Using supervised machine learning to automatically build relevance judgments for a test collection

Mireille Makary / Michael Oakes/ Ruslan Mitkov
RGCL
University of Wolverhampton
Wolverhampton, UK
m.makary@wlv.ac.uk / michael.oakes@wlv.ac.uk / r.mitkov@wlv.ac.uk

Fadi Yammout
Computer Science Department
Lebanese International University
Beirut, Lebanon
fadi.yamout@liu.edu.lb

Abstract—This paper describes a new approach to building the query based relevance sets (qrels) or relevance judgments for a test collection automatically without using any human intervention. The methods we describe use supervised machine learning algorithms, namely the Naïve Bayes classifier and the Support Vector Machine (SVM). We achieve better Kendall's tau and Spearman correlation results between the TREC system ranking using the newly generated qrels and the ranking obtained from using the humanly built qrels than previous baselines. We also apply a variation of these approaches by using the doc2vec representation of the documents rather than using the traditional tf-idf representation.

Keywords—test collections, evaluation, qrels, relevance judgments, machine learning, doc2vec, tf-idf

I. INTRODUCTION

The evaluation of information retrieval systems requires the use of a standard framework which is known as test collections. A test collection consists of a set of documents, a set of topics and a set of relevance judgments or query-based relevance sets (qrels) [1]. The task of building the qrels is costly and requires much effort from human assessors to judge the relevance of each document retrieved for the topic submitted to the information retrieval system. When it comes to large scale environments such as the web, this process becomes infeasible; it is not possible to judge millions of documents. A well-known framework which allows large-scale evaluation for text retrieval is the TREC test collections. The Text REtrieval Conference (TREC) is organized annually by NIST. In addition to the set of documents, the list of 50 topics, TREC provides each test collection with a relevance judgment list built by human assessors based on a pooling technique [2]. The research groups which participate in building the test collection are given the documents and topic sets. Each group uses the topics provided and retrieves a ranked set of documents using their information retrieval system. They submit their runs back to NIST where the organizers will form a pool of documents of depth 100 for each topic, by collecting the top 100 documents from each run. They remove duplicate documents. The resulting pool is then judged by human assessor to determine its relevance. This forms the relevance judgment list or the query-based relevance sets (qrels). Any document not found in the pool is considered to be non-relevant. Building the qrels is a major task and consumes a lot of time, resources and money. Several methods have been proposed to build the qrels with reduced human intervention. These previous methods have been partially successful in automating the generation of relevance judgments, as evaluated by the correlation coefficient for the ranking of the systems using the newly generated qrels and the ranking obtained from using the human qrels. The focus of this paper is to propose a new method which a) can achieve a better correlation between system-generated and human-generated rankings, b) require no user intervention and c) be applied to any type of test collection. We review some of the related work completed in previous years in section 2. In section 3, we describe the new techniques which involve using supervised machine learning algorithms. We described the experiments conducted and report the results obtained using the TREC6, 7, and 8 [12, 13, 14] test collections in section 4. We conclude in section 5 and propose a future direction for this work.

II. RELATED WORK

To test the reliability of the pooling technique, Zobel [3] constructed a series of experiments that ended up by proving its effectiveness in evaluating retrieval systems and their rankings. However, only 50%-70% of the relevant documents are discovered especially for the queries that have a large number of answers. System effectiveness was little changed when the pool size was increased. He also measured the degree to which a certain system is contributing to the pool, by removing its results from the pool. Despite all the variations made to the pooling technique, it was proven reliable to build the qrels. Several studies over the years tackled the problem of ranking the retrieval systems with incomplete or unavailable relevance judgments. Soboroff et al. [4] were the first to suggest a method based on the random sampling (RS) of relevant documents in the pool. However, their method required knowing the mean and standard deviation of the number of actual relevant documents in the pool which is not available in practice. Aslam and Savel quantified the similarity of the retrieval systems by assessing the similarity of their retrieval results and they devised a new measure for this quantification (Average System similarity, ASS) which evaluates the system based on its performance rather than on its popularity. This would not penalize novel systems which produced very different sets of qrels from the others [5]. Wu and Crestani [6] used a reference count (RC) method to rank the systems

without relevance judgments, based on assigning a score to each document based on the number of systems which retrieved that document. A data fusion technique (CB) was proposed by Nuray and Can [7]. They combined the top b documents from each of the k participating systems. Then the top s% (where s is a percentage rather than a fixed number of documents) of the merged results were considered as the "pseudo-qrels". Examining the uniqueness of systems, Spoerri [8] ranked the participating teams rather than the entire set of runs. Only one run was chosen for a team if several similar runs had been submitted. Sakai et al. [9] ranked the documents by the number of runs which returned the document and then by the sum of the ranks of that document in the different runs. In work that involved clustering [10], Shi et al. suggested a method to improve the negative effect of different participating TREC systems which produced very similar retrieval results. Thus all systems were evaluated and then clustered into different subsets. In each subset, only one system was selected as a representative for that cluster and therefore only the results returned by the representative were used for evaluation. The results obtained by their clustering technique (Average System Similarity based on Clustering, ASSBC) outperformed all previously described methods. In this paper, we also aim to automatically generate the qrels without any human intervention and we approach this problem as a ranking problem. We test our techniques which use the Naïve Bayes classifier and the Support Vector Machine using a linear kernel on three of the test collections which were used in previous studies to show that our method is not dependent on just one test collection. We show that we achieve better correlations than the previously described methods. The experimental design and the algorithms of the new techniques are described in the in the following section.

III. EXPERIMENTAL DESIGN

Supervised machine learning algorithms require some knowledge about the data which needs to be classified. Therefore, if we need to build a set of relevance judgments, we need to have an initial training set of documents which can be considered as relevant to the topic in question. Based on the hypothesis which states that if a document was retrieved by more than one system, it is likely that the document is relevant to the topic [4, 6], for each topic, we select the set of documents which were retrieved by more than S% of the TREC systems to be our training set for both methods described later. The S% is defined as the minimum percentage which ensures that each topic has at least one document in the set which we consider relevant to that topic. In two approaches, we used the Naïve Bayes (NB) classifier and a Support Vector Machine (SVM) with linear kernel to classify documents and hence build the qrels.

A. *First approach: classify documents as relevant and non-relevant*

For each topic in the test collection, we consider the documents retrieved by S% of the TREC systems as relevant to the topic for which they were retrieved and then we select the same percentage S of documents which were retrieved the least number of times by the TREC systems, and we consider them as non-relevant. Thus the documents in the top S% will have a label of "Relevant", while the documents in the lowest S% will have a label "Non Relevant". We then use these two sets as training sets for the classifiers (whether the NB or SVM), then we use the trained classifiers to predict the label of each remaining document retrieved for that topic. The predicted label will be either "Relevant" or "Non Relevant". We repeat the same process for all the 50 topics. At the end of that process, we will have an automatically generated relevance judgments list where each document in the pool is labeled with a binary relevance value, either relevant or non-relevant for the topic for which it was retrieved.

B. *Second approach: classify documents by topic*

In this second approach, instead of selecting two different document sets, the highest S% and the lowest, we create only one training set for all the topics. The labels we use for the documents will be the topic id rather than the "relevant"or "non-relevant" label used in the first approach. We start first by selecting, for each topic, the documents retrieved by S% of the TREC systems. We assign the topic id (e.g. 401 from TREC8) as a label for each document in this set. Thus for each topic we label the S% of documents with their id so for topic 1, S% retrieved for that topic will have a label 1, for topic 2, the S% retrieved for the topic will be labeled 2, etc. So we will end up with a pool of documents where each document is labeled with a number (e.g. doc1 topic5, doc100 topic 48, etc.). The labeled documents were used as training data for the classifiers (NB and SVM). Next, we used the trained classifiers to predict the topic id, or label for each remaining unlabeled document retrieved from the initial pool of documentsin a "multiclass" classification. In this second approach, the number of relevant documents can be expanded because the classifier could predict that a document belongs to a certain topic, although it was initially retrieved at a rank lower than 100 and was not picked when forming the pool for that topic. The documents used from the training set are also considered relevant to the topic. We go back to the initial documents retrieved for a topic and then we check the label of each document. If it is the same as the topic for which it was retrieved, it will be considered relevant otherwise, we mark it as non-relevant to that topic.

To evaluate both approaches, we compute the mean average precision (MAP) of the systems using the trec_eval package. Then, we measure the Kendall's tau and Spearman correlations between the TREC systems ranking using the newly obtained qrels and the ranking obtained from using the humanly generated qrels. We compare our results to the scores reported from previous studies. Both these approaches were tested first using the tf-idf representation of the documents. We repeated the experiments using the doc2vec [11] document representation. The results obtained from tf-idf were better and this was expected since we have a very small training set for the doc2vec to learn enough about the documents. The NB classifier has also a smoothing parameter alpha which can be tuned. Our experimental details and results are shown in section 4.

IV. EXPERIMENTS

We tested both approaches described in the previous section on TREC test collections. We used TREC6, 7 and 8 to be able to compare our results with previous studies. TREC6 had 74 participating systems, 103 systems participated in TREC7, and TREC8 had 129 participating systems. The S% of the systems which guarantees that each topic has some documents returned was set to 80% for TREC7 and 8, while for TREC6 we used 75% of the systems.

Using the first classification approach which classified the documents as relevant or non-relevant and with the use of the tf-idf representation for the documents, the NB seems to give better correlation results for TREC6, while the SVM using a linear kernel works better for TREC7 and TREC8. We show the results obtained in Table 1.

TABLE I.

	Using SVM		**Using NB**	
	Kendall's tau	**Spearman**	**Kendall's tau**	**Spearman**
TREC6	0.5266	0.7145	**0.5408**	**0.7322**
TREC7	**0.4328**	**0.5116**	0.4250	0.4769
TREC8	**0.5259**	**0.7646**	0.4617	0.7312

Table1: *Kendall's tau and Spearman correlation based on MAP values for TREC6, 7 and 8 using relevant/ non-relevant classification*

The alpha smoothing parameter has a default value of 1 for the Naïve Bayes classifier. We tested different values of alpha: 0.1, 0.2, 0.3… 1. For an alpha value of 0.1, we saw better correlations than in Table 1, except for TREC6, as shown in Table 2.

TABLE II.

	Using NB – alpha 0.1	
	Kendall's tau	**Spearman**
TREC6	0.4669	0.6433
TREC7	0.4413	0.4985
TREC8	0.4559	0.7320

Table2: *Kendall's tau and Spearman correlation based on MAP values for TREC6, 7 and 8 with alpha=0.1 using relevant / non-relevant classification*

The second classification approach which labels the documents by topic id allows discovering more documents for a topic which may not have been part of the pool judged by human assessors. This second approach seems to provide better correlation results when compared with the first classification method. We also tested different values for the smoothing parameter alpha when using the NB classifier and the best value was also obtained for a 0.1 alpha value. All the results are reported in Table 3 and Table 4.

TABLE III.

	Using SVM		**Using NB**	
	Kendall's tau	**Spearman**	**Kendall's tau**	**Spearman**
TREC6	0.5712	0.7631	**0.5864**	**0.7749**
TREC7	0.4116	0.5223	**0.5128**	**0.6386**
TREC8	0.4494	0.7266	**0.5144**	**0.7821**

Table3: *Kendall's tau and Spearman correlation based on MAP values for TREC6, 7 and 8 using topic classification*

TABLE IV.

	Using NB – alpha 0.1	
	Kendall's tau	**Spearman**
TREC6	0.5887	0.7787
TREC7	0.5661	0.6746
TREC8	0.5330	0.7907

Table4: *Kendall's tau and Spearman correlation based on MAP values for TREC6, 7 and 8 with alpha=0.1 using topic classification*

The Kendall's tau and Spearman correlation scores achieved from the classification by topic using the NB classifier outperform the first classification technique which classifies the retrieved documents as relevant or non-relevant. This is due to the fact that the classification by topic can find new relevant documents which are not in the pool formed initially. The best correlations shown in table 4 resulted from tuning the alpha parameter to a 0.1 value.

Now we compare our best technique, classification by topic, with the previous methods which were discussed in the related work section (see table 5). The Spearman correlation values based on MAP scores were reported by the authors for the TREC6 and 7. For TREC8, only Soborof et al. [4] reported results. He obtained an average of 0.5 for Kendall's tau, while we obtained a value of 0.533.

TABLE V.

	RS	**RC**	**CB**	**Single %**	**ASS**	**ASSBC**	**NB By Topic**
TREC6	0.436	0.384	0.717	0.618	0.630	**0.854**	**0.778**
TREC7	0.411	0.382	0.453	0.550	0.585	0.631	**0.674**

Table4: *Spearman correlation based on MAP values for TREC6, 7*

The ASSBC method which divides the TREC systems into different clusters [10] provides the highest Spearman coefficient for TREC6, but this high value could be achieved only after removing 57 TREC systems out of 74 (78%) as a result of dividing the systems into 16 clusters and therefore using 16 systems as representative of the clusters. This number was determined empirically. Our technique does not exclude any participating system in the process of building the qrels. We were able to achieve better results than the remaining previous techniques for TREC7, as we obtained the best correlations overall.

The same set of experiments was repeated using the doc2vec representation instead of the traditional tf-idf. We divided our data into three categories: training data, cross validation data and test data. We used as training data 50% of the documents retrieved by the S% of the systems, while the other 50% was used for cross validation of the doc2vec model. The remaining documents in the pool constituted the test data

which had to be labeled using the trained doc2vec model and both the NB and SVM classifiers. The test data in our case is much larger than the training data. The number of documents (from the S% cutoff) used to train both the classifiers and the doc2vec model is very low compared to the number of documents for which a label must be predicted. Yet, the results are not far from the ones we achieve using the tf-idf. We summarize in table 6 the Spearman correlation coefficients computed for the different test collections using both classifiers and the two different approaches. The parameters used for the doc2vec model were as following: min_count=1, window=10, size=100, sample=1e-4, negative=5, workers=8.

TABLE VI.

	Using SVM		Using NB	
	By Topic	**Relevant/Non-Relevant**	**By Topic**	**Relevant/Non-Relevant**
TREC6	0.6257	0.6813	0.7555	0.7550
TREC7	0.6175	0.5594	0.6116	0.5158
TREC8	0.7293	0.6334	0.7081	0.6839

Table6: *Spearman correlation based on MAP values for TREC6, 7 and 8 using doc2vec document representation*

The experiments described so far constitute an extrinsic evaluation of automatically-generated qrels, where we evaluate the ability of the qrels to reproduce the system rankings produced by the human judges at TREC. We now describe an intrinsic evaluation of our qrels, where we evaluate the accuracy of the generated qrels when compared to the qrels built by the human assessors. We compute the recall metric which is the number of documents we judge relevant using our automated techniques out of the total number of relevant documents which were judged by human assessors for all topics. And then we compute the precision metric which represents the number of relevant documents retrieved out of the total number of retrieved documents at a particular rank. To this end, we computed the precision and recall measures at different ranks (@5, @10, and @20… @ 100, @ 20 ... @ 1000). The formula used for the precision metric is shown in equation (1) below:

$$\text{Precision} = d_{AH} / d_A \qquad (1)$$

Where d_{AH} is the total number of documents judged relevant by both the classifier and the human judges, and d_A is the number of documents judged relevant by the classifier. As for the recall metric, the formula used is shown below:

$$\text{Recall} = d_{AH} / d_H \qquad (2)$$

Where d_{AH} is also the total number of documents judged relevant by both the classifier and the human judges, and d_A is the number of documents judged relevant by human assessors. These two measures can be combined into the F-score, which is their harmonic mean. The F value at a certain rank (i) is computed using the formula below, where p is the precision at rank (i) and r is the recall at rank (i).

$$F=2 / ((1/p+1/r)) \qquad (3)$$

We plot the values obtained for our experiments which used supervised machine learning algorithms.

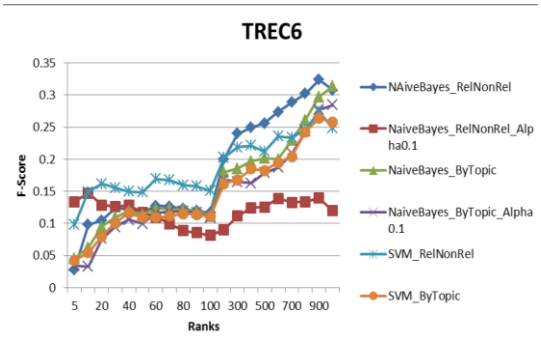

Figure 1. *F-score values obtained for all classification approaches for TREC6*

As shown in figure 1, the first classification approach which labels the documents as relevant or non-relevant using the Naïve Bayes classifier seems to give better F-score values at lower ranks starting at rank 300 for TREC6, while we can see that the SVM classifier works better at higher ranks which might be more important in some real life search scenarios.

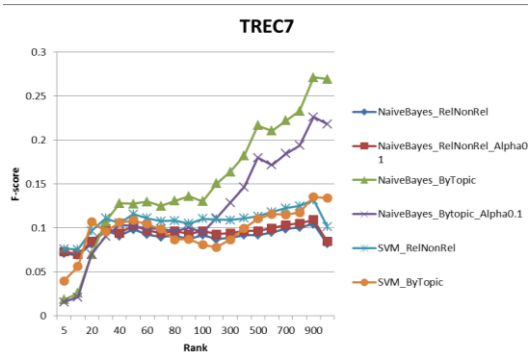

Figure 2. *F-score values obtained for all classification approaches for TREC7*

For TREC7 F-scores shown in figure 2, the second classification approach which labels the documents by topic id using the Naïve Bayes classifier seems to give better F-scores starting at rank 50, while the SVM classifier has a higher F-score for the top 20 ranks. This result is consistent with the best Spearman correlation value obtained for TREC7.

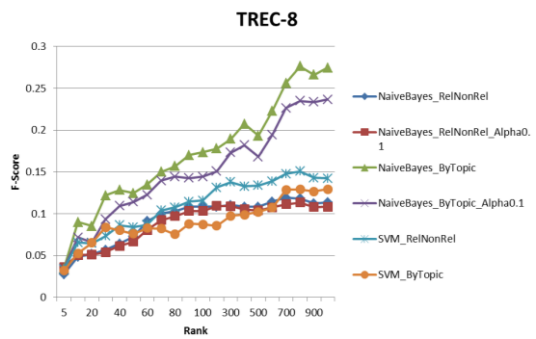

Figure 3. *F-score values obtained for all classification approaches for TREC8*

The results for our classification of the documents by topic, our second approach, using the Naïve Bayes classifier are shown in figure 3 for TREC8. These results are similar to what was reported for TREC7. And also in this case, the highest F-scores are consistent with the highest Spearman correlation value obtained in the extrinsic evaluation.

V. Conclusion

In this paper, we presented two different approaches using supervised machine learning which lead to building the set of relevance judgments for a test collection automatically, without any human intervention. The techniques are simple, yet efficient and outperform almost all previous methods which tackled the same problem. Our methods have been successfully applied to different TREC collections. A future direction for the work will be to test these approaches on non-TREC and non-English test collections to evaluate their effectiveness.

References

[1] Cleverdon C. The cranfield tests on index language devices. Aslib Proceedings, Volume 19, pages 173– 192, 1967.

[2] Spärck Jones K. and van Rijsbergen C.J. Information retrieval test collections, Journal of Documentation, 32, 59-75, 1976.

[3] Zobel J. How reliable are the results of large-scale information retrieval experiments? In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 307-314, 1998.

[4] Soboroff I., Nicholas C., and Cahan P. Ranking retrieval systems without relevance judgments, In Proceedings of ACM SIGIR 2001, pages 66–73, 2001.

[5] Aslam J. A. and Savell R. On the effectiveness of evaluating retrieval systems in the absence of relevance judgments. In Proceedings of ACM SIGIR 2003, pages 361–362, 2003.

[6] Wu S. and Crestani F. 2003 Methods for ranking information retrieval systems without relevance judgments, Proceedings of the 2003 ACM symposium on Applied computing, March 09-12, 2003, Melbourne, Florida

[7] Nuray R. and Can F. Automatic ranking of information retrieval systems using data fusion, Information Processing and Management, 42:595–614, 2006.

[8] Spoerri A. 2007 Using the structure of overlap between search results to rank retrieval systems without relevance judgments, Information Processing and Management: an International Journal, v.43 n.4, pp.1059-1070, July, 2007Efron M.: Using multiple query aspects to build test collections without human relevance judgements, SIGIR, 2009.

[9] Sakai T., Chin-Yew L. Ranking Retrieval Systems without Relevance Assessments – Revisited, The Third International Workshop on Evaluating Information Access (EVIA), June 15, 2010, Tokyo, Japan

[10] Shi Z., Li P., Wang B. Using Clustering to Improve Retrieval Evaluation without Relevance Judgments, Coling 2010: Poster Volume, pages 1131–1139, Beijing, August 2010

[11] Le Q, Mikolov T. Distributed Representations of Sentences and Documents, ICML14, 1188-1196, 2014

[12] Harmon, D.K., Vorhees, E.M.: Overview of the Sixth Text Retrieval Conference (TREC-6). DIANE Publishing Company (1996)

[13] Harmon, D.K., Vorhees, E.M.: Overview of the Seventh Text Retrieval Conference (TREC-7). DIANE Publishing Company (1996)

[14] Harmon, D.K., Vorhees, E.M.: Overview of the Eighth Text Retrieval Conference (TREC-8). DIANE Publishing Company (1996)