QE is performed on the labeled positive documents. Our extensive experiments show that the proposed method improves QE's effectiveness, and outperforms strong baselines on various standard TREC collections.

1. Introduction

Relevance feedback improves the query representation by taking feedback information into account via query expansion. A classical relevance feedback algorithm was proposed by Rocchio (1971) for the SMART retrieval system (Salton, 1971). It takes a set of documents as the feedback set. Unique terms in this set are ranked in a descending order of *tf.idf* weights. A number of top-ranked expansion terms are then added to the query, and finally documents are returned for the expanded query.

Many other relevance feedback techniques and algorithms have been developed, mostly derived from Rocchio's algorithm (Amati, 2003; Carpineto, de Mori, Romano, & Bigi, 2001; Miao, Huang, & Ye, 2012; Robertson, 1990; Robertson, Walker, Hancock-Beaulieu, Gatford, & Payne, 1995; Ye, He, Huang, & Lin, 2010). In addition, the relevance-based language model makes use of feedback information by estimating the generation probability of both the query and the feedback documents from a latent relevance model (Lavrenko & Croft, 2001). The feedback documents can be obtained by many possible means. In general, some methods utilize explicit evidence, such as labeled relevant documents from real users, while others use implicit evidence, such as the click-through data. Obtaining feedback information involves extra efforts, e.g. relevance judgment by real users, which is usually expensive. For every given query, the corresponding feedback information is not necessarily available. An alternate solution is *pseudo-relevance feedback* (PRF), which uses the top-ranked documents in the initial retrieval as relevant documents for the feedback (Buckley, Salton, Allan, & Singhal, 1994; Efthimiadis, 1996; Lynam, Buckley, Clarke, & Cormack, 2004; Mitra, Singhal, & Buckley, 1998; Xu & Croft, 1996, 2000). PRF is an effective technology for *query expansion* (QE). The basic idea of QE based on PRF is to extract expansion terms from the top-ranked documents to formulate a new query for the second round retrieval. Through QE, some relevant documents missed in the initial round can then be retrieved to improve the overall performance. Particularly, QE refers to QE based on PRF in the rest of this paper.

* Corresponding author.
E-mail addresses: jhuang@yorku.ca (J.X. Huang), jun97@yorku.ca (J. Miao), benhe@yorku.ca (B. He).

0306-4573/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ipm.2012.08.002

Despite the marked improvement over the initial retrieval performance (Amati, 2003; Robertson et al., 1995), QE can also fail. There have been many studies on QE's effectiveness. For example, a wide range of predictors were proposed to indicate the query performance, which is usually correlated with QE's effectiveness (Amati et al., 2004; Carpineto et al., 2002; He and Ounis, 2009b). All these previous studies have agreed on a conclusion that the quality of feedback documents is crucial to QE's performance. Since feedback documents are not assessed by real users when using QE, the quality of the feedback document set is not guaranteed. In particular, a feedback document may not be useful even if it is relevant to the query topic – the document could be just partially relevant, and there might be only a small part of the document that is about the query topic, while the rest of the document is irrelevant. In this case, off-topic expansion terms are added to the query, leading to degraded retrieval performance. *Relevance* is not enough to judge whether the document can help QE to improve the performance in such a scenario. Thus, we use "quality" instead of "relevance" to evaluate a feedback document. Particularly, a "good/positive" quality feedback document is not only completely relevant but also effective in improving the final performance of QE. Hence, there is a crucial need for selecting good quality feedback documents (He and Ounis, 2009b; Ye et al., 2011). Details about the *quality* of a feedback document are described in Section 6.2.

Recently, there have been efforts in applying machine learning methods to find good feedback documents or expansion terms (Cao et al., 2008; Lee et al., 2008; He and Ounis, 2009a). However, the potential effectiveness of learning techniques in improving QE may not be fully exploited due to the difficulty in learning classifiers with only few proper training data, i.e. the labeled feedback documents in our case. To solve the problem of traditional QE algorithms, which leads to overfitting because of the limited amount of training data and large term space, Xu and Akella (2008) propose an online Bayesian logistic regression algorithm to select good feedback documents for QE. However, human efforts are involved to update the features in an iterative way which adds extra costs to its implementation and applications.

In this paper, we propose to incorporate co-training into a retrieval system in an adaptive manner at the feedback stage. The proposed method is called adaptive co-training (AdapCOT) in this paper. Co-training is an effective method for utilizing unlabeled data for classification, which we believe is suitable for finding good feedback documents with only limited training data available. In co-training, the input data is used to create two predictors that are complementary to each other. Each classifier is then used to classify unlabeled data. After that, the classified data are used to train the other complementary classifier. Typically, the complementarity is achieved either through two redundant views of the same data (Blum and Mitchell, 1998) or through different supervised learning algorithms (Goldman and Zhou, 2000). In our proposed approach, after the initial search, we create a training data set by choosing the top-ranked documents from the ranked list as positive examples and the other set of documents from the bottom of the list as negative examples. The remaining documents are treated as the test set. The top-ranked documents are usually regarded as highly relevant, and have a general aboutness of the query topic throughout them. Features representing the documents are split into two exclusive sets. Then, we classify the examples in the test set, label them, and add the most confident ones to the labeled data for the next training round. Such a process proceeds for a few iterations. In addition, at each iteration, the quality of the labeled data is monitored. The co-training process stops if the learning quality on the labeled data is below a predefined threshold. Through the above described co-training process, documents similar to the positive examples, which are considered highly related to the query topic, are added to the feedback set. In contrast, documents that are only partially relevant are discarded for relevance feedback. Finally, the resulting labeled positive documents are used for relevance feedback.

We propose an adaptive co-training method to select feedback documents for QE, which is the main contribution of this paper. Since the quality of feedback documents is crucial to QE's performance, the proposed AdapCOT method markedly improves the effectiveness and robustness of QE. To the best of our knowledge, this is the first study that successfully applies co-training, an effective learning paradigm, for selecting feedback documents in ad hoc retrieval. Meanwhile, taking the quality of learned classifiers into account makes the QE process more adaptive, and therefore our proposed method is robust on different kinds of data. Extensive experiments on five standard TREC collections show that our proposed approach outperforms strong baselines.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 explains our proposed AdapCOT method in details, and Section 4 introduces the classifiers used in our study. The proposed method is then evaluated through extensive experiments in Sections 5 and 6. Finally, Section 7 concludes the paper and suggests future research directions.

2. Related work

Co-training using unlabeled data has shown it effectiveness for reducing error rates in text classification (Blum and Mitchell, 1998; Goldman and Zhou, 2000; Joachims, 1999; Nigam et al., 2000; Zelikovitz and Hirsh, 2001). The idea of co-training is originally proposed by Blum and Mitchell for boosting the learning performance when there is only a small amount of labeled examples available (Blum and Mitchell, 1998). Under the assumption that there are two redundant but not completely correlated views of an example, unlabeled data are shown to be able to augment labeled data (Blum and Mitchell, 1998). Co-training is also used for extracting knowledge from the World Wide Web (Craven et al., 2000), or for email classification (Kiritchenko et al., 2004). The result shows that it can reduce the classification error by a factor of two using only unlabeled data. However, the performance of co-training depends on the learning methods used (Craven et al., 2000; Kiritchenko et al., 2004).

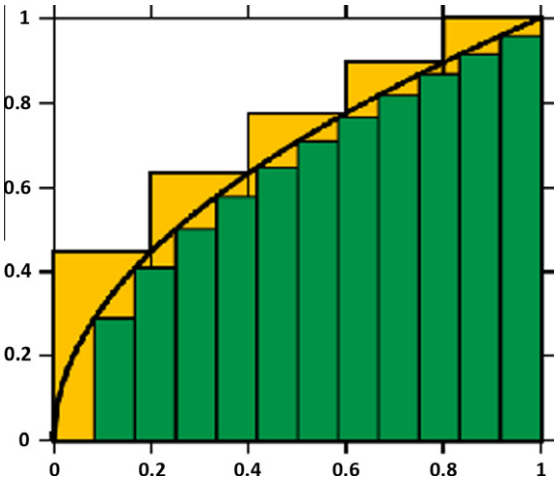

Fig. 1. AUC calculation.

There are a number of other studies that explore the potentials of co-training in recent years. Modified versions of the co-training method have been proposed. Goldman and Zhou (2000) use two different supervised learning algorithms to label data for each other. Raskutti presents a new co-training strategy which uses two feature sets. One classifier is trained using data from original feature space, while the other is trained with new features that are derived by clustering both the labeled and unlabeled data (Raskutti et al., 2002).

All the above work shows that the co-training method can be used to boost the performance of a learning algorithm when there is only a relatively small set of labeled examples given. However, this idea is based on the assumption that the dataset comes with two sets of features which are distinct in nature, but this is not the case in document retrieval. Chan et al. (2004) suggest to randomly split one single feature set into two sets for co-training. Their experimental results show that a random split of the feature set leads to comparable, and sometimes, even better results than using the natural feature sets. Huang et al. (2006) investigate the effect of Chan et al.'s version of co-training on the TREC HARD track data for passage retrieval by employing a dual-index model for term weighting. The main work of Huang et al. (2006) applies co-training to identify more relevant passages based on a small set of training data and use the results to re-estimate parameters of the probabilistic weighting function, BM25.

Recently, there have been efforts in applying machine learning methods to find good feedback documents or expansion terms. Lee et al. apply a clustering-based resampling method to select good feedback documents based on the relevance model. Relevance density is utilized to evaluate the quality of feedback documents. Cao et al. use features such as the proximity of expansion terms to the query terms, the expansion terms and query terms co-occurrences to predict which expansion terms are useful (Cao et al., 2008). Based on similar features, He and Ounis select good feedback documents using standard machine learning algorithms (He and Ounis, 2009a). All methods proposed in Lee et al. (2008), Cao et al. (2008), and He and Ounis (2009a) provide a moderate success over a traditional QE baseline.

In this paper, we propose to incorporate an adaptive co-training method for selecting feedback documents which is a substantial extension of Huang et al. (2006). In Huang et al. (2006), an initial study of applying co-training for QE was proposed over a small TREC HARD track dataset with 23 queries for passage retrieval. The basic idea of this method is to initialize a co-training process by taking the top-ranked and bottom-ranked passages in initial retrieval as positive and negative examples, respectively. The features representing the passages are randomly split into two sets, on which two different classifiers are learned respectively to label the remaining documents. Despite the limited but encouraging improvement on the TREC HARD track dataset, the approach (Huang et al., 2006) for selecting feedback documents using co-training suffers from the following issues. First, the random feature division may lead to the situation when the discriminative power of the features is completely unbalanced in the two sets. In this case, one of the classifiers may not be properly learned due to the low-quality features it has. Second, the previous approach is unable to cope with queries for which the top-ranked documents in the initial retrieval results are rather poor for training classifiers. The co-training process in Huang et al. (2006) will stop only when a particular number of iterations have been done. Thus, new feedback documents will be added even if the trained classifiers are of poor quality. As a result, the new feedback documents are not reliable and could bring negative effects into the QE process. In this case, we should stop the co-training process to avoid unreliable feedback documents. Hence, The AUC measure (Witten and Frank, 2005) is introduced for monitoring the quality of the learned classifiers.[1] AUC is calculated by using

[1] The Area Under a ROC Curve (AUC), calculated by the Mann–Whitney statistics, is a standard measurement for the soundness of a classification (Witten and Frank, 2005). The value of AUC ranges between 0 and 1. A high AUC value indicates a success in the classification and a low value indicates the contrary.

approximations of integrals. It adds up the area of all trapezoids (green[2] rectangles and yellow triangles under the curve) as in Fig. 1.[3] Although AUC is not a perfect measurement according to some studies (Lobo et al., 2008; Hand, 2009), we still use it in this paper to keep consistent with our previous work (Huang et al., 2006).

The iterative co-training process ends when AUC is lower than a given threshold. This adaptive criterion is different from that in the previous work (Huang et al., 2006; Xu and Akella, 2008), whose halting condition of iterations is only determined by a predefined value. Details about how to set this threshold will be discussed in Section 6. Compared to this preliminary work, this paper includes additional learning models, extensive experiments on more datasets, more systematic result analyses, more comprehensive discussion and more conclusive findings.

3. Adaptive co-training for QE

We propose an adaptive co-training method for selecting the feedback documents in this section. Section 3.1 introduces the document representation for the classification and Section 3.2 describes the proposed method in details.

3.1. Document representation

The standard co-training method assumes that an instance comes with two complementary sets of features in nature. For example, the features that describe a Web page can be the words on the page and the links that point to that page (Blum and Mitchell, 1998). However, for text retrieval, it is not obvious how to derive two complementary sets of features to represent the feedback documents, while the link information is usually not available.

There are quite a few options that we can apply to represent the feedback documents. In summary, these options can be grouped into two main categories, namely the term-based features, and the high-level features.

The term-based document representation characterizes a feedback document by its composing terms. A typical example is to represent a feedback document by a vector of the expansion term weights in the candidate feedback documents, where the weight of an expansion term is estimated by its normalized document generation probability (Rocchio, 1971; Salton, 1971). In contrast to the term-based document representation, other related studies use relatively high-level features, such as the proximity of the expansion term and the original query terms, the co-occurrences of the expansion term and the original query terms in the collection (Cao et al., 2008; He and Ounis, 2009a). The high-level features are usually considered having a higher descriptive power of the feedback documents than the term-based features, as they are found to be important factors affecting QE's effectiveness (Cao et al., 2008; He and Ounis, 2009a). However, there is also difficulty to apply the high-level features for co-training in practice: it is expensive to compute the features such as the proximity of the expansion term and the original query terms, for each retrieved document. In particular, since such features are query-dependent, which means their values change from query to query, they have to be computed during retrieval time. This is infeasible for large-scale collections. Therefore, in this paper, we rather follow the term-based document representation.

For D, the set of retrieved documents for a given query, we rank unique terms in D by a descending order of their Kullback–Leibler Divergence (KLD) weights. KLD is a popular choice of expansion term weighting, which has been shown to be effective in many state-of-the-art QE methods (Amati, 2003; Ye et al., 2009). For example, a KLD-based QE mechanism provides the best statMAP over the standard feedback sets in the TREC 2009 Relevance Feedback track (Ye et al., 2009). KLD measures how a term's distribution in the feedback documents diverges from its distribution in the whole collection. The higher KLD is, the more informative the term is. For a unique term in D, the KLD weight is given by:

$$\mathrm{KLD}(t) = P(t|D)\log_2\frac{P(t|D)}{P(t|C)} \tag{1}$$

where $P(t|D) = \frac{c(t,D)}{c(D)}$ is the generation probability of term t from D. $c(t, D)$ is the frequency of t in D, and $c(D)$ is the count of words in D. $P(t|C) = \frac{c(t,C)}{c(C)}$ is the collection model. $c(t, C)$ is the frequency of t in collection C, and $c(C)$ is the count of words in the whole collection C.

The *maxN* terms with the highest KLD weights in D are taken as the features to represent the documents. In each document d in D, the weight of a feature term t is again given by the Kullback–Leibler divergence of the term's distribution in d from its distribution in the whole collection. *maxN* is a parameter, which is obtained by tuning over a set of training topics. Once the feedback documents are represented by the feature terms which belong to the set F in Fig. 2, they are labeled by the AdapCOT method that is described in the next section.

3.2. Adaptive co-training

In this section, we devise an adaptive mechanism to apply co-training over a small set of training data which contains a few positive and negative documents in this case. Positive documents are those of good quality, and the negative ones are on the opposite side. In the proposed iterative co-training process, the halting criterion automatically adapts to the quality of

[2] For interpretation of color in Fig. 1, the reader is referred to the web version of this article.
[3] http://en.wikipedia.org/wiki/File:Integral_approximations.svg.

Input

L - a set of co.init.p top-ranked and co.init.n bottom-ranked feedback documents
U - a set of unlabeled feedback documents
C1, C2 - two classifiers
F - a set of feature terms
Control parameters: co.init.p, co.init.n, co.p, co.n, co.K, co.AUC

Output

L - an expanded set of labeled feedback documents

Method

1. Split F into two feature sets F1 and F2
2. Do the following for co.K iterations:
   (1) Learn classifier C1 from L using F1 to represent the documents in L
   (2) If the resulting AUC value is lower than co.AUC, break;
   (3) Use C1 to classify examples in U based on F1
   (4) Select co.p/co.n most confidently predicted positive/negative instances from U
   (5) Add these co.p + co.n instances with their corresponding labels into L
   (6) Remove these co.p + co.n examples from U
   (7) Learn classifier C2 from L using F2 to represent the documents in L
   (8) If the resulting AUC value is lower than co.AUC, break
   (9) Use C2 to classify examples in U based on F2
   (10) Repeat steps (4) - (6)

Fig. 2. The adaptive co-training algorithm (AdapCOT).

the learned classifiers. It monitors the quality of the learned classifiers at the end of each iteration using the AUC measure. The co-training process stops when the quality of the learned classifiers is below a given threshold, which will be discussed later in this paper. The basic idea of our proposed method is to represent the documents by important terms in the retrieved set, which are highly weighted. A small number of top-ranked and bottom-ranked documents from the initial retrieval are used as positive and negative examples, respectively. The top-ranked documents are assumed to be of a high quality, and provide a good coverage on the query topic in their content. In contrast, the bottom-ranked documents are used as negative examples in which the content is assumed to be off-topic. A learning procedure is designed to label the retrieved documents so that the documents selected for QE are meant to be highly similar to the positive set, while being dissimilar to the negative set.

The proposed method learns two different classifiers from two non-overlapped sets of features which represent the documents. More specifically, it allows the two classifiers, C1 and C2, to label the unlabeled instances, i.e. the rest of the retrieved documents, for each other through an iterative process until one of the halting criteria is met. The positively labeled documents are then used for feedback in QE.

Fig. 2 gives a general description of the proposed AdapCOT method. The proposed method is based on the initial retrieval results returned by the search engine. The *co.p* top-ranked and the *co.n* bottom-ranked documents are used as the labeled positive and negative examples, respectively. The rest of the retrieved documents remain unlabeled. As explained in the previous section, *maxN* unique terms with the highest KLD weights are used as features to represent the retrieved documents. To facilitate the co-training process, these *maxN* terms are split into two different sets. Each term acts as one feature of the documents in our AdapCOT method. Instead of a random split suggested in Chan et al. (2004), we rank these *maxN* terms in decreasing order of their KLD weights, and group them into an odd-ranked set and an even-ranked set, respectively. One of the advantages of our split method over the random split is that our method balances the quality of terms in the two feature sets,[4] and avoids the case that one of the classifiers is not properly learned due to an extremely unbalanced split.

Once the feature terms are split into two sets, the two classifiers are learned on the labeled documents one by one, and label the remaining unlabeled documents for each other. Such a co-training process ends when one of the halting criteria is met. In our proposed method, we introduce two halting criteria as follows. First, the co-training process ends after co.K iterations. Second, the co-training process ends if AUC, a measure used for evaluating the successfulness of the classification over the labeled examples, is below a given threshold co.AUC. The second criterion is a key step in the proposed method. It ensures that the learned classifiers are good enough to classify the unlabeled documents. It is particularly useful when the initial retrieval is of a poor quality, and the "bad" feedback documents would not be labeled as

[4] In query expansion, terms with high weights are usually considered as being of high quality, and are therefore added to the query.

positive since the co-training process does not proceed further when the classifiers are not properly learned. Note that the AUC value is computed using the documents labeled by the learned classifiers, and the actual relevance assessment information is not used. Weka,[5] a powerful data mining tool, is used to calculate the value of AUC for the implementation of our AdapCOT.

Furthermore, in order to guarantee the quality of the labeled positive documents for relevance feedback, we introduce the following restrictions (He and Ounis, 2009b):

- Only the 50 top-ranked documents in the initial retrieval can be added to the labeled positive document set. This is based on the empirical observation that QE is unlikely to benefit from a feedback document that is roughly ranked lower than 50.
- When two labeled positive documents receive an identical confidence value from the classifier, the one ranked higher is preferred. This is also based on the empirical observation that the higher ranked documents are likely to be better feedback documents than the lower ranked ones.

In addition, our proposed AdapCOT method automatically labels the most confidently predicted positive documents for relevance feedback. It is inexpensive to implement since the co-training process does not involve any relevance assessment information by human effort.

4. Classifiers

In our studies, we choose to apply three different learning methods to facilitate the proposed adaptive co-training method. These methods are among the most popular machine learning methods applied in IR, including the low-cost Naive Bayes and Logistic Regression, and the relatively sophisticated support vector machines. Since our work in this paper is a significant extension of Huang et al. (2006), we only test the combinations of two different classifiers as in Huang et al. (2006). Thus, we have three combinations as shown in Table 2.

Naive Bayes (NB) is a simple statistical learning algorithm based on Bayes' theorem with strong independence assumptions. Despite their naive design and simple independence assumptions, NB classifiers have worked quite well in many complex real-world situations (Witten and Frank, 2005). In this paper, we apply the popular Gaussian kernel density function as follows:

$$P(x|C_i) = \frac{1}{\sigma_i\sqrt{\pi}} \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_i}\right) \tag{2}$$

where the probability $P(x|C_i)$ of observing an instance x in class C_i is approximated by a Gaussian density function with a standard deviation σ_i and mean μ_i.

Logistic Regression (LR) is a generalized linear model for binomial regression (Witten and Frank, 2005). The basic idea of LR is to apply a *logit* function to scale membership values, which may not be proper probabilities, to values between 0 and 1:

$$f(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}} \tag{3}$$

where a membership value z is mapped to a value between 0 and 1.

Support vector machines (SVMs) (Joachims, 1999) are widely used in text classification in recent years. Its underlying principle is structure risk minimization. Its objective is to determine a classifier or regression function which minimizes the empirical risk (i.e., the training set error) and the confidence interval (which corresponds to the generalization or test set error). Given a set of training data, an SVM determines a hyperplane in the space of possible inputs. This hyperplane will attempt to separate the positives from the negatives by maximizing the distance between the nearest positive examples and and negative examples. There are several ways to train SVMs. One particularly simple and fast method is the sequential minimal optimization (Platt, 1999) which is adopted in our study. In addition, we apply the non-linear homogeneous polynomial kernel function at degree m as follows:

$$k(x_i, x_j) = (x_i \cdot x_j)^m \tag{4}$$

where x_i and x_j are real vectors in a p-dimensional space, and p is the number of features. The exponential parameter m is set to 1 as a default value in our study. In the following sections, the proposed co-training method is evaluated through extensive experiments.

5. Experimental setup

We evaluate our proposed AdapCOT method on most of the existing TREC datasets with ad hoc topics, including the disk1&2, disk4&5, WT10G, .GOV2, and ClueWeb B collections. Basic statistics about the test collections and topics are given in Table 1.

[5] http://www.cs.waikato.ac.nz/ml/weka.

Table 1
Information about the test collections.

Coll.	TREC task	Topics	# Docs
disk1&2	1–3, Ad-hoc	51–200	741,856
disk4&5	2004, Robust	301–450, 601–700	528,155
WT10G	9, 10 Web	451–550	1,692,096
.GOV2	2004–2006 Terabyte Ad-hoc	701–850	25,178,548
ClueWeb09	2009 Relevance Feedback	rf.01–rf.50	49,375,681

The disk1&2 and disk4&5 (no Congressional Record according to the robust track of TREC 2004 and 2005) collections contain newswire articles from various sources, such as Associated Press (AP), Wall Street Journal (WSJ), and Financial Times (FT), which are usually considered as high-quality text data with little noise. The WT10G collection is a medium size crawl of Web documents, which was used in the TREC 9 and 10 Web tracks. It contains 10 GB of uncompressed data. The .GOV2 collection, which is 426 GB in size and contains 25 million documents, is a crawl from the .gov domain. This collection has been employed in the TREC 2004, 2005 and 2006 Terabyte tracks. The ClueWeb09 collection is a very large crawl of the Web, and is currently the largest TREC test collection. We use the category B of ClueWeb09,[6] which contains about 50 million English Web pages, and its associated topics used in the TREC 2009 Relevance Feedback track. We index all documents in the above five collections. For all five test collections used, each term is stemmed using Porter's English stemmer, and stopwords are removed.

Each topic contains three topic fields, namely title, description and narrative. We only use the title topic field that contains very few keywords related to the topic. The title-only queries are usually as short as a realistic snapshot of real user queries in practice (Xu and Akella, 2008; Zhai and Lafferty, 2001). On each collection, we evaluate our proposed model by a 10-fold cross-validation. The test topics associated to each collection are randomly split into ten equal subsets. In each fold, we use nine subsets of test topics for training, and use the remaining subset for testing. The overall retrieval performance is averaged over all 10 test subsets of topics. We use the TREC official evaluation measures in our experiments, namely the stat-MAP at 1000 on ClueWeb09 (Aslam et al., 2006), and the Mean Average Precision (MAP) at 1000 on the other four collections (Voorhees, 2005). For each query, the top *co.init.p* and the bottom *co.init.n* ranked documents are used as the initial labeled examples for the co-training. Only the top 1000 retrieved documents are involved in the co-training process.[7] All statistical tests are based on Wilcoxon matched-pairs signed-rank test at the 0.05 level.

In our experiments, we apply Okapi's BM25 for document ranking. BM25 is a classical probabilistic model based on approximation to the assumed two-Poisson term frequency distribution (Robertson et al., 1995). It assigns the relevance score for a document d with respect to a given query Q by (Robertson et al., 1995):

$$score(Q|d) = \sum_{t \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} qtw \tag{5}$$

where tf is the frequency of the given term t in the document. k_1 is a parameter. Its default setting is $k_1 = 1.2$ (Robertson et al., 1995). $w^{(1)}$ is the raw term weight, which is given by the re-written point-5 formula as follows:

$$w^{(1)} = \log \frac{N - N_t + 0.5}{N_t + 0.5} \tag{6}$$

where N is the number of documents in the collection, and N_t is the number of documents containing the term t in the collection. qtw is the query term weight, which is given by:

$$qtw = \frac{(k_3+1)qtf}{k_3+qtf} \tag{7}$$

where qtf is the number of occurrences of the given term in the query. k_3 is a parameter. Its default setting is $k_3 = 1000$ (Robertson et al., 1995). K is: $K = k_1\left((1-b) + b\frac{l}{avg\_l}\right)$, where l and avg_l are the document length and the average document length in the collection, respectively. The length refers to the number of tokens in a document. b is a hyper-parameter. In this paper, it is obtained by Simulated Annealing (Kirkpatrick et al., 1983) over the training topics on the collection used (Robertson et al., 2004).

We apply the KLD weighting for query expansion (Ye et al., 2009). The applied QE algorithm considers the top-ranked documents as the feedback set D_f. An expansion weight $w(t)$ is assigned to each unique term in D_f. $w(t)$ is the mean of KLD weights computed over each feedback document by Eq. (1). Particularly, D in Eq. (1) refers to the selected feedback documents in D_f.

The $|expT|$ terms with the highest mean KLD weights over D_f are added to the query, where $expT$ is the set of expansion terms. In our experiments, we set $|expT|$, the number of expansion terms to be added to the query, to 20, as it is shown to be effective in some previous studies (Büttcher and Clarke, 2006; Collins-Thompson, 2009; Keikha et al., 2011).

We mainly report on the results obtained by setting *co.init.p*, the initial number of top-ranked documents used as positive examples, to 2, 3, 4, 5, 8, 10, 15 and 20. Impact of varying *co.init.p* on AdapCOT's retrieval performance is also discussed later. In

[6] category B is a subset of ClueWeb09 A Category which is 25 TB in size. In this paper, ClueWeb09 refers to the B category.
[7] That is, a document ranked at 1000 is considered the bottom-ranked.

addition, *co.init.n*, the number of bottom-ranked documents used as negative examples, is set to the twice of *co.init.p*, i.e. *co.init.n* = *co.init.p* · 2. This is based on the findings in previous study (Huang et al., 2006) that it would better have more initial negative examples than the positive ones. Moreover, in the experiment conducted by Blum and Mitchell (1998) for classifying university Web pages, *co.p* and *co.n* are set to 1 and 3, respectively. That is, in each iteration, each classifier is allowed to add 1 new positive and three new negative examples to *L*. In this paper, in order to reduce the number of parameters for tuning, we follow the setting of *co.p* and *co.n* recommended in Blum and Mitchell (1998). Moreover, *co.K*, the number of iterations in each co-training process, is set to 3 according to the recommendation in Huang et al. (2006). Thus, at most three positive and nine negative examples are added to *L*. If *co.K* is 0, AdapCOT acts in the same way as traditional QE methods do. Other control parameters, namely the number of terms used for document representation (*maxN*), and the threshold *co.AUC*, are obtained over the training data in the cross-validation process. The related evaluation results are presented in the next section.

6. Experimental results

We evaluate our proposed AdapCOT for selecting feedback documents against different baselines, namely the initial retrieval using BM25, query expansion (QE) using top-ranked documents, and the baseline co-training method (baseCOT) without the adaptive quality control proposed for AdapCOT.

6.1. Comparison with initial retrieval

In the first step of our experiments, Table 2 compares AdapCOT's retrieval performance with the an initial retrieval baseline using BM25. In this step, the setting of *co.init.p* is default to 3. As we can see from this table, AdapCOT markedly outperforms the BM25 baseline in all cases. This is not of a great surprise since AdapCOT applies QE, which usually improves the initial retrieval performance. Moreover, although the use of different classifiers leads to very similar retrieval performance, in the rest of the paper, only the results obtained by using LR and SVM for co-training are presented because this combination gets four best results over all the five collections.

In addition, we examine the accuracy of the co-training method in selecting feedback documents. Regarding the "goodness" of a feedback document, we consider the following three different categories: (1) by using the document alone for relevance feedback, a "good" feedback document leads to at least 5% improvement over AP, the average precision of the query in initial retrieval; (2) a "bad" feedback document leads to a decrease in AP by at least 5%; and (3) other feedback documents are considered "neutral".

Table 3 presents the percentage of the feedback documents selected by AdapCOT in the above three categories. Overall, a moderate accuracy of the AdapCOT method is observed. A major proportion of the feedback documents picked up by AdapCOT, about 2 out of 3, fall into either "good" or "neutral" categories. The remaining "bad" feedback documents may hurt the retrieval performance. However, the impact of "bad" feedback documents could be neutralized since they are to some degree outnumbered by the "good" ones. In the rest of this section, we show that the moderate accuracy of AdapCOT indeed leads to improved retrieval performance over strong baselines.

6.2. Comparison with QE

Next, we compare our proposed AdapCOT method with the QE baseline. QE is based on the assumption that the top-ranked documents are relevant, from which the most important terms are useful for retrieving more relevant documents.

Table 2
MAP/statMAP obtained by the BM25 baseline and AdapCOT.

Coll.	BM25	NB + LR	NB + SVM	LR + SVM
disk1&2	0.2307	0.2795[a], 21.15%	0.2797[a], 21.24%	0.2797[a], 21.24%
disk4&5	0.2499	0.2910[a], 16.45%	0.2916[a], 16.69%	0.2916[a], 16.69%
WT10G	0.2090	0.2338[a], 11.87%	0.2333[a], 11.63%	0.2338[a], 11.87%
.GOV2	0.3041	0.3245[a], 6.71%	0.3243[a], 6.64%	0.3235[a], 6.38%
ClueWeb09	0.2052	0.2288[a], 11.50%	0.2285[a], 11.35%	0.2330[a], 13.55%

[a] Indicates a statistically significant improvement over the baseline.

Table 3
The proportion of good/bad feedback documents picked up by the AdapCOT method.

Coll.	Good (%)	Neutral (%)	Bad (%)
disk1&2	59.02	2.05	40.78
disk4&5	56.16	5.48	38.36
WT10G	57.14	14.28	28.57
.GOV2	54.17	16.67	29.17
ClueWeb09	61.78	19.11	19.11
Total	60.27	6.54	33.33

The size of the feedback document set has considerable impact on QE's effectiveness. In contrast, the AdapCOT method follows a slightly different assumption. It assumes the positivity of a small set of top-ranked documents, and the negativity of another small set of bottom-ranked documents. It selectively adds other retrieved documents to the positive document set through an iterative process until a halting criterion is met. Therefore, we compare the retrieval performance of AdapCOT with QE using the same number of top-ranked documents ($|D_f|$) for relevance feedback, and we vary $|D_f|$ from 2 to 20 to see how it influences the results of our AdapCOT and normal QE. The setting of *co.init.p*, the number of initial positive examples for co-training, is set to $|D_f|$. Meanwhile, we use the same number of expansion terms for both AdapCOT and QE, which is set to 20 as mentioned in Section 5. The related results are given in Table 4. In addition, Fig. 3 plots the retrieval performance of the QE baseline and AdapCOT against different settings of $|D_f|$ = *co.init.p*. An encouraging conclusion drawn from Table 4 and Fig. 3 is that AdapCOT in general outperforms the QE baseline with different $|D_f|$ settings. AdapCOT performs particularly well on the large-scale .GOV2 and ClueWeb09 collections, where it achieves statistically significant improvement over the QE baseline even if $|D_f|$ is optimal for QE. AdapCOT also appears to be more robust than the QE baseline. In most cases, Adap-

Table 4
MAP/statMAP obtained by the QE baseline and AdapCOT.

	disk1&2		disk4&5		WT10G	
$\|D_f\|$	QE	AdapCOT	QE	AdapCOT	QE	AdapCOT
2	0.2528	0.2684[a], 6.17%	0.2639	0.2808[a], 6.40%	0.2206	0.2332[a], 5.95%
3	0.2732	0.2797, 2.57%	0.2857	0.2916, 2.06%	0.2320	0.2338, 0.776%
4	0.2757	0.2742, −0.182%	0.2812	0.2877, 2.31%	0.2209	0.2247, 1.72%
5	0.2775	0.2725, −1.80%	0.2775	0.2877, 3.68%	0.2109	0.2216[a], 5.07%
8	0.2703	0.2727, 0.888%	0.2723	0.2833, 4.04%	0.1824	0.2007[a], 10.03%
10	0.2699	0.2759, 2.22%	0.2636	0.2775[a], 5.27%	0.1782	0.1974[a], 11.11%
15	0.2603	0.2652, 1.88%	0.2476	0.2689[a], 8.60%	0.1612	0.1852[a], 14.89%
20	0.2561	0.2675, 4.45%	0.2301	0.2571[a], 11.73%	0.1412	0.1768[a], 25.21%
	.GOV2		ClueWeb09		Average	
2	0.2689	0.3038[a], 12.98%	0.1787	0.2303[a], 28.88%	0.2370	0.2633, 11.10%
3	0.3003	0.3235[a], 7.73%	0.2047	0.2330[a], 13.82%	0.2591	0.2679, 3.28%
4	0.2958	0.3218[a], 8.79%	0.2083	0.2116, 1.58%	0.2564	0.2630, 2.57%
5	0.2847	0.3165[a], 11.17%	0.1898	0.2000[a], 5.37%	0.2481	0.2597, 4.68%
8	0.2644	0.3042[a], 15.05%	0.1863	0.2013[a], 8.05%	0.2351	0.2524, 7.36%
10	0.2538	0.2946[a], 16.07%	0.1747	0.1948[a], 6.58%	0.2280	0.2480, 8.77%
15	0.2375	0.2850[a], 20.00%	0.1777	0.1728, −2.76%	0.2169	0.2354, 8.53%
20	0.2325	0.2798[a], 20.34%	0.1680	0.1542, −8.21%	0.2056	0.2271, 10.46%

[a] Indicates a statistically significant improvement.

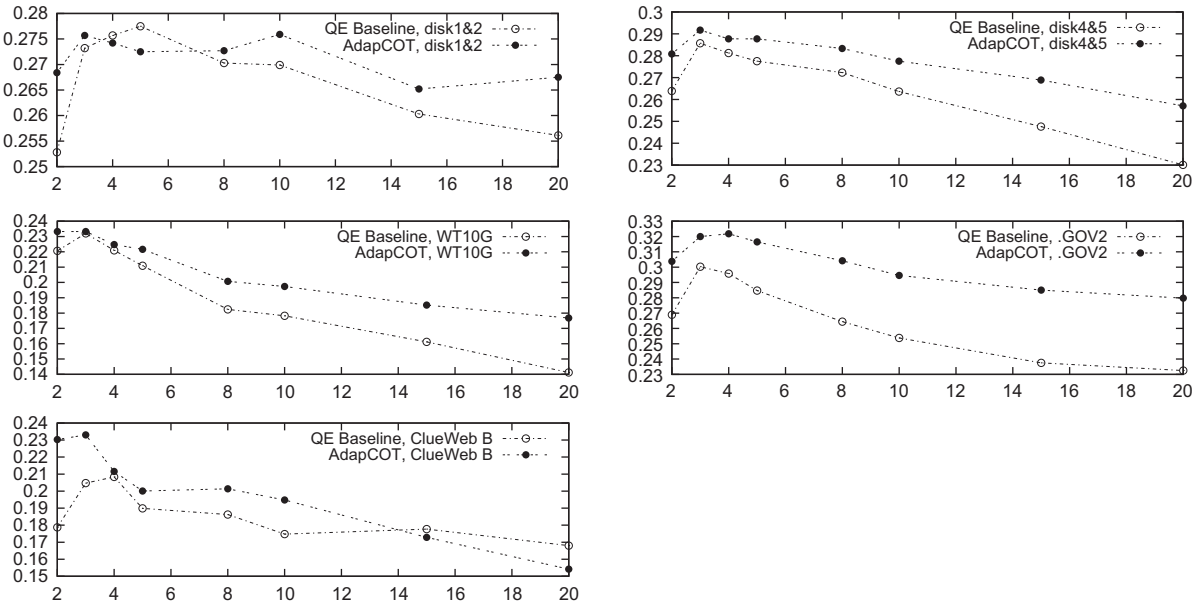

Fig. 3. The MAP/statMAP obtained by AdapCOT and by the QE baseline (*Y*-axis) with different settings of $|D_f|$ = *co.init.p* (*X*-axis).

Table 5
The percentages of "good" feedback documents selected by the QE baseline and AdapCOT, respectively.

	disk1&2		disk4&5		WT10G	
$\|D_f\|$	QE	AdapCOT	QE	AdapCOT	QE	AdapCOT
2	62.68	55.67	47.60	37.74	46.50	41.67
3	62.00	59.02	45.38	56.16	40.33	47.14
4	60.34	52.90	43.68	37.74	37.50	43.33
5	59.60	49.90	43.13	40.54	35.60	50.00
8	57.33	53.56	40.44	36.58	32.88	48.65
10	57.00	53.12	38.59	32.14	31.60	33.15
15	54.94	50.65	35.80	23.81	28.13	32.44
20	53.33	47.53	32.89	28.57	27.05	29.63
	.GOV2		ClueWeb09		Average	
2	20.92	46.43	33.68	55.67	42.28	47.44
3	30.15	54.17	31.97	61.78	41.97	60.27
4	39.38	51.43	35.20	65.54	43.22	55.65
5	49.23	53.33	33.88	63.97	44.29	51.55
8	37.46	41.38	35.98	62.61	40.82	48.56
10	44.15	50.00	36.73	58.50	41.61	45.38
15	42.10	58.62	35.92	55.57	39.38	44.22
20	40.31	68.42	35.61	57.90	37.84	46.41

COT improves the QE baseline with different $|D_f|$ settings, even if the QE baseline has very poor retrieval performance. Besides, in Tables 2 and 4, optimized BM25 outperforms the QE baseline and the AdapCOT in some cases, but generally QE and AdapCOT obtain better results when $|D_f|$ is optimal. Moreover, the best result of the QE baseline on .GOV2 (MAP 0.3003) is not so good as that of BM25 (MAP 0.3041), whereas AdapCOT (MAP 0.3243) still surpasses BM25 significantly when $|D_f|$ is set to 3. This indicates the robustness of our AdapCOT method.

Furthermore, we examine whether using AdapCOT would give a higher percentage of good feedback documents than just taking the top-ranked documents in Table 5. Although this is indeed the case on some of the collections used, it is surprising that the retrieval performance of both the QE baseline and AdapCOT is not necessarily correlated with the percentage of good feedback documents when $|D_f|$ becomes large. By further analysis on the experiments, we discover that the effectiveness of QE depends heavily on some key documents, which have the most important contribution to the expanded query. Therefore, if these key documents are included in the feedback set, the retrieval performance is usually good as long as the good feedback documents are not outnumbered by the bad ones. In this case, AdapCOT is particularly useful when the key documents are not highly ranked, which would not be picked up the QE baseline. A typical example is query 195 on disk1&2, for which the initial average precision (AP) obtained by BM25 is 0.0273. For this query, the document with id AP891017-0231 has the most contribution to the expanded query. However, as this key document is only ranked 35th in the initial retrieval, it is not used for feedback by the QE baseline. In contrast, AdapCOT manages to automatically identify this key document and add it to the feedback document set. It is then of no surprise that the QE baseline is unable to improve over the initial retrieval, while AdapCOT provides an AP of 0.1638 with a 500% improvement.

6.3. Impact of co.maxN and co.AUC

We also study the impact of the other two control parameters, namely the threshold (*co.AUC*) and the number of terms used for the document representation (*co.maxN*). Figs. 4 and 5 plot the retrieval performance of AdapCOT against these two control parameters, respectively. It seems that the sensitivity of AdapCOT's performance to the parameters depends on the collection used. Overall, AdapCOT's parameter sensitivity is relatively high on the three Web collections, namely WT10G, .GOV2 and ClueWeb09, while being low on the other two collections. On average, a *co.maxN* value around 100 is shown to be safe over all five collections used. On the other hand, the setting of *co.AUC* depends on the collection used. For the three collections with relatively small sizes, namely disk1&2, disk4&5, and WT10G, the retrieval performance does not seem to be sensitive to the setting of *co.AUC*. The MAP values obtained remain stable across different *co.AUC* settings. For the other two very large Web collections, namely .GOV2 and ClueWeb09, we observe a relatively high variance of the retrieval performance over different *co.AUC* settings, while *co.AUC* = 0.30 leads to the best retrieval performance in average over these two very large Web collections.

6.4. Comparison with baseCOT

A major advantage of our proposed approach is to employ a thresholding strategy that carries out a quality control on the learned classifiers. Therefore, we also compare our proposed AdapCOT with the original co-training method without the introduction of the threshold, which is equivalent to the algorithm described in Fig. 2 without steps 2.(2) and 2.(8). By apply-

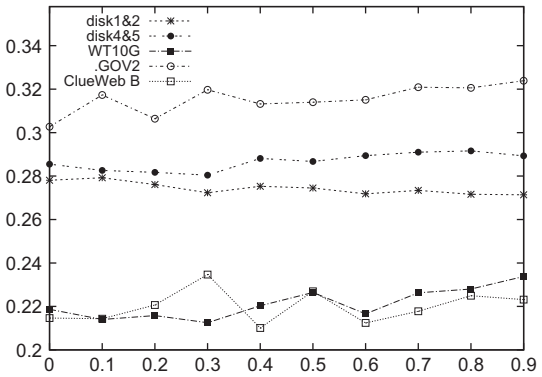

Fig. 4. The MAP/statMAP obtained by AdapCOT (*Y*-axis) against the control parameter *co.AUC*.

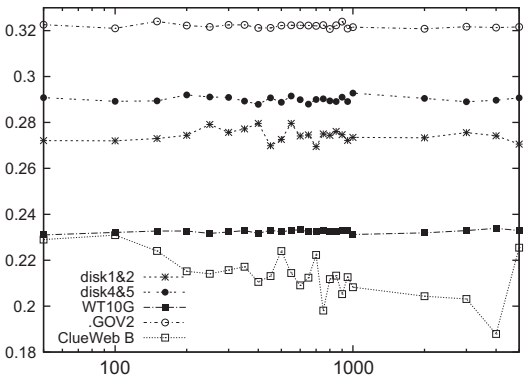

Fig. 5. The MAP/statMAP obtained by AdapCOT (*Y*-axis) against the control parameter *co.maxN*.

Table 6
MAP/statMAP obtained by the baseline co-training method (baseCOT) and the adaptive co-training (AdapCOT).

Coll.	baseCOT	AdapCOT
disk1&2	0.2769	0.2798, 1.05%
disk4&5	0.2855	0.2916, 2.14%
WT10G	0.2187	0.2338[a], 6.90%
.GOV2	0.3127	0.3235[a], 3.45%
ClueWeb09	0.2158	0.2330[a], 7.97%

ing this baseline co-training method (baseCOT), the co-training process proceeds for *co.K* iterations, regardless of the quality of the learned classifiers. As shown in Table 6, our proposed AdapCOT method indeed improves retrieval performance over the original co-training baseline, particularly on the three Web collections. This is possibly due to the heterogeneous nature of these three Web collections, in which documents tend to be noisy, and the quality of training data becomes crucial for designing a co-training process to find good feedback documents.

Additionally, our proposed AdapCOT method does not cost more time than the QE significantly in our experiments. The AdapCOT method has extra computational cost because of the co-training process. However, this cost depends on the amount of features (*maxN*) and number of the iterations instead of the size of collections. Fig. 5 indicates that the best results are obtained when *maxN* is relatively small. The number of iteration in our experiments is set to 3. Since one iteration only consumes 8–10 s on our workstation(P4 2.6G, 4G RAM, 1.5T HDD), the extra time cost is not extensive when compared to the QE baseline.

6.5. Comparison with term selection method

In this section, we compare our proposed AdapCOT with the state-of-the-art term selection method (TS) (Cao et al., 2008). The basic idea of TS is to select expansion terms using SVM based on a list of features, including the distance of the expansion term to original query terms, the expansion term's distribution in the feedback documents, etc. In the literature, TS is the first

Table 7
MAP/statMAP obtained by the term selection method (TS) and AdapCOT.

Coll.	Topics	TS	AdapCOT
AP	51–100	0.3090	0.3251, 5.21%
WSJ	51–100	0.3036	0.3111, 2.47%
disk4&5	351–400	0.2208	0.2319, 5.03%

method that applies SVM to select expansion terms. It is a strong baseline, which has shown marked (up to 28.36%) improvement over the KL-divergence language model (Cao et al., 2008).

In order to establish a fair comparison, we apply AdapCOT on the same collections with the same test queries used in Cao et al. (2008). Table 7 provides the related experimental results. Our AdapCOT method is shown to be generally more effective than TS. AdapCOT provides a moderate, up to 5.21% improvement over TS on the three test collections used. This indicates that AdapCOT's retrieval performance is at least comparable to, if not better than, the state-of-the-art TS method for enhancing QE effectiveness.

6.6. Summary

In summary, the proposed AdapCOT method outperforms both the initial retrieval and the QE baselines in our experiments. The settings of the control parameters *co.init.p* and *co.AUC* have a considerable impact on AdapCOT's effectiveness, particularly on the large-scale Web collections. Moreover, it is indeed helpful to introduce a parameter *co.AUC* to monitor the soundness of the learned classifiers. The AdapCOT method improves co-training as shown in our experiments. Finally, we compare our proposed AdapCOT method to the state-of-the-art TS method. These two methods follow different approaches to providing fine-grained feedback information for QE. The TS method selects expansion terms from a large number of feedback documents. Despite its effectiveness, there is still a potential in improving QE due to the fact that a majority of the feedback documents are not closely related to the query topic, and hence can be discarded. In contrast, AdapCOT refines the feedback set by removing the noisy documents, based on the idea that the partially relevant documents can actually hurt QE. Our evaluation results have demonstrated considerable and statistically significant improvement brought by AdapCOT, showing that it is indeed crucial to distillate the good feedback documents before QE takes place.

7. Conclusions and future work

An important conclusion of this paper is the necessity to distillate good feedback documents for QE. A good feedback document should have a general interest in the query topic throughout itself, while a bad feedback document, even if it is relevant, may have only a passing-by interest in the query topic, and contains many off-topic terms. Therefore, there is a need for distillating the high quality feedback documents to improve QE's effectiveness and robustness. To address this problem, we have proposed an adaptive co-training method, called AdapCOT, to find good feedback documents for QE. The proposed method overcomes a major problem of applying machine learning methods to QE, namely the lack of proper training data. On five TREC collections, extensive experiments confirm our argument that QE's retrieval performance can be improved by selecting high quality feedback documents. According to the experimental results over five standard TREC test collections, our proposed AdapCOT leads to considerable and statistically significant improvement over the QE baseline. In particular, AdapCOT is very effective on the large-scale Web collections, showing that our proposed method can be applied in Web environment. Moreover, current IR systems can benefit from our proposed method by a good choice of feedback documents, since most of them offer the functionality of QE in the form of feedback based on a pseudo relevance set.

In the future, we plan to investigate ways to further improve the AdapCOT method. For example, we can employ a co-training process that makes use of two nature sets of document features, namely the term-based features (Huang et al., 2006) and the high-level features (He and Ounis, 2009a; Cao et al., 2008). As mentioned in Section 3.1, a major obstacle of applying the high-level features for co-training is the associated computational cost. However, we may still examine if the use of the high-level features is workable in a laboratory setting.

Acknowledgements

This research is supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Early Researcher Award/Premiers Research Excellence Award and the IBM Shared University Research (SUR) Award. We thank anonymous reviewers for their valuable and detailed comments on this paper. We also would like to thank IBM Canada for providing IBM BladeCenter blade servers to conduct experiments reported in the paper.

References

Amati, G. (2003). *Probabilistic models for information retrieval based on divergence from randomness*. Ph.D. thesis, Department of Computing Science, University of Glasgow.

Amati, G., Carpineto, C., & Romano, G. (2004). Query difficulty, robustness, and selective application of query expansion. In S. McDonald & J. Tait (Eds.), *ECIR* (pp. 127–137). Springer.
Aslam, J. A., Pavlu, V., & Yilmaz, E. (2006). A statistical method for system evaluation using incomplete judgments. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 541–548). ACM.
Blum, A., & Mitchell, T. M. (1998). Combining labeled and unlabeled data with co-training. In *COLT* (pp. 92–100).
Büttcher, S., & Clarke, C. L. A. (2006). In *Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 182–189). ACM.
Buckley, C., Salton, G., Allan, J., & Singhal, A. (1994). Automatic query expansion using smart: Trec 3. In *TREC'94* (pp. 69–80).
Cao, G., Nie, J. Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In S. H. Myaeng, D. W. Oard, F. Sebastiani, T. S. Chua, & M. K. Leong (Eds.), *SIGIR* (pp. 243–250). ACM.
Carpineto, C., de Mori, R., Romano, G., & Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *CM Transactions on Information and System, 19*, 1–27.
Carpineto, C., Romano, G., & Giannini, V. (2002). Improving retrieval feedback with multiple term-ranking function combination. *CM Transactions on Information and System, 20*, 259–290.
Chan, J., Koprinska, I., & Poon, J. (2004). Co-training with a single natural feature set applied to email classification. In *Web intelligence* (pp. 586–589). IEEE Computer Society.
Collins-Thompson, K. (2009). Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 837–846). ACM.
Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T. M., Nigam, K., et al (2000). Learning to construct knowledge bases from the world wide web. *Artificial Intelligence, 118*, 69–113.
Efthimiadis, E. N. (1996). Query expansion. *Annual Review of Information Systems and Technology, 31*.
Goldman, S. A., & Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. In P. Langley (Ed.), *ICML* (pp. 327–334). Morgan Kaufmann.
Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning, 17*, 103–123.
He, B., & Ounis, I. (2009a). Finding good feedback documents. In D. W. L. Cheung, I. Y. Song, W. W. Chu, X. Hu, & J. J. Lin (Eds.), *CIKM* (pp. 2011–2014). ACM.
He, B., & Ounis, I. (2009b). Studying query expansion effectiveness. In M. Boughanem, C. Berrut, J. Mothe, & C. Soulé-Dupuy (Eds.), *ECIR* (pp. 611–619). Springer.
Huang, X., Huang, Y. R., Wen, M., An, A., Liu, Y., & Poon, J. (2006). Applying data mining to pseudo-relevance feedback for high performance text retrieval. In *ICDM* (pp. 295–306). IEEE Computer Society.
Joachims, T. (1999). Transductive inference for text classification using support vector machines. In I. Bratko & S. Dzeroski (Eds.), *ICML* (pp. 200–209). Morgan Kaufmann.
Keikha, M., Seo, J., Croft, W. B., & Crestani, F. (2011). Predicting document effectiveness in pseudo relevance feedback. In *Proceedings of the 20th ACM international conference on information and knowledge management* (pp. 2061–2064). ACM.
Kiritchenko, S., Matwin, S., & Abu-Hakima, S. (2004). Email classification with temporal features. In M. A. Klopotek, S. T. Wierzchon, & K. Trojanowski (Eds.), *Intelligent information systems* (pp. 523–533). Springer.
Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science, 220*.
Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models. In *SIGIR* (pp. 120–127).
Lee, K. S., Croft, W. B., & Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In *SIGIR* (pp. 235–242).
Lobo, Jorge M., Jimnez-Valverde, Alberto, & Real, Raimundo (2008). AUC: A misleading measure of the performance of predictive distribution models. In *CIKM* (pp. 403–410). ACM.
Lynam, T. R., Buckley, C., Clarke, C. L. A., & Cormack, G. V. (2004). A multi-system analysis of document and term selection for blind feedback. In *CIKM* (pp. 261–269). ACM.
Miao, J., Huang, X., & Ye, Z. (2012). Proximity-based rocchio's model for pseudo relevance feedback. In *SIGIR* (pp. 10). ACM.
Mitra, M., Singhal, A., & Buckley, C. (1998). Improving automatic query expansion. In *SIGIR* (pp. 206–214). ACM.
Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*, 103–134.
Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: Support vector learning* (pp. 185–208). MIT Press.
Raskutti, B., Ferrá, H. L., & Kowalczyk, A. (2002). Combining clustering and co-training to enhance text classification using unlabelled data. In *KDD* (pp. 620–625). ACM.
Robertson, S. E. (1990). On term selection for query expansion. *Journal of Documentation, 46*, 359–364.
Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gatford, M., & Payne, A. (1995). Okapi at TREC-4. In *Proceedings of TREC-4*.
Robertson, S. E., Zaragoza, H., & Taylor, M. J. (2004). Simple BM25 extension to multiple weighted fields. In *CIKM* (pp. 42–49). ACM.
Rocchio, J. (1971). Relevance feedback in information retrieval. In *The SMART retrieval system* (pp. 313–323).
Salton, G. (1971). *The SMART retrieval system*. New Jersey: Prentice Hall.
Voorhees, E. (2005). *TREC: experiment and evaluation in information retrieval*. The MIT Press.
Witten, I., & Frank, E. (2005). *Data mining: practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann.
Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In H. P. Frei, D. Harman, P. Schäuble, & R. Wilkinson (Eds.), *SIGIR* (pp. 4–11). ACM.
Xu, J., & Croft, W. B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information and System, 18*(1), 79–112.
Xu, Z., & Akella, R. (2008). A bayesian logistic regression model for active relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 227–234). ACM.
Ye, Z., Huang, X., He, B., & Lin, H. (2009). York university at TREC 2009: relevance feedback track. In *Proceedings of TREC 2009*.
Ye, Z., He, B., Huang, X., & Lin, H. (2010). Revisiting Rocchio's relevance feedback algorithm for probabilistic models. In *AAIRS* (pp. 151–161).
Ye, Z., Huang, X., & Lin, H. (2011). Finding a good query-related topic for boosting pseudo-relevance feedback. In *JASIST* (pp. 748–760).
Zelikovitz, S., & Hirsh, H. (2001). Using LSI for text classification in the presence of background text. In *CIKM* (pp. 113–118). ACM.
Zhai, C. X, & Lafferty, J. D (2001). Model-based feedback in the language modeling approach to information retrieval. In *CIKM* (pp. 403–410). ACM.