Classification based Filtering for Personalized Information Retrieval

Sachintha Pitigala[1], Cen Li[2]
[1] Center for Computational Sciences, MTSU, Murfreesboro, TN, USA
[2] Department of Computer Science, MTSU, Murfreesboro, TN, USA
*spp2k@mtmail.mtsu.edu, Cen.Li@mtsu.edu

***Abstract*— PubMed keyword based search often results in many citations not directly relevant to the user information need. Personalized Information Retrieval (PIR) systems aim to improve the quality of the retrieval results by letting users supply more information than keywords. There are two main problems relate to current PIR systems developed for PubMed: (1) requiring the user to supply a large number of citations directly relevant to search topic, and (2) produces too many search results with high false positive. This paper describes a Classification based multi-stage Filtering (ClaF) approach to address these problems. A small set of citations relevant to the information need is needed from the user. The system automatically finds similar citations to the inputs and builds a larger training set. This training set is used to train multiple text classifiers, each with a different classification scheme. The trained text classifiers are used in a Multi-stage filtering process to find the relevant citations to the user information need. Results show the proposed ClaF system is feasible and produces good retrieval results.**

Keywords: Information Retrieval, Personalized Information Retrieval, Text Classification, PMRA, PubMed.

I. Introduction

SCIENTIFIC literature databases had an exponential growth over the past decade. Google Scholar [1], PubMed [2], The SAO/NASA Astrophysics Data System [3] and CiteSeerX [4] are some of the popular citation databases on the internet. These online databases open a new way of accessing and searching for the information for the scientific community.

PubMed is the largest literature database in the biomedicine field. PubMed is developed and maintained by the National Center for Biotechnology Information (NCBI) at the National Library of Medcine (NLM) [5]. It contains over 24 million biomedicine and related citations covering over 5000 journals ([2], [5]). Given a keyword based user query, PubMed typically returns a large number of citations relevant to the search query. For example, over one-third of PubMed queries returned 100 or more citations [6]. Sifting through these citations to locate the ones that represent the most relevant articles for one's query can be a time consuming process. It is desirable to have search tools that are capable of capturing each user's unique research interest and returning a smaller set of citations of the truly relevant articles from a large literature databases such as PubMed. These types of search tools are referred as Personalized Information Retrieval (PIR) Systems.

For the traditional Information Retrieval (IR) systems, user information needs are provided as user queries consisting of keyword terms. For PIR systems, the unique interests of a user's information need are better captured with the use of additional information provided by the user. Currently, PIR systems can be divided into two main categories based on the way it gathers the user interest. The first category of the PIR systems gathers the user information and interest explicitly from the user ([7], [8], [9], [10], [11]). The second category of PIR the systems gathers its user personalized information implicitly, for example in terms of the click-through links in the search history ([12], [13], [14]). This research focuses on developing a PIR system for the PubMed based on user explicit information.

Many explicit PIR systems allow users to provide additional information about their query through an advanced search option where the user may explicitly enter the area of interest, publication period, journals or authors of interest, along with the query terms ([7], [8]). These additional information further filters the search output, thus reduces the size of the search output.

Yet, it is possible to get more explicit information about the user's query intent in order to deliver more personalized results. For example, explicit PIR systems allow the users to enter a text paragraph to explain his/her information need, or input paragraphs or the abstract of a research article. eTBLAST [9] is an explicit PIR tool for PubMed built based on free text inputs. The inputs provide more information to the search tool than the keywords. It produces better results compared to the traditional keyword based method.

MedlineRanker [10] and MScanner [11] are explicit PIR systems for PubMed that take as input a set of citations that are deemed relevant to a user's information need. The systems derive the information needs from this set and searches for the relevant citations best matching the input. The focus of the systems is on ranking the search results based on the input citations. They do not directly focus on reducing the search output size from PubMed. Both systems require a user to provide at least 100 citations highly relevant to the user interest in order to get reasonable search results. This requirement is unrealistic in many situations.

*Corrosponding Author: Sachintha Pitigala. Email: spp2k@mtmail.mtsu.edu

The goal of this research is to build a PIR System for PubMed that is capable of delivering highly relevant search results, reducing the search output size by limiting irrelevant citations in the search output, and only requires the users to input a small set of citations of the relevant articles. In this study, the proposed PIR system is referred as Classification based Filtering (ClaF) system.

To evaluate the performance of the ClaF system, TREC 2005 dataset [15] is used. This dataset consists of 50 information needs from real biomedicine researchers. Each information need in TREC contains a document pool and each document in the pool is labeled as Definitely Relevant (DR), Possibly Relevant (PR) or Not Relevant (NR) [15].

The ClaF system takes a set of PubMed citations as user input. The input citations represent the user research interest or information need. We call this citation set the user *seed* documents. *Seed* documents typically consist of 5 to 20 citations carefully chosen by the user. It has been reported that learning text classifiers based on a small training data is difficult ([16], [17]). To better illustrate this difficulty, result from a simple experiment is discussed here. To form the training data, first, five documents were randomly selected from the combined DR and PR set of an information need. Then five more documents were randomly selected from the NR set of the same information need. Naive Bayes text classifier is trained using the 10 documents. Table 1 shows the average classification results (averaged over 10 random runs) of the classifiers trained for five different information needs. It is clear that the classification accuracies of the text classifiers are extremely low and it is making many false positive classifications.

In order for the PIR system to be effective in retrieving relevant citations based only on a small number of citations from the user, the system should be able to:

1. Increase the size of the training data based solely on the user *seed* citations while maintaining the quality of the training data;
2. Reduce the false positive classifications
3. Rank the final search output efficiently and effectively.

To achieve the first goal, a method based on PubMed Related Articles (PMRA) [18] and Cosine Similarity [19] is developed. A Text Classification based multi-stage filtering model is used to reduce the number of false positive classifications. Finally, cosine similarity measure and the *seed* citations are used to rank the final predicted relevant documents. Experimental results from 10 different information-needs from the TREC 2005 dataset show that the system produces reliable search results for the given information need.

The rest of the paper is organized as the following: Section 2 discusses the PMRA and the cosine similarity measure and text classification methods used in this study. Section 3 presents the proposed ClaF system, TREC 2005 genomic dataset and preprocessing steps. Section 4 describes the experiment procedure and experimental results of the ClaF system. Section 5 discusses the conclusions about the study and presents the future research directions.

Table 1: Accuracy of Naive Bayes classifiers for five information needs from the TREC 2005 dataset. In this experiment, the training set contains 5 relevant and 5 non-relevant citations from the information need.

Topic ID (Information Need)	Precision	#Articles classified as positive	# Articles correctly classified as positive	# Actual positive articles in dataset
117	0.06617	12360	685	704
146	0.02891	16013	420	432
120	0.02811	13139	318	340
114	0.02905	13903	354	374
126	0.02487	13856	284	302

II. BACKGROUND

To increase the size of the training data based on *seed* documents supplied by a user, a similarity-based approach is developed to find citations from the entire database that are similar to the *seed* citations. Given the size of the PubMed database, to perform a real time similarity computation between each of the *seed* citation and every citation in the database is generally not practical. Therefore, this study uses the PMRA feature [18] to build a small *Target Set*, based on which a larger training data is formed.

A. PubMed Related Articles (PMRA) feature

The PMRA feature computes the similarity between pair-wise citations in the database. The relevancy between two citations is calculated using the words they have in common, with citation length adjustment. Words from title, abstract and Medical Subject Headings (MeSH) terms are used to represent a citation in this algorithm. PubMed related citations are calculated using the entire PubMed database for given citation. This process takes several days to complete [20]. Therefore, PubMed related citation list for any given citation is pre-calculated and sorted in the PubMed. The most relevant citations for any given citation, called the PMRA list, are stored in PubMed database. The PMRA list is a useful feature in PubMed. A PubMed log analysis showed that a fifth of PubMed searches invoke the PubMed related articles (citations), suggested by the PMRA list, at-least once [21].

In our system, the PMRA lists of the user *seeds* are combined to form a *Target Set*. This *Target Set* is then used to find more positive training example for text classification. Cosine Similarity measure is chosen as the similarity measure to find documents similar to the ones in the *seed* set.

B. Cosine Similarity

Cosine similarity is heavily used in the information retrieval and text mining community. A previous study showed that cosine similarity and the overlap model out-performed many other similarity measures in the TREC dataset [22]. Cosine Similarity provides a simple and effective method to compute the similarity between articles by measuring the angle between the two vectors representing the two articles.

In this study, cosine similarity is used to compute the similarity between each citation in the *Target set* and the ones in the *seed* set. The candidate citations with the highest similarity values are added to the positive training example set.

Once the training data set is formed, ClaF system learns the text classifiers based on the training data. Text classification automatically assigns documents into one or more categories based on its content. Popular text classification approaches include the Naive Bayes (NB) classification [23], the Support Vector Machines (SVM) [23], the Rocchio method [24], the regression based models [25], the k-Nearest Neighbor (kNN) method [24], and the Neural Networks [25]. NB, kNN and SVM text classification approaches have been used in the ClaF system.

The following sections briefly explain the theory behind the kNN text classification, the Naive Bayes classification, and Support Vector Machine (SVM) approach.

C. k-Nearest Neighbor (kNN) Text Classification

k-Nearest Neighbor (kNN) algorithm is also known as instance-based learning or lazy learning. The kNN algorithm does not have an explicit training step. During classification, it examines the class labels of *k* nearest neighbors that are the most similar to the test object, and classifies the test object with the majority label from its *k* neighbors. A similarity measure is used to find the *k* nearest neighbors from the training set. Cosine Similarity is used here to find the nearest neighbors. In this study, kNN training set consists of equal number of positive (relevant) and negative (non-relevant) training examples. We need to predefine a value for *k* in the kNN text classifier. In order to break ties in majority vote, an odd integer for *k* such as 1,3,5,7... is often used. The best value of *k* depends on the dataset.

D. Naive Bayes Text Classification

The Naive Bayes is a fast and robust text classification method. It is based on the *posterior* probability model derived using the Bayes theorem [23]. Given a document *d*, its probability of belonging to a class *c* is $P(c|d)$. The goal of the Naive Bayes classification is to find the optimal class for a given document, i.e., the class that gives the maximum posterior probability, $\hat{P}(c|d)$ [23]. This is expressed as:

$$C_{map} = \arg\max_{c\in C} \hat{P}(c|d) \qquad (1)$$

where, C_{map} is the class with the maximum posterior probability, $c \in \{c_1, c_2, c_3, \ldots . c_n\} = C$ is the set of class labels and d is the given document. Then, applying the Bayes theorem and the Naive Bayes conditional independence assumption Equation 1 can be re-written as:

$$C_{map} = \arg\max_{c\in C} \prod_{i=1}^{nd} \hat{P}(t_i|c)\ \hat{P}(c) \qquad (2)$$

where $\{t_1, t_2, t_3, \ldots . t_{nd}\}$ is the set of terms in document d, and nd is the total number of terms in the document. During the training stage, the probabilities, $\hat{P}(c)$ and $\hat{P}(t_i|c)$, are estimated from the training data. $\hat{P}(c)$ is the prior probability of class c.

At the classification stage, given terms $\{t_1, t_2, t_3, \ldots . t_{nd}\}$ for a document d, Equation 2 is used to compute the posterior probability of the document for each possible class, $c \in C$. The class assigned to the document is the one having the highest posterior probability.

E. Support Vector Machines (SVM)

SVM is a popular and powerful algorithm for text classification and many other pattern recognition problems. It is originally a non-probabilistic binary classifier invented by Vapnik and his colleagues [26]. SVM method gives a formal explanation to find the optimal hyper-plane to separate data. Moreover, it finds the optimal hyperplane that maximizes the margin between two data regions. The data points on the marginal hyperplanes are called *support vectors*. If the initial data is not linearly separable in the feature space, SVM uses kernel functions to transform data into a higher dimensional feature space where a hyperplane exists to do the separation. In this study, the LIBSVM software [27] with WEKA [28] is used to perform SVM classification.

III. Methodology

The overall architecture of the proposed text Classification based multi-stage Filtering (ClaF) system is presented in Figure 1. ClaF requires a user to input a small set (e.g., 5 to 10) of citations. These citations represent the user information need and are referred as *seeds*. From the *seeds*, ClaF extracts the useful information for information retrieval. The following section presents the steps ClaF uses to extract the information from the *seeds*.

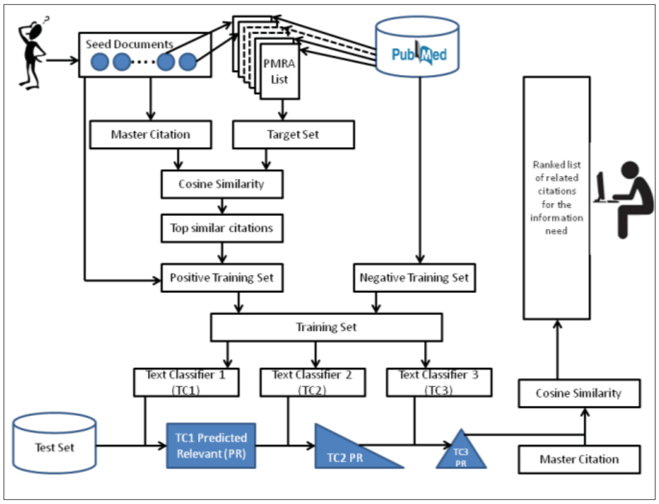

Figure 1: Overview of the ClaF system.

A. Data Preprocessing of user seeds

First, citation title, abstract and the Medical Subject Headings (MeSH) terms are extracted from each user *seed*. Information such as details about the author, affiliation data and journal information are ignored in this study. Then, the title, abstract text and the MeSH terms are tokenized into list of terms. From the document term list, stopwords [29] and words containing only digits are removed. Next, stemming is applied to obtain a normalized term list for the document. Finally, the normalized terms from the title and MeSH terms with subheading qualifier are added again to the normalized term list to give more weight to those terms. Normalized term list from each user *seed* are used to build a *Master Citation* in the ClaF system. This *Master Citation* is used to represent the user information need.

B. Building the Master Citation

The set of user *seeds* collectively represents the user information need. Each single *seed* represents a segment of the user information need. Therefore, it is necessary to combine the *seeds* into a single citation to find the similar citations from the *Target Set*. This single citation is referred as the *Master Citation* in the system.

To form the *Master Citation*, first, a unique word list is created along with document frequencies and term frequencies using the normalized term list from each *seed*. Then, all the terms appearing in two or more *seeds* are added to the *Master Citation* term list. *Master Citation* is a unique representation of user information need. Next, *Master Citation* is used to build a larger training set for text classification in the ClaF system.

C. Expand the Training Set.

The experimental results given in the introduction section have shown the need to have more training examples in order to learn a more accurate text classifier. However, requesting a larger *seed* set from the user is not practical. Therefore, an automated procedure is needed to expand the training set based on the small set of user *seeds*.

To expand the training set, ClaF searches for documents that are the most similar to the *Master Citation* using the Cosine Similarity. To speed up this process, a *Target Set* of citations, e.g., a subset of the PubMed database, is formed from which potential documents are searched.

The PMRA lists [18] are used to build the *Target Set*. For each given citation in PubMed, its PMRA list is pre-calculated. To build the *Target Set*, first, the PMRA list for each user *seed* is retrieved. Then, *seed* PMRA lists are combined into a single citation list. This unique citation list is called as the *Target set*. Next, ClaF finds the documents similar to the *Master Citation* from this *Target set* using the Cosine Similarity. Finally, citations having the highest similarity to *Master Citations* are added to the training set. Together, these newly added citations and the user *seeds* form the positive (relevant) training examples. A similar size document set is randomly selected from the entire PubMed database and labeled as negative (irrelevant) training examples.

Next, a text classifier based multi-stage filtering takes place to gradually refining/reducing the set of citations classified/predicted as relevant.

D. Multi-stage Filtering using Text Classification

At the beginning of the filtering process, 3 classifiers are learned from the expanded training data set. In this study, Naive Bayes (NB), Support Vector Machines (SVM) and k-Nearest Neighbor (kNN) text classifiers are used as the three base text classifiers. The three learned classifiers are applied in 3 stages in refining and filtering of the retrieval results.

Stage 1 text classifier (TC1) is first used to classify the test set into two categories: relevant (positive) and irrelevant (negative). Test set can be the entire PubMed database or a subset of PubMed database selected by the user. For example, if the user is interested in only retrieving the relevant citations published in the last five years. Then, the test set includes only the citations published in those five years. The set of citations predicted as positive by TC1 is often quite large, including many false positives.

To remove the false positives from the retrieval results, citations classified as positive by TC1 undergoes two more classifications using stage 2 Text Classifier (TC2) and stage 3 Text Classifier (TC3) in a pipeline fashion. Only the citations classified as positive from the previous stage are fed into the next classification stage for further refinement.

ClaF uses three-stage text classifier based filtering to refine the set of retrieved citations. A different choice of the base classification scheme at each of the 3 stages can lead to a slightly different final retrieval results. We take a conservative approach in choosing the classification schemes: apply classification schemes having high recalls in the early stages of the filtering pipeline and apply classifiers that are most susceptible in incorrectly remove true positives in later stages in the filtering pipeline, i.e., to preserve the true positives in the retrieval results as much as possible.

This approach is different from the standard voting schemes used for classification, where the accuracy of the voting schemes doesn't depend on the order of the classifiers used. This approach is also different from the active learning methods. While most active learning methods focus on improving the classification accuracy by incrementally improving the training data, in ClaF, the training data is improved just once through expansion. All the text classifiers are trained using the same expanded training data. After that, ClaF focuses on reducing the false-positives in the search output rather than improving the accuracy of the text classifier.

The three-stage filtering method may be generalized into filtering pipeline with more or less stages, i.e., multi-stage filtering. For example, one may use two-stage or four-stage filtering with two or four classifiers respectively. Classification schemes other than Naive Bayes, kNN, and SVM may be used in each stage of the process. The conservative approach should be used to order the classifiers in the filtering stages.

E. Ranking the Final Output

The classification results from TC3 represent a much-improved set of highly relevant citations to the user information need. However, it may still contain some of the false-positives. As the final step, ClaF ranks the resulting set of citations based on the Cosine Similarity of each against the *Master Citation*. The top ranked citations are presented as the final retrieval results.

IV. Results and Discussion

The ClaF methodology is tested and validated using the TREC 2005 ad hoc retrieval task dataset [15]. It contains 50 information needs (topics) from the real biologists. The entire document collection for the 50 topics contains 34,633 unique PubMed citations. Each information need (topic) has a corresponding set of labeled citations ranges between 290 and 1356 [15]. Expert biologists have labeled each citation as to whether or not it is relevant to the information need. The labels can be one of the following three: Definitely Relevant (DR), Possibly Relevant (PR) and Non Relevant (NR) for the given topic. The 10 topics having the highest number of relevant documents (definitely relevant and possibly relevant) are used in this study. Those topic numbers are 117, 146, 120, 114, 126, 109, 142, 111, 107 and 108. Next, the experimental procedure of this study is described.

A. Experiment Procedure

For each chosen information need (topic), ClaF uses the following steps to form the user information need and to retrieve the relevant citations:

- n ($n = \{5,10,15,20,25\}$) citations are randomly selected from the Definitely Relevant (DR) and Possibly Relevant (PR) set of the topic. Those n citations are labeled as the user *seeds* for the current topic;
- The PMRA lists for the *seeds* are retrieved and combined to form the *Target Set*;
- The *seeds* are pre-processed into terms and used to form the *Master Citation*;
- N (N=50) citations that have the highest cosine similarity to the *Master Citation* are computed from the *Target Set*; This set and the original n *seeds* form the positive training examples.
- Randomly select n+N citations from the TREC 2005 genomic track dataset to form the negative examples;
- Train the three Text Classifiers using the expanded training data;
- Classify the TREC 2005 Genomic dataset using TC1.
- TC1 classifies a subset of citations as “relevant”.
- Apply TC2 to refine and reduce the set of the “relevant” citations;
- Apply TC3 to further refine and reduce the set of the “relevant” citations from TC2;
- Compute and rank the Cosine similarities between the “relevant” citations from TC3 and the *Master Citation.*

Each experiment is repeated 10 times by randomly selecting *seeds* from the given information need. *Seed* set size (n) range from 5 to 25 with increment of 5. Results of the 10 information needs are presented in the next section.

B. Experimental Results

The experiments are designed to test the effectiveness of the ClaF system in terms of the effectiveness of each of its three main steps (1) expanding training set size by building *Target Set* and forming *Master Citation*, (2) multi-stage filter, and (3) ranking of the final output.

B.1 Improvements from Expanding the Training Data Set

If the size of the initial user *seeds* is 5 ($n = 5$), then the initial training data size is 10 with the negative training examples. After expanding the training set, a larger training set size of size 110 is obtained. This larger training set is used to train the three base classifiers. For kNN, three nearest neighbors are used to classify the new instances. Linear SVM method from the LibSVM [27] in WEKA [28] is used. The classification accuracies obtained using the expanded training data are compared against those of the original training data (*seed* only training data). Table 2 shows the average improvement of classification accuracies for the 10 information topics. Equation 3 calculates the improvement of accuracy for a topic. The average improvement over five different training sets is reported.

$$AI = \frac{(AETS - ASTS)}{ASTS} * 100\ \% \qquad (3)$$

where, *AI = Accuracy Improvement*, *ASTS= Accuracy with the Seed only Training Set (Initial Training Set)* and *AETS= Accuracy with the Expanded Training Set*.

From Table 2, it is clear expanding the training set using the *Target Set* and *Master Citation* lead to a big improvement of the classification accuracies of base text classifier across all 10 information needs. The PMRA feature helps to build a high quality small *Target Set*, and Cosine Similarity is effective in computing citations having the highest similarity to the *Master Citation* from the *Target Set.*

Table 2: Improvement of Classification Accuracy for the three base classifiers using expanded training data

Topic ID	Average Improvement (%)		
	NB	3-NN	SVM
117	+ 61.20	+ 184.81	+ 60.92
146	+ 82.26	+ 155.81	+ 14.86
120	+ 150.31	+ 334.95	+ 61.13
114	+ 73.82	+ 158.35	+ 68.80
126	+ 150.31	+ 110.65	+ 11.75
109	+ 67.20	+ 82.81	+ 9.74
142	+ 182.62	+ 190.39	+ 150.09
111	+ 89.73	+ 225.86	+ 139.93
107	+ 66.96	+ 104.96	+ 24.58
108	+ 67.37	+ 131.71	+ 27.42

B.2 Multi-stage Filtering

ClaF uses the multi-stage classification based filtering to find the relevant citations from the whole dataset. The approach to select the classification schemes for each of the three stages is to select classification schemes that are less likely to filter away the true positive citations in the early stages of filtering. Since Naive Bayes (NB) classifier has a higher recall value than that of SVM and 3-NN classifiers, NB classifier is used in the first stage filtering. The 3-NN classifier is used in the second stage filtering, and SVM is used in the final stage of the filtering process.

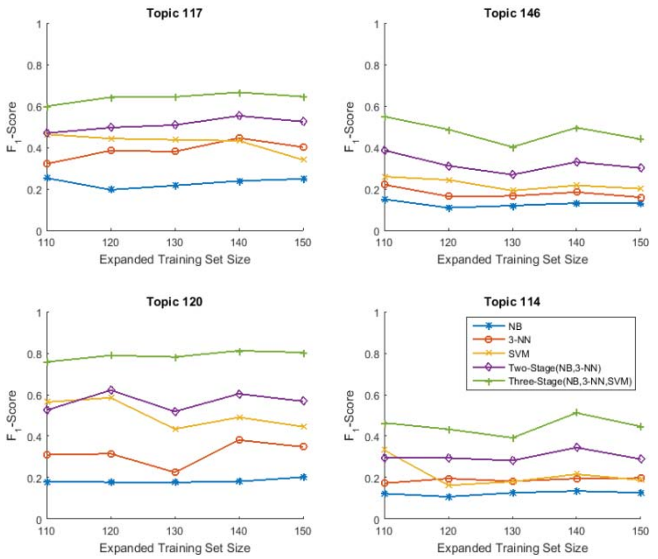

Figure 2: F_1-*Scores* computed for topics 117, 146, 120 and 114 using Three-stage (NB, 3-NN, SVM) filtering, Two Stage (NB, 3-NN) filtering, and NB, SVM and 3-NN only methods.

For comparison purposes, filtering performed with a two-stage process using NB and 3-NN, as well as three one-stage processes with just NB, 3-NN and SVM are also performed. Figure 2 shows the F_1-*Score* for four topics using the five different filtering methods. F_1-*Scores* are calculated using Equation 4. It provides a balanced measure of both recall and precision.

$$F_1 - Score = 2 \cdot \frac{(precision \cdot recall)}{(precision + recall)} \qquad (4)$$

As shown in Figure 2, The F_1-*Scores* of the classification results from the three-stage method (NB, 3-NN, SVM) are higher than the other four methods for all four topics.

The final search output is dependent on the order of the text classifiers chosen for the 3 stages. For example a 3-stage filtering with the 3 classifiers in the order: (1) NB, (2) 3-NN, and (3) SVM produces a different result than one with the 3 classifiers in the order: (1) NB, (2) SVM and (3) 3-NN. The first ordering produces a better result with higher F_1-*Scores*. This is because NB is a text classifier that generates classifications with high recall for the given topic. In the second stage 3-NN is used, followed by the SVM text classifier. Since SVM outperformed the other two text classifiers in the one-stage method, SVM is used in the third stage to get an accurate final output.

B.3 Ranked Retrieval Results

ClaF ranks the set of predicted relevant citations from the three-stage process against the *Master Citation* using cosine similarity. The top *N* ranked citations are considered the final retrieval results to be presented to the user for the information need. The retrieval accuracy is computed in terms of the percentage of the top *N* citations having the label of Definitely Relevant (DR) or Possibly Relevant (PR) to the given information need. Table 3 shows the retrieval accuracy of the top 10 citations (P10) and the top 100 citations (P100) in the final search output for the 10 topics.

Table 3: Retrieval accuracy of the top 10 citations (P10) and the top 100 citations (P100) in the final retrieval results.

Topic ID	P10	P100	Topic ID	P10	P100
117	0.9100	0.8462	**109**	0.9520	0.8910
146	0.9100	0.8430	**142**	0.5520	0.6166
120	0.8760	0.7836	**111**	0.6800	0.6698
114	0.8080	0.5740	**107**	0.6700	0.5554
126	0.5820	0.4244	**108**	0.6720	0.3890

It is observed that P10 measure is greater than 0.8 for five information needs. That is, 8 out of the top 10 retrieved citations are relevant to the information need. P10 measure for all the other topics is also greater than 0.55. P100 measure is greater than 60% for six information needs. That is, 60 or more citations from the top 100 retrieved citations are relevant to the information need. Considering that the percentage of citations relevant to each topic is rather small in the entire database, these results are quite encouraging. However, a much lower accuracy is observed for a few topics, e.g., P100 for topic 126 and topic 108. This may be attributed to the fact that there are too few positive citations for the topic. For example, topic 108 has a total of 203 positive citations. For each experiment *n* (*n={5,10,15,20,25}*) positive citations are selected to form the *seed* citation set. The number of remaining positive citations is very small compared to the size of the TREC dataset. This makes the retrieval tasks harder if only the top 100 citations are to be returned. However the P10 and P100 results from the ClaF system present a 13% and 22% improvement over the results reported by the systems during the TREC conference [15]. These results make the ClaF approach a more feasible for personalized retrieval.

V.CONCLUSIONS

The main goal of this study is to build a PIR system based on a small set of input citations. Also, this PIR system focused on retrieving a small set of citations as the search output by eliminating the false-relevant citations. One of the main problems with PIR system is to try to achieve high retrieval quality by training a PIR system using a small set of user provided *seed* citations. In the proposed ClaF, first, the training set is expanded to a reasonably large dataset based on the *seed* citations. Similar citations to the *Master Citation* from the PMRA based dataset are used in expanding the training dataset. This expanded training data allow the NB, kNN and SVM text classification schemes to produce better quality classifiers. Experimental results show that the procedure of expanding training set is successful in achieving its goal. Text classifiers trained from the expanded training set are used in the three-stage filtering method. Three-Stage filtering method is used to successively removing the false positively classified citations from the results. For all the information needs, the F_1-*Scores* of the three-stage method improved dramatically over the base text classifiers. Also, there is a significant improvement in P10 and P100 measures for a majority of the information needs. Therefore, one can conclude that three-stage filtering method improves the quality of the final search output.

Three-Stage filtering approach can be used for other information retrieval scenarios. The three-stage method may be generalized into multi-stage filtering approach. Our planned next step is to adapt and test the multi-stage method for other domains. We also plan to experiment with using other classification schemes to build the base classifiers. In addition, we plan to experiment with incorporating other feature selection methods and advanced similarity measures into the ClaF system.

REFERENCES

[1] *Google Scholar*. Retrieved 11 16, 2014, from Google Scholar http://scholar.google.com/

[2] *PubMed*. Retrieved 11 16, 2014, from PubMed: http://www.ncbi.nlm.nih.gov/pubmed

[3] The SAO/NASA Astrophysics Data System. Retrieved 11 16, 2014, from Astrophysics Data System: http://adswww.harvard.edu/

[4] *CiteSeerX*. Retrieved 11 16, 2014, from CiteSeerX: http://citeseer.ist.psu.edu/index

[5] *PubMed Fact Sheet*. Retrieved 11 16, 2014, from U. S National Library of Medicine : http://www.nlm.nih.gov/pubs/factsheets/pubmed.html

[6] Islamaj Dogan R, Murray GC, Neveol A, et al. Understanding PubMed user search behavior through log analysis. Database. 2009 doi:10.1093/database/bap018

[7] *PubMed Advanced Search Builder*. Retrieved 11 16, 2014, from U. S National Library of Medicine : http://www.ncbi.nlm.nih.gov/pubmed/advanced

[8] Google Advanced Search. https://www.google.com/advanced_search?hl=en

[9] Mounir Errami, Jonathan D. Wren, Justin M. Hicks, Harold R. Garner: eTBLAST: a web server to identify expert reviewers, appropriate journals and similar publications. Nucleic Acids Research 35(Web-Server-Issue): 12-15 (2007)

[10] Fontaine JF, Barbosa-Silva A, Schaefer M, et al. MedlineRanker: flexible ranking of biomedical literature. Nucleic Acids Res. 2009;37:W141–W146.

[11] Poulter G, Poulter G, Rubin D, et al. MScanner: a classifier for retrieving Medline citations. BMC Bioinformatics. 2008;9:108.

[12] Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing search via automated analysis of interests and activities. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2005) 449–456

[13] Qiu F. and Cho J. Automatic identification of user interest for personalized search. In Proc. 15th Int. World Wide Web Conference, 2006, pp. 727–736.

[14] Shen X., Tan B., and Zhai C. Implicit user modeling for personalized search. In Proc. Int. Conf. on Information and Knowledge Management, 2005, pp. 824–831.

[15] Hersh WR, Cohen AM, et al. The Fourteenth Text Retrieval Conference (TREC 2005) NIST; 2005. TREC 2005 Genomics track overview.

[16] Lang, K. Newsweeder: Learning to filter netnews. In Machine Learning: Proceedings of the Twelfth International Conference (ICML '95), pp. 331–339.

[17] Lewis DD, Yang Y, Rose TG, Li F. RCV1: A new benchmark collection for text categorization research. J Mach Learn Res 2004;5:361–97.

[18] Lin J, Wilbur WJ. Pubmed related articles: a probabilistic topic-based model for content similarity. BMC Bioinformatics 2007;8:423

[19] Lee, M.D., Pincombe, B.M., & Welsh, M.B. (2005). An empirical evaluation of models of text document similarity. Proceedings of the 27th Annual Conference of the Cognitive Science Society, pp. 1254-1259. Mahwah, NJ: Erlbaum.

[20] PubMed Online Training : Related Citations. Retrieved 11 16, 2014, from U. S National Library of Medicine : http://www.nlm.nih.gov/bsd/disted/pubmedtutorial/020_190.html

[21] Lin J, DiCuccio M, Grigoryan V, Wilbur WJ: Exploring the Effectiveness of Related Article Search in PubMed. In Tech. Rep. LAMP-TR-145/CS-TR-4877/UMIACS-TR-2007-36/HCIL-2007-10. University of Maryland, College Park,Maryland; 2007.

[22] Rorvig M. Images of similarity: a visual exploration of optimal similarity metrics and scaling properties of TREC topic-document sets. Journal of the American Society for Information Science, Volume 50 Issue 8. 1999.

[23] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schtze. Introduction to Information Retrieval. Cambridge University Press, 2008 pg 234-250,293-320.

[24] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schtze. Introduction to Information Retrieval. Cambridge University Press, 2008 pg 269-277.

[25] Vladimir Cherkassky, Filip M. Mulier. *Learning from Data: Concepts, Theory, and Methods* . WILEY-INTERSCIENCE, 2007.

[26] Burges, Christopher J.C. "A Tutorial on Support Vector Machines for Pattern." Data Mining and Knowledge Discovery, 1998.

[27] C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[28] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten; The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1 (2009).

[29] PubMed Stopwords (11 24, 2014, date last accessed): http://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T43/