Building query-based relevance sets without human intervention

Mireille Makary

A thesis submitted in partial fulfilment of the requirements of the University of Wolverhampton for the degree of Doctor of Philosophy

June 2019

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or any other purpose (unless otherwise indicated). Save for any express acknowledgments, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

Signature: ……

Date: ……

Abstract

Test collections are the standard framework used in the evaluation of an information retrieval system and the comparison between different systems. A text test collection consists of a set of documents, a set of topics, and a set of relevance assessments which is a list indicating the relevance of each document to each topic. Traditionally, forming the relevance assessments is done manually by human judges. But in large scale environments, such as the web, examining each document retrieved to determine its relevance is not possible. In the past there have been several studies that aimed to reduce the human effort required in building these assessments which are referred to as qrels (query-based relevance sets). Some research has also been done to completely automate the process of generating the qrels. In this thesis, we present different methodologies that lead to producing the qrels automatically without any human intervention. A first method is based on keyphrase (KP) extraction from documents presumed relevant; a second method uses Machine Learning classifiers, Naïve Bayes and Support Vector Machines. The experiments were conducted on the TREC-6, TREC-7 and TREC-8 test collections. The use of machine learning classifiers produced qrels resulting in information retrieval system rankings which were better correlated with those produced by TREC human assessments than any of the automatic techniques proposed in the literature. In order to produce a test collection which could discriminate between the best performing systems, an

enhancement to the machine learning technique was made that used a small number of real or actual qrels as training sets for the classifiers. These actual relevant documents were selected by Losada et al.'s (2016) pooling technique. This modification led to an improvement in the overall system rankings and enabled discrimination between the best systems with only a little human effort. We also used the bpref-10 and infAP measures for evaluating the systems and comparing between the rankings, since they are more robust in incomplete judgment environments. We applied our new techniques to the French and Finnish test collections from CLEF2003 in order to confirm their reproducibility on non-English languages, and we achieved high correlations as seen for English.

Acknowledgments

The journey I took to complete the work on this thesis was enriching and of a great value because of the knowledge and experience I gained throughout the years. I cannot but express my deepest appreciation to the people who made it possible to complete this work.

Firstly, my sincere gratitude goes to Dr. Michael Oakes, my director of studies, for his continuous support, patience, motivation and knowledge he shared. I learnt a lot from him, he made me a better researcher. I could not have made it without his guidance and help during the research and the writing of the thesis.

Secondly, I would like to express my genuine appreciation to my supervisor, Prof. Ruslan Mitkov, for his great support, for allowing me to get involved in different research activities through workshops and conferences, and for being available to answer any of my concerns.

My sincere thanks also go to Dr. Fadi Yamout, who made it possible for me to start my research, for his support and understanding over the years and for ensuring a suitable environment at the Lebanese International University to work on my research.

I would like to thank all my colleagues at the University of Wolverhampton especially Shiva Taslimipoor for the help she provided during my visits and when I was overseas.

Finally, the greatest ‘thank you’ is to my parents Yaacoub and Nina, to my person, my sister Gisèle, my best friends Katia and Loreine for their unconditional love and support during these years and for encouraging me to overcome all the obstacles.

Contents

List of Tables

List of Figures

List of Acronyms

AQML	Actual Qrels for the Machine Learning Technique
AP	Average Precision
ASS	Average System Similarity
ASSBC	Average System Similarity Based on Clustering
bpref	Binary Preference
CLEF	Cross-Language Evaluation Forum
DUC	Document Understanding Conferences
ELI	Expected Level of Importance
infAP	Inferred Average Precision
IR	Information Retrieval
ISJ	Interactive Searching and Judging
KP	Keyphrase
KEA	Keyphrase Extraction Algorithm
ML	Machine Learning technique
MM	MaxMean
MAP	Mean Average Precision
MTF	Move-to-front
NB	Naïve Bayes
NIST	National Institute for Standards and Technology
NLP	Natural Language Processing
NN	Nearest Neighbor technique
NS	Non-stationary
Qrels	Query-based relevance sets
RC	Reference Count
RAT	Relevance Assessment Tool
SE	Search Engine
SVM	Support Vector Machines
TREC	Text Retrieval Conference
VSM	Vector Space Model

Chapter 1 – Introduction

Information retrieval, as defined by Gerard Salton (1968), is "*a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information".* The retrieval of information is often an answer to a user's information need. It is all about finding the relevant material from within a large collection of stored information (Manning and Schütze, 2008). The information to be accessed can be of type text like documents, web pages, emails, books, news articles, etc. or of type multimedia such as images or videos (Baeza-Yates and Ribeiro-Neto, 2011). In this thesis, we are only interested in text retrieval.

Retrieving information happens through an information retrieval system which consists of several components that work together to answer the user's information need. The process starts when the user formulates a query that best explains his need and then submits it to the information retrieval system through a user interface. We refer to this process as "adhoc search" (Croft et al., 2009). Then, depending on the model used by the retrieval system which could be based on the vector space model (VSM) or a probabilistic model, the query will be matched against a representation of documents which have been indexed to make the search faster, and then the matching documents will be returned by order of their relevance to the

query according to the retrieval system. This ordered list of results is referred to as a ranked retrieval result.

As simple as the process sounds, many issues are encountered within the task of obtaining a relevant retrieval result. Back in the days, the amount of information to be processed and used for retrieval was very small compared to the amount of data we can find today on the web. Thus, being able to find only information relevant to the user's need among all the available data is really hard; ranking the relevant result properly is as important as finding relevant text. These two issues are related to the information retrieval model and ranking function implemented in the system. Evaluation, the focus of this thesis, is another core issue in information retrieval. Every time a new retrieval algorithm is implemented, it has to be evaluated in order to determine its efficiency and effectiveness in terms of precision and recall. Comparing between different information retrieval systems is also fundamental in the information retrieval evaluation process. For this purpose, a common evaluation framework is needed: test collections. A test collection has three main components: (1) a set of documents, (2) a set of queries formulated about different information needs and (3) a set of relevance judgments which relate a document to a query with an indication of the document's relevance. The relevance could be binary to indicate whether a document is either relevant or non-relevant to the query which it has been retrieved for. The most effort, cost and time-consuming task is the

one related to building the relevance judgments as human assessors would have to manually judge each document with respect to each query. In a large-scale environment, such as the web, it is practically impossible to judge every document, therefore a pooling technique proposed by NIST and explained in chapter 2 is applied to reduce the number of manual judgments, yet it is still an expensive task. Several studies have aimed to reduce this cost, or to build the set of relevance assessments, usually referred to as query-based relevance sets (qrels), automatically with minimal or no human intervention. Even though most of them succeeded in producing these qrels and in using them to rank retrieval systems, they could not achieve the same rankings produced by the human judged qrels, and this is why it is still an active field of research. The automatic methods we propose in this thesis were able to outperform all previous studies that do not use any human intervention. With the enhancement we introduced to one of the approaches of using a few actual human-judged qrels, we were able to partially overcome the limitation that automatic techniques have so far, which is not being able to discriminate between the best systems.

1.1 Challenges

Test collections are considered the core of information retrieval evaluation and the main framework that allows evaluating a single retrieval system or comparing between several ones. Thus, there is a need to build test

collections for the different tasks performed in information retrieval such as the filtering track which finds relevant documents according to a set of user profiles, question answering, and web search. Building a test collection is an expensive task, and the most challenging part of it is forming the relevance assessments since they require human assessors to judge the documents manually to determine their relevance to the topic in question. This could be done for small test collections, however, to simulate a real environment such as the web, large-scale test collections with millions of web pages are required to properly evaluate the systems. It becomes infeasible for human judges to perform assessments for all the retrieved pages. Therefore, it is a real challenge to build test collections that are large and reliable enough to be used for evaluation.

Automatic techniques developed to produce qrels for a test collection without human intervention cannot really evaluate a system in terms of the recall metric, which involves measuring how many relevant documents a system was able to retrieve, since there is no prior knowledge of the documents' relevance. This is why, the evaluation in the absence of judgments is usually approached as a ranking problem in which a set of retrieval systems are ranked according to a performance measure – it could be the mean average precision, binary preference or inferred average precision, or other measures as described in chapter 3. The metric value is computed twice, a first time using the real or actual qrels which were assessed by human judges, and a

second time using the automatically produced qrels, which are referred to as pseudo-qrels. The correlation between the two rankings obtained is an indicator of the quality of the produced pseudo-qrels. The correlations can be quantified using Kendall's tau and Spearman's rho as described in chapter 3. A perfect match between the two rankings is expressed by a correlation value equal to 1. No previous techniques have been able to achieve a perfect correlation using TREC data.

For fully automatic techniques that do not involve any human intervention, a first challenge that was drawn from Soboroff et al. (2001) is that we cannot assume any prior knowledge about the test collection or the distribution of the relevant documents in the pool, their mean or the standard deviation. Another challenge which is related to the techniques that consider the reference count of the documents in the pool, that is the number of runs that retrieved each document, to choose those which can be considered relevant to form the pseudo-qrels (Wu and Crestani, 2003; Rajagopal et al., 2014) is the problem of the "tyranny of the mass" described by Aslam and Savell (2003) where popularity takes over performance. Here the assumption that the documents retrieved by most runs are likely to be relevant is not necessarily true. The third and most important challenge of all these automatic methods is their inability to discriminate between the best performing systems using the pseudo-qrels. As for the approaches that require some limited human intervention to form the qrels, the challenge

remains with the pooling technique used to determine which documents to select and judge (Cormack et al., 1998; Carterette et al., 2006; Efron, 2009; Sakai and Lin, 2010; Pavlu et al., 2012; Mollà et al., 2013).

1.2 Research Questions

The main research question of this thesis is related to the evaluation framework used in information retrieval, the test collections, and aims to reduce the human effort required to build the set of qrels for a TREC-like test collection or even to automatically produce it. Since previous work showed that it was possible to automatically generate the set of qrels, the purpose of the work done was to propose new techniques that could provide better results than the ones described in the literature.

In his work, Efron (2009) showed that it is possible to build a set of qrels automatically by simply expressing an information need using several aspects or queries which means formulating several queries to address the same topic. Pooling the top 100 documents from the retrieval results of all the queries related to the same topic resulted in a set of qrels which produce rankings which are highly correlated with those produced by human-built qrels. Although forming the qrels was done automatically, human effort was still required to develop the aspects describing a topic, of the information need. This raised the first research question:

Q1. Can we use keyphrases describing a topic as queries to retrieve more qrels?

The first research question was satisfied by using keyphrase extraction from the documents based on the hypothesis that a relevant document contains relevant information and relevant information could lead to more relevant documents. Therefore, automatically extracted keyphrases from the documents were considered as aspects of the information need and they were used as substitutes for the queries formulated by Efron's technique. The details of the methodology, the tools used, and the experimental design for this set of experiments are described in section 3.4.

The pseudo-qrels produced using the KP extraction technique described in this thesis resulted in positive correlations between the overall TREC automatic runs' rankings and the rankings obtained from using the KP pseudo-qrels, but the correlations Efron obtained from using the aspects derived from topics were still better. Furthermore, we were unable to standardize the parameters over several test collections. The steps we followed to answer this research question with the evaluation results obtained are provided in Chapter 4.

We wanted to look for a more solid technique which could be applied to any test collection. The technique proposed by Rajagopal et al. (2014) assigns relevance to the documents automatically based on the number ("count") of

systems that retrieved them. Two different cutoff percentages (50% and 35%) based on the reference count of each document were chosen. The documents which had a count greater than the selected cutoff percentage were considered relevant while the remaining documents were considered non-relevant. Both these cutoff percentages led to a high false positive rate and the resulting qrels did not allow discrimination between the best systems. On the contrary, the correlations were negative. Based on the hypothesis which considers that a document retrieved by several systems has a high probability of being relevant to the topic submitted, we thought we could use a higher cutoff than the one used by Rajagopal et al. and therefore form a set with a high probability of being relevant. We used this set to find more relevant documents based on a distance metric between the documents' vectors. The closest matching documents in the pool to those already found relevant were added to the set of relevant documents. This initiated the second research question:

Q2. Is it possible to use machine learning techniques to expand an initial set of presumed relevant documents and produce more qrels?

Section 3.5 details the process followed to generate the set of qrels and describes the tools used to implement this approach. The experiments tested the use of the nearest neighbour approach, the unsupervised K-Means algorithm and the supervised machine learning using the Naïve Bayes

classifier and Support Vector Machines. Chapter 5 describes the work done to answer the second research question. Research question Q3 focuses on answering how to build the pseudo-qrels fully automatically. Even though the KP technique proposed to answer Q1 provided positive correlations between the overall system rankings, and the machine learning (ML) technique which aimed to answer Q2 outperformed all the automatic techniques in the literature, they both failed to discriminate between the best performing TREC system runs and this led to the third research question:

Q3. If it is not possible to form the qrels fully automatically, how many human-judged qrels should be supplied to start the process?

Chapter 6 answers this question by applying a variation on the ML technique proposed in chapter 5 which we called "Actual Qrels for the Machine Learning" (AQML) technique since it uses a small number of actual or known relevant documents and expands them using supervised machine learning classifiers. In the literature, the traditional measure used to evaluate a system is the Mean Average Precision (MAP) defined in section 3.6.1. A set of systems can be ranked according to their MAP scores, so they can be compared. However, when we have a set of incomplete judgments, more robust measures are used; the binary preference (bpref) described in section 3.6.2 and the inferred average precision (infAP) in section 3.6.3. Using these measures, we can ask the research question below:

Q4. Do bpref and infAP give more accurate system rankings than MAP when we have an incomplete set of judgments?

The evaluation conducted in chapter 6, section 6.5 is an attempt to answer the research question above. All the experiments were conducted on TREC English test collections. In order to evaluate the ability of the proposed ML and AQML techniques to work with any language, we also tried to answer the fifth research question in chapter 6, using the CLEF French and Finnish test collections described in section 3.3.2:

Q5. How well do the techniques developed in this thesis work for languages other than English such as French and Finnish?

Since evaluating the effectiveness of an information retrieval system is as important as comparing between different systems, measuring the system recall is an indication of how well this system performs, or how well it can retrieve relevant documents especially since the best systems usually succeed in finding rare relevant documents. Therefore, the work presented in this research attempts to answer whether the automatically produced qrels can be used in situations where high recall is required. The answer comes as a synthesis of all the research work done in this thesis and will be discussed in the conclusion chapter since the matter of high recall seems to be a limitation of the automatic techniques as they do not make use of actual relevant documents judged by human assessors.

1.3 Contributions

The main contributions of this thesis can be summarized are as follows:

- **C1:** We were able to outperform all previous fully automatic techniques that rank retrieval runs without relevance assessments.
- **C2:** The fully automatic technique we developed works even better when using a few actual known relevant documents.

The contributions in individual chapters are listed below:

C3: In chapter 4, we show that it is possible to use aspects or keyphrases describing a topic to produce pseudo-qrels without any human intervention. The correlations between the rankings produced from using the pseudo-qrels and the gold standard rankings based on human-qrels were significant.

C4: In chapter 5, we demonstrate that machine learning algorithms like the nearest neighbour, Naïve Bayes classifier and Support Vector machines can be used to automatically produce a set of qrels that rank TREC runs in a way which is highly correlated with the rankings produced by human-built qrels.

C5: In chapter 6, we show that a variation on the machine learning technique proposed in chapter 5, using a small number of human-judged qrels as the training set, was needed to overcome the limitation of discriminating between the best performing retrieval systems.

C6: The final contribution we present in chapter 6 is to show that the techniques developed for English test collections also work with non-English test collections, in particular French and Finnish.

1.4 Origin of the material

The chapters described in this thesis are based on the following publications:

- Chapter 4: the KP technique used to automatically build the set of pseudo-qrels for a TREC test collection is based on the work undertaken in Makary et al. (2016a): *Using key phrases as new queries in building relevance judgments automatically, LWDA Conference, 2016, p. 175-176*
- Chapter 5: the nearest neighbour technique experiments with the evaluation results inspired by previous work completed by Rajagopal et al. (2014) were reported in Makary et al. (2016b): *Towards automatic generation of relevance judgments for a test collection*, *2016 Eleventh International Conference on Digital Information Management (ICDIM), DOI: 10.1109/ICDIM.2016.7829763.Publisher: IEEE*. As for the work that involves using supervised machine learning to produce pseudo-qrels, it has been published in Makary et al. (2017): *"Using Supervised Machine Learning to Automatically Build Relevance Judgments for a Test Collection," 2017 28th International Workshop on Database and Expert Systems Applications (DEXA), 2017, pages 108-112, DOI: 10.1109/DEXA.2017.38.*

1.5 Thesis Outline

The thesis is organised as follows.

- Chapter 2. This chapter is a literature review of previous work done in attempts to automatically produce the pseudo-qrels of a test collection or to rank different retrieval systems in absence of relevance assessments or with an incomplete set. It also shows the different types of relevance that exist in information retrieval evaluation, in addition to some interfaces that help human assessors to evaluate a retrieval result manually. A section related to inter-assessor disagreement is also present in the chapter, showing that inter-assessor disagreement does not have high impact on ranking retrieval runs.
- Chapter 3. The methodology chapter describes the TREC and CLEF test collections we used to run our experiments. It is divided into three major sections, each describing one of the three techniques we propose in this thesis to produce pseudo-qrels. Further to explaining how each technique works, we present technical details about the tools, packages, libraries and classes we used to implement those methods.
- Chapter 4. This chapter provides the details about the technique based on using keyphrases extracted from documents to build the pseudo-qrels. The steps followed in the experiments, the correlation results and the intrinsic evaluation performed are also provided in this chapter.
- Chapter 5 describes the use of supervised machine learning algorithms: the nearest neighbour, the Naïve Bayes classifier and Support Vector machines with two different approaches to build the pseudo-qrels for TREC

test collections, the experiments conducted, and the evaluation of the different techniques are presented in this chapter. We conclude with the limitation of this technique that we address in the next chapter.

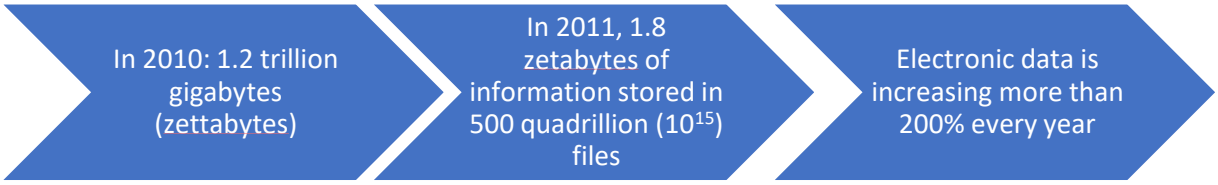
- Chapter 6. This chapter introduces an enhancement to the technique based on supervised machine learning which deploys a few actual human-judged qrels which are selected based on the Losada et al. (2016) pooling technique. The motivation for using the Losada pooling technique, the evaluation of the pseudo-qrels and the results obtained are described in this chapter, in addition to the experiments we ran on the CLEF French and Finnish test collections.
- Chapter 7. This chapter summarizes all the findings of the thesis, revisits the research questions and to what extent we were able to answer them. It includes the conclusions drawn from each of the chapters and a future direction for the work.

Chapter 2 – Literature Review

2.1 Introduction

At one time, retrieving information was an activity that engaged only a few people such as librarians, paralegals and professional searchers. However, now that the internet technology has evolved, millions of people deal with daily information retrieval tasks using web search engines. The pace at which we are exploring, investigating and learning, is increasing tremendously. Our information is stored and managed electronically.

According to a recent IDC Digital Universe Study Sponsored by EMC (Gantz and Reinsel 2012),



Even though this data is accessible, it can be used only if it can be transformed into information in a reasonable time. This is the subject of study in Information Retrieval (IR). Users with a certain information need formulate a query and submit it to a search engine like Google, Yahoo, or Bing, and expect to get in return a list of documents or pages which are relevant to the query they submitted. The information retrieved can be

either media (audio, video) or in the form of text. So, text retrieval is only one field in IR.

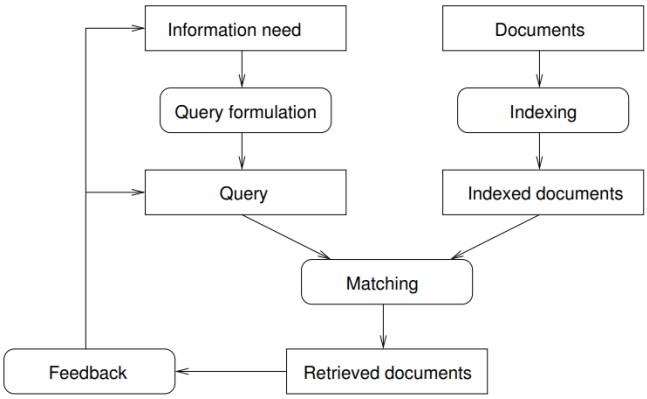


Figure 2. 1*: Information Retrieval Process (Hiemstra, 2009)*

2.1.1 Information Retrieval Process

The information retrieval process starts when a user has an information need - defined by Croft et al. (2009) as "the underlying cause of the query that a person submits to the search engine, or the motivation for using a search engine". He then formulates it into a query, submits it to a retrieval system (a search engine) and waits for the results. The retrieval system, in response to a query, produces a list of documents that are believed (or known) to be useful o the user. This list of documents is referred to as a ranked list: a document at rank i is considered at least as useful as the document at rank i+1**.** The diagram below describes the complete process. The user with a certain information need formulates a query which consists of several words or terms. He then submits the query to an information retrieval system or search engine that implements a retrieval function. The available documents are usually indexed to make the search process faster.

The retrieval function finds a match between a document and the query submitted. In order to find a match, the documents can be represented as terms which are usually single terms or multiword units such as "White House", or they can be represented using the fundamental words: (1) a bag-of-words that ignores the words' order or (2) word stems leaving only the radix of the word, suppressing the suffixes. In this case, the document is represented as a vector of words. The matching process can be based on the Vector Space Model (VSM), where both the document and query are represented as vectors. The binary representation of the document indicates the presence (1) or the absence (0) of the query term in the document as shown in figure 2. 2. For example, the value "1" in row D1 and column t1 shows that term t1 is present in document D1 while t2 is not found in D1 since it the matrix shows a 0 value.

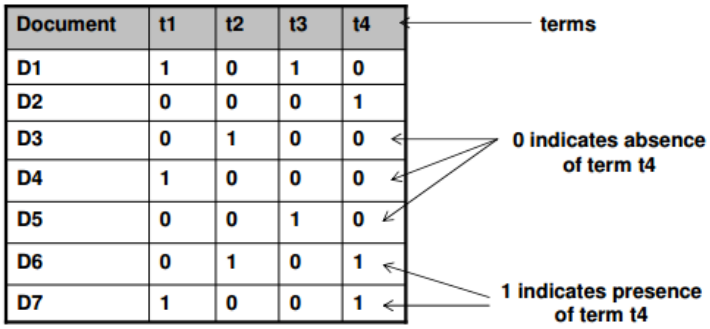


Figure 2. 2: a vector representation of a document indicating whether a term is present or absent in the document.

The most similar documents to the query are identified using a distance measure or proximity measure between the document vector and the query vector.

Document	t1	t2	t3	t4	PM
D1	1	0	1	0	2/4
D2	0	0	0	1	1/4
D3	0	1	0	0	0/4
D4	1	0	0	0	1/4
D5	0	0	1	0	1/4
D6	0	1	0	1	1/4
D7	1	0	0	1	2/4

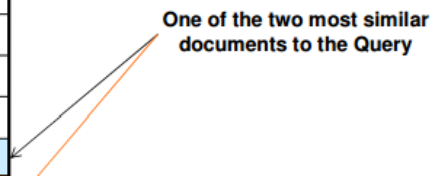


Figure 2. 3: matrix that shows the term frequency in each document.

Another representation for the documents is the tf.idf measure (Salton, Wong and Yang, 1975). tf stands for term frequency in a document: the number of occurrences of each term in the document. idf stands for inverse document term frequency: it is the number of documents containing a particular term. One of the measures that can be used to compare the document vector to the query vector is the cosine similarity measure (Baeza-Yates and Ribeiro-Neto, 2011) and the best match is retrieved using the simplest measure and the number of matching terms. The cosine similarity can be defined as in equation (2.1):

$$Cosine\_similarity\left(d_j, q\right) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \qquad (2.1)$$

Where $w_{i,j}$ is the weight assigned to term i in document j and $w_{i,q}$ is the weight assigned to the term i in the query. A value of 1 for the cosine similarity indicates a perfect match.

Looking at Figure 2.3, we can say that documents D1 and D7 are one of the best matches for the query since they contain 2 terms out of 4. When a first list of matching documents is retrieved, the user checks the content of the

documents and if he finds that the obtained results do not satisfy his need, he can reformulate the query, expressing his need using other terms which provide a better representation of the information required. Alternatively, he can select some of the relevant documents retrieved to use in the relevance feedback process, which means that the content of the relevant documents will be used to expand the initial query submitted in order to describe the information he is looking for in a better way. Not only is retrieving relevant documents a main concern for the user but also, the order in which the returned documents are displayed. The user is generally satisfied if the information needed is ranked at the top of the list returned. Therefore, IR systems should be continuously improved. As a first step, it is necessary to evaluate the quality of the search engine: we can measure how many relevant documents it returns and their position or rank in the retrieved list. To achieve this, we need to know the relevance for every single document in the ranked list, also referred to as relevance assessment, so one of the major issues in IR is the concept of relevance. Is a document relevant, strongly relevant or non-relevant to a particular query? As a simple definition, a relevant document contains the information that a user was looking for when he submitted a query (Croft et al. 2009). To evaluate how good a retrieval system is, we need to quantify the results obtained. This is why we usually use a binary form of relevance: relevant (1) or non-relevant (0). The process of determining the relevance of each document by an

assessor is practically infeasible, costs a lot of money and requires a lot of time. Another question comes to mind: given several retrieval systems, how do we decide which one is better? How can we compare systems if different documents, queries and evaluation metrics are used? A standard evaluation framework is required: we use what we call "test collections". We use these to compute metrics like precision and recall.

2.1.2 Scope of the Literation Review

A deep study of the concept and types of relevance is presented in section 2 and the judging and search interfaces that are currently deployed are discussed in section 3. Since human assessors usually disagree in judging a document, we go over some studies and experiments that were related to assessor disagreement and errors in assessment in sections 4 and 5. An overview of all the work done in the IR field to reduce the human effort in building the relevance judgment lists is presented in section 6 and in the last section we draw a conclusion outlining the work done in this thesis.

2.2 Relevance: Types and Criteria

2.2.1 Types of relevance

Determining relevance depends on several factors such as topic, novelty, style, or context. If the user is not familiar with the topic to which the information is related, the process of relevance judgment can be difficult.

A text document is topically relevant to a query if it covers the same topic. For example, news about a storm in Lebanon is topically relevant to the query "severe weather events". A user living in another country might not consider the story relevant, so it is not user relevant. Thus, the user relevance includes more features which will influence the decision made regarding a document: it might be an old story or written in a different language than expected by the user. We need to make sure that the search system provides satisfying answers to the user's information need. However, when evaluating systems in laboratory studies, we usually use test collections and the relevance assessment is somehow artificial. Topicality in such environments is easy to define because it can be connected to the texts of the documents and their representation. Novelty can be seen as content novelty or source novelty as per Barry (1994). The content novelty determines the extent to which the information presented to the user and provided by the document is novel. For example, the user might answer: "I have never heard of it", "I read this information many times earlier". As for the source novelty, it measures to what extent the source of information: authors, journals, publications, etc. are known to the user.

Saracevic (1996) defined different manifestations of relevance:

- The System/Algorithmic relevance
- The Topical relevance

- The Cognitive relevance or Pertinence
- The Motivational/Affective relevance
- The Situational relevance or Utility

The System/Algorithmic relevance includes the similarity between the user's query and the document retrieved determined by the system's matching criterion, e.g. the query-document word overlap. It compares how close the document is to the query. The Topical relevance is about the semantic fit between the document and the query. Determining whether the document is topically relevant to the query depends on the content of the document, whether it discusses the same topic or not. The Cognitive relevance or Pertinence is manifested by the user's knowledge level, the novelty of the information presented to him and its quality. The Motivational/Affective relevance is similar to the use orientation described by Cuadra (1968), i.e. how the user intends to use the information he seeks. The Situational relevance or Utility is related to the situation or the task that should be completed by the user. It also refers to the way the documents retrieved help the user achieve the task he has in hand.

2.2.2 Criteria influencing relevance

Studies like the ones conducted by Schamber (1991) and Barry (1994) showed that the same set of features affect the relevance judgment process between different users, like the knowledge level, cognitive state,

perceptions of the user, in addition to accuracy, recency and clarity, with situational factors such as time constraints and the effort and cost required to obtain the information (Barry, 1994).

When the user is an expert in his domain or has sufficient information about the topic he is looking for, he will be able to judge more advanced information available in a document than a user who has heard about the topic for the first time while judging the documents. The user's ability to understand the information presented to him also affects the judgment process. The user assessment is affected by the accuracy of the information presented to him. The user usually looks for accurate, correct and valid information. In addition to correctness, how up-to-date the information is will also affect the relevance judgment process. This notion is defined as "recency" (Barry, 1994). As long as the information in a document is presented in a clear and readable manner, the user will be able to judge the document. When offered several documents to judge within a short period of time, the user will not spend a lot of time to judge a particular document if it looks hard for him to judge or he is even not very interested in its content. He would rather go over all the other documents to finish the judging task assigned to him.

Cuadra (1968) identified 38 variables that affect judgments of relevance. These variables were grouped into five categories: (1) variables related to the Document, (2) variables related to Judgment Conditions, (3) variables

related to the Information Requirement Statement, (4) variables related to the Judge and (5) variables related to the Available Mode of Expression. The project took almost two years to study only half of these variables. In the fifteen studies conducted, more than 500 subjects were used as relevance judges. These judges were librarians, information specialists, faculty staff and students in psychology and in library science. During these experiments, several variables were shown to be related to each other. Any relevance judging situation must have a set of documents, a particular information requirement statement, particular judges, judgment conditions and mode of expressing the relevance judgments. When any of these factors change, the remaining elements will also be affected. The "use orientation" variable (Cuadra, 1968) for instance affects the judgment result. The use orientation means how the user is going to use the information he is looking for: for example, just for reference purposes or he needs to learn more about the topic. The relevance judgment was shown to vary depending on the use orientation.

The concept of relevance has been represented differently by different researchers. Saracevic (1976) explained relevance using a communication framework. In any communication, there is a message exchanged between a source and a destination. He defined relevance as a relation between the source's subject knowledge, the subject literature which means the information provided about it in the past and the system files (the set of

documents used by the system) and the destination's user cognitive state, use orientation, and the context of the communication. He named the relation between these elements "views of relevance".

Schamber (1991) conducted a study in which she asked occupational users of weather-related information in a multimedia environment to judge a list of results. She came up with 22 detailed categories for the answers yielded by the respondent. The same was done by Barry (1994), but the respondents were faculty members and students who examined printed and textual information. He showed that users usually go beyond topical appropriateness in judging documents. From the experiments conducted, 23 criteria of relevance shared among users were mentioned by the respondents. Many of the criteria found by Schamber were also listed by Barry (1994). The table below summarizes these criteria and shows a grouping for them:

Content of the documents	Depth/scope Objective accuracy/validity Tangibility Effectiveness Clarity Recency
User's previous experience and background	Background/experience Ability to understand Content novelty Source novelty Stimulus document novelty
User's beliefs and preferences	Subjective accuracy/validity Affectiveness
Sources within the information environment	Consensus within the field External verification Availability/environment Personal availability

Sources of documents	Source quality Source reputation/visibility
Documents as physical entity	Obtainability Cost
Criteria related to the user'situation	Time constraints Relationship with author

Table 2. 1: Categories of relevance criteria presented in groups.

After interviewing nine respondents, no new categories were added which led the authors to believe that these categories were consistent among different users in the judgment process. In addition to the accuracy, recency and clarity criteria which were discussed earlier, the quality of the sources is another important factor, because reputable, well trusted and expert sources increase the level of confidence of the user in the information provided. Making the information accessible should also be taken into consideration. One of the criteria that were only listed in the Barry study is the effectiveness of the method or procedure used to answer the user's information need. In his article, Saracevic (1996) defines attributes of relevance or manifestations of relevance which express the relation between the user, query and document.

2.2.3 Graded relevance

Relevance was also defined as being multidimensional. In his summary of the history of relevance, Mizaro (1997) described four aspects of relevance. The first aspect is the document or the information, the second is the query,

information need, the third is the task, topic or context and the fourth is the time variable, since the relevance might change with time because the user gained more experience or additional knowledge in the field of interest, and therefore some information has become relevant even though it was not before.

To a user, even if a document is considered to be relevant, there are degrees of this relevance. Thus, we can divide relevance into several grades, and we call it graded relevance. The problem is determining how many grades of relevance are considered sufficient. An example of graded relevance that uses three categories might be to judge a document as relevant, highly relevant or non-relevant. There is no fixed answer to this question. The user's definition of relevance and all preferences that affect his judgment are always a main concern. The use of binary relevance (relevant (1) and non-relevant (0)) is prevalent in evaluating and comparing IR systems (Kekäläinen, 2005). The effect of the graded and binary relevance on measuring different IR systems' performance and the impact on ranking these systems was studied by Lesk and Salton (1968), Burgin (1992), Voorhees (2001) and Kekäläinen (2005). Lesk et al. reported stability in the system rankings despite the differences between the user assessments. This was mainly due to three reasons: the evaluation results are the averages over many topics; disagreement does not affect documents that are highly relevant or non-relevant; the documents which are judged differently are

ranked after the documents unanimously agreed on. The same outcome was obtained by Burgin. The experiments done by Voorhees (2001) used the TREC test collections. The relative effectiveness of different retrieval strategies was constant despite marked differences in the relevance judgments used to define perfect retrieval. Another variation in the experiment was that the documents to judge were presented to the authors of the query submitted or the ones who formulated it as opposed to assessors that only viewed or used the query (non-author judges). An additional experiment was completed to compare whether the system performance will be affected by having a single judge to assess a document or allowing several judges to assess the same document. The impact of the environment on judgments was among the conditions included in the study. The overall performance of retrieval runs was very similar. In a later study of Kekäläinen (2005), assessments from TREC documents were done on a four-point scale: (0) irrelevant, (1) marginally relevant document: the document points to the topic but does not contain any further information, (2) fairly relevant document: the document gives additional information about the topic but the presentation is not enough and (3) highly relevant document. The experimental results showed that the correlation between different IR systems diminishes when fairly and highly relevant documents are given more weight, which means that the system ranking changes. The highly relevant documents are usually easier to find by the retrieval systems

than fairly or marginally relevant documents and usually the best performing systems are the ones able to find more relevant documents than other and particularly find rare relevant documents which are hard to find by others. This is why when not all relevant documents are treated equally, and fairly relevant documents are given more weight, this could result in a better discriminating power between the retrieval systems. There have been few studies discussing graded relevance and its use in information retrieval evaluation and to our knowledge no work has been done to automate the process of generating graded qrels for a test collection. This could be due to the fact that for adhoc retrieval tasks we are more concerned in retrieving relevant documents without emphasising how relevant a document is to the submitted query, especially since the number grades that could be considered sufficient for evaluation cannot be standardised.

In this thesis, we are concerned more with the variables related to documents and judging conditions since it is possible to come up with an automated process that deals with subject matter, amount of information, time for judging, the size of the document sets, breadth of document sets and others. However, the aim is to devise a methodology to automatically assign binary relevance to documents retrieved by an IR system in an attempt to form the set of qrels automatically without involving human assessment in order to reduce the effort and cost spent to create test collections and use them in system evaluation.

2.3 Information retrieval judging interfaces

The process of collecting relevance assessments is difficult to achieve and it consumes a lot of time. In some cases, quality control mechanisms are used to detect spammers who trick and complete the tasks inaccurately especially in crowdsourcing, or poor work done in the judgment process or even because experts can make mistakes (Alonso, 2013). This is why it important to have a tool that captures a copy of what aspects of each judged document the assessor saw as relevant in addition to collecting the assessment for a batch of documents at once. The visualization and judging interfaces developed to date are few in number. Most of them share common features like highlighting the text which looks relevant to the topic and logging the user judgment.

2.3.1 Haines and Thistlewaite

Haines and Thistlewaite (1996) developed a Relevance Assessment Tool (RAT.) It was used to assess documents for large TREC web collections and it was used in the experiments conducted to measure the quality of different search engines (Hawking et al., 2000). A snapshot of the tool is shown below.

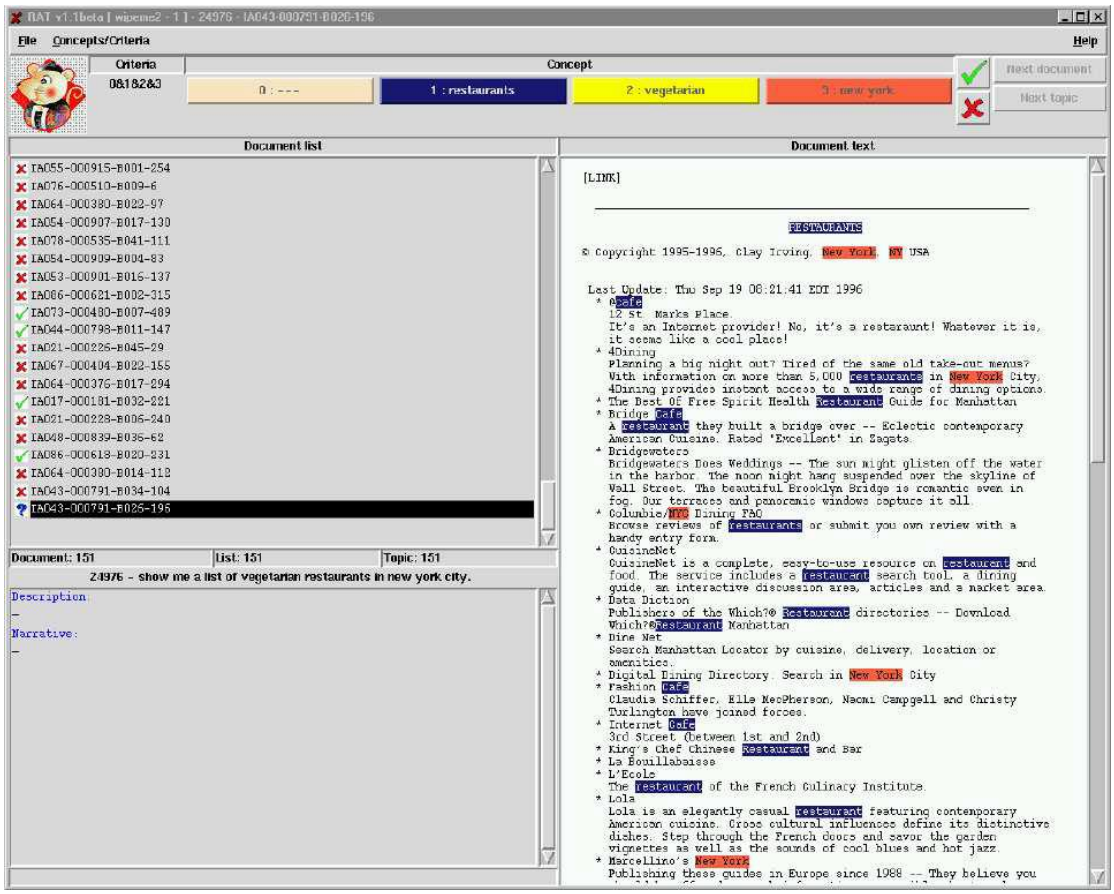


Figure 2. 4: RAT in action

When starting the judgment process, each judge has to enter concepts that can be used in determining the relevance and criteria for relevance. So, in the above figure, the concepts of the query submitted were "vegetarian", "restaurants" and "New York", and the criterion for relevance was a conjunction of all the concepts. In the left window, the list of document IDs are displayed and when the judge clicks on a document, the content of the document is displayed in the right window and the concepts which were entered initially are highlighted in different colours ("restaurant" in blue, "vegetarian" in yellow, etc.) in order to help the judges to see the relevant parts related to the query they submitted and then they can decide if a

document is relevant by clicking on the (✓) button or non-relevant by clicking on the (×) button.

2.3.2 Clarke's System / TREC MultiText Project

Cormack et al. (1998) used an interactive search system in the TREC MutliText project which uses manual Boolean query construction, which means that the terms in the query are either found or not found in the document. The system then ranks the documents obtained based on the length and the number of passages that satisfy the query. Each passage is given a score based on its content and then a document is assigned a score depending on the passages it contains (Clarke et al., 1998). The search terms found in each paragraph are highlighted and the system allows the assessor to record their judgments.

2.3.3 Carterette et al.'s System

Carterette et al. (2008) designed an interface that allows assessors to use both binary and graded relevance in order to prove their hypothesis: if a document A is more relevant than document B, it will be easier for the assessor to judge document A as relevant and it will take less time than judging document B. They designed three different interfaces for three different types of judgments: (1) absolute judgments using a five-point scale: Bad, Fair, Good, Excellent and Perfect, (2) binary relevance judgements: relevant or irrelevant and (3) preference judgments in which the assessor can say he "definitely" prefers one page or document over the

other. The time spent to make a judgment differed between assessors. The variables that affect time are the time spent reading a document or a page and the time spent to make correct judgment. Table 2.2 below shows the median number of seconds spent by different assessors for each interface. Absolute judgments took twice as long as the preference judgments because it can be harder to make an absolute judgment rather than making a preference judgment.

	Preference	**Definite**	**Absolute**	**Overall**
Assessor 1	3.50	3.41	7.96	**3.70**
Assessor 2	3.24	3.67	6.12	**3.55**
Assessor 3	2.35	2.82	5.56	**2.82**
Assessor 4	4.13	4.30	8.78	**4.71**
Assessor 5	2.72	3.30	8.20	**3.17**
Assessor 6	2.09	2.40	3.21	**2.31**
Overall	**2.87**	**3015**	**6.33**	**3.23**

Table 2. 2: Median seconds per judgment by each assessor in each interface (Carterette et al., 2008)

Figures 2.5 and 2.6 are screenshots of the interface designed by Carterette et al. described above:

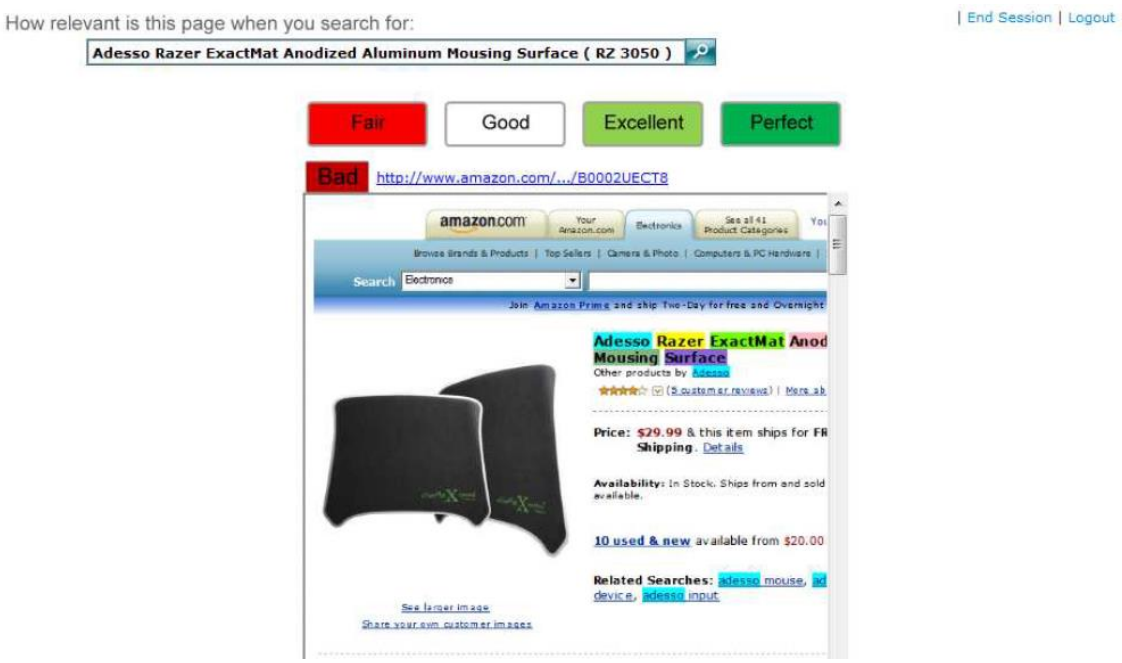


Figure 2. 5: absolute judgments interface

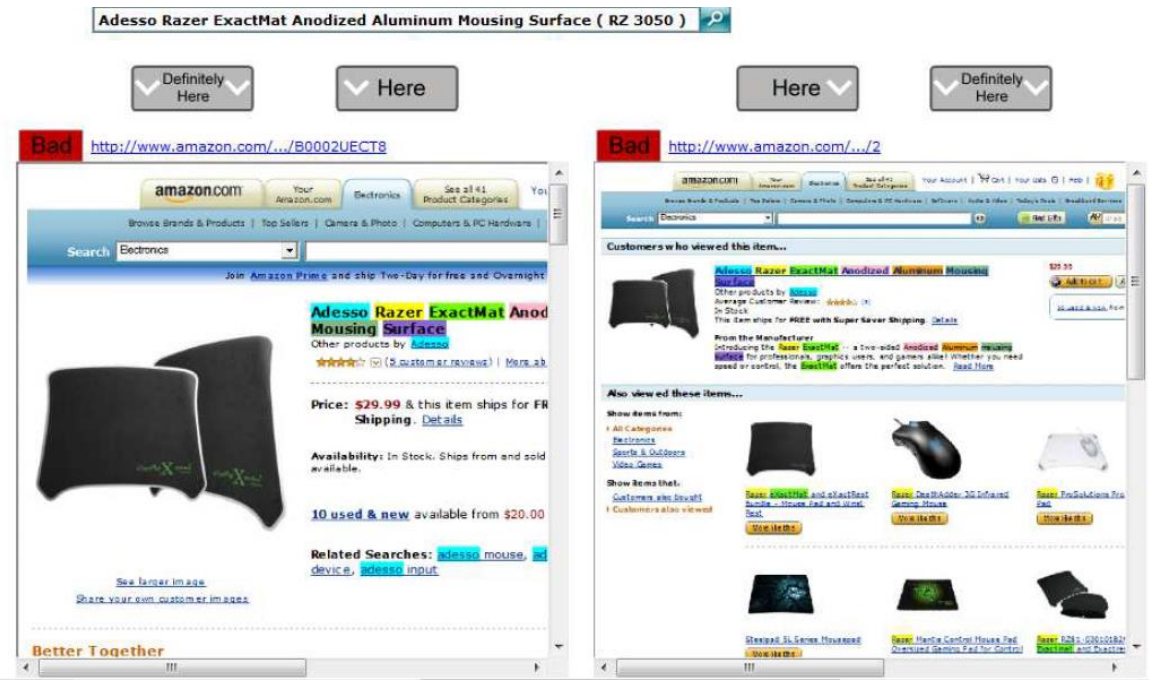


Figure 2. 6: preference judgments interface

The query submitted is shown in a text box at the top left of the interface. Each of the interfaces has buttons on the top to enter a document judgment.

In the graded relevance interface, there are four buttons for each scale defined earlier. In the preference judgment interface, the user can either choose that the relevant information is “Here” or “Definitely Here” in the document presented to him for judgment. A “Bad” button is also shown in the interfaces. This button is used to judge the page or document as non-relevant or in case the web page was spam. A web spam page, also referred to as search spam page is a web page that has content created to improve search rankings without having any value for the user.

2.3.4 Waterloo System

In another study by Cormack and Mojdeh (2009), they classified every document in the TREC 2009 exercise using machine learning methods with no tokenization or parsing of tags. They also created a custom HTML interface for the judging process (Figure 2.7). The user selects which topic is to be used for the query submission and he can also choose which documents to view, the ones that were judged relevant, non-relevant or those which are still unjudged. After clicking on the button “click” the document will be shown below the query and the user can judge whether the document is relevant, non-relevant or flag for later reassessment.

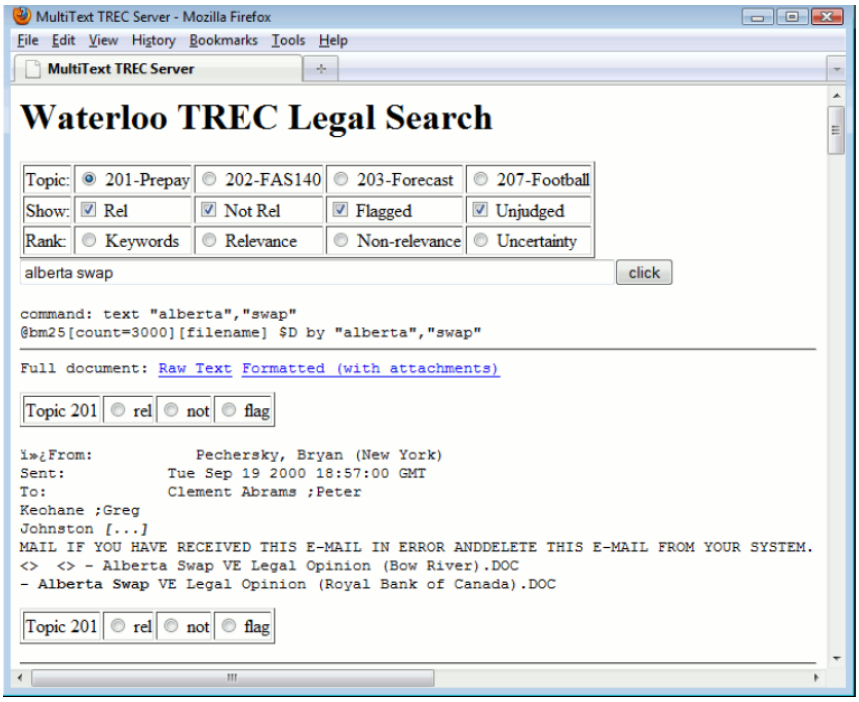


Figure 2. 7: Interactive search and judging interface

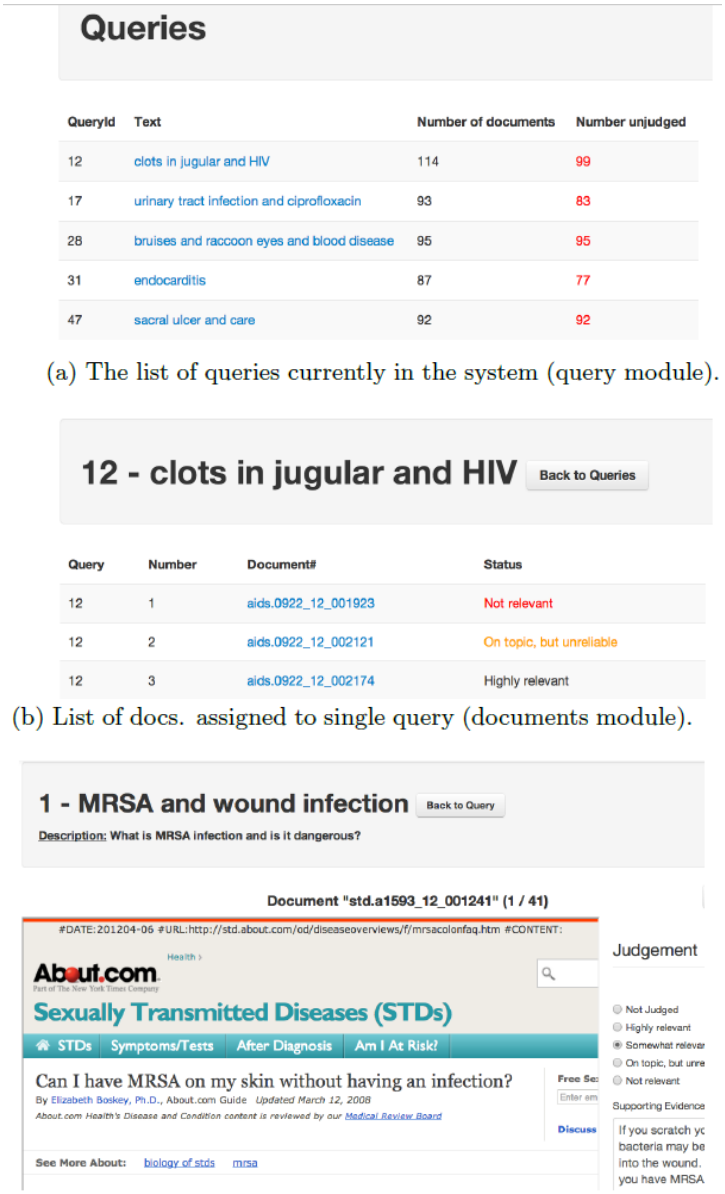
2.3.5 Lewandowski and Sünkler System

Another tool to help users assess documents was developed by Lewandowski and Sünkler. (2013): Relevance Assessment Tool (RAT). It allows researchers to evaluate their search systems, design tests and collect judgments. It is web-based however, not an open source tool.

A recent open source system for information retrieval relevance assessment (Relevation![1]) was developed by Koopman and Zuccon (2014). It is a web-based system that allows assessors to perform their relevance judgment tasks, even remotely. The assessors upload the documents and queries and then assign a label for relevance to each document. The system was used in

[1] http://ielab.github.io/relevation.

the TREC Medical Records track and the CLEF eHealth evaluation. The system consists of a query module that provides the list of queries uploaded to the system, the number of documents assigned to the query and the number of unjudged documents as shown in Figure 2.8 (a). The document module of Relevation! displays the text document assigned to the selected query with a label to assign relevance in the Status column. In Figure 2.8 (b), the query 12 has 3 documents where the first was judged as non-relevant, the second considered topically relevant and the third highly relevant. The judgment module in part (c) of Figure 2.8 allows the judges to enter their assessment and to select part of the text as supporting evidence for the document relevance. The relevance records can also be exported. These features were not implemented in previous assessment systems like RAT and Hawking's tools.



(a) The list of queries currently in the system (query module).

(b) List of docs. assigned to single query (documents module).

(c) Assessing a document (judgements module).

Figure 2. 8: Relevation! Screenshot

All the assessment interfaces described previously require human intervention. The assessor has to log or record his judgment for the documents assigned to him which requires time. The assessments have to be then collected from all judges and reviewed in order to build the final set of qrels.

2.3.6 Crowdsourcing

Because several tasks in information retrieval are very expensive, especially the ones related to relevance evaluation and manual document annotation, the crowdsourcing paradigm has been introduced to deal with such tasks. The one relevant to the work done in this thesis is the relevance evaluation. Crowdsourcing benefits from the online human resources available from different countries and who are nowadays available due to the growth of the internet and bandwidth connectivity, and it was seen as a solution to the scalability problem for the web test collections. Alonso et al. (2008) evaluated the use of Amazon Mechanical Turk[2] (MTurk) (Turk, 2012) to assess documents. MTurk is a crowdsourcing service provided by Amazon Web services platform and has an artificial intelligence component deployed to complete a certain task. Workers can register from any country and then different tasks can be assigned, such as labelling images, assessing documents, or annotating them according to the requester's need which is usually that of an individual or an organization that requires some tasks to be completed. Figure 2.9 below shows the interface of the task designed by Alonso et al. and offered by MTurk to a worker in order to assess a text retrieved for a given query.

[2] http://www.mturk.com

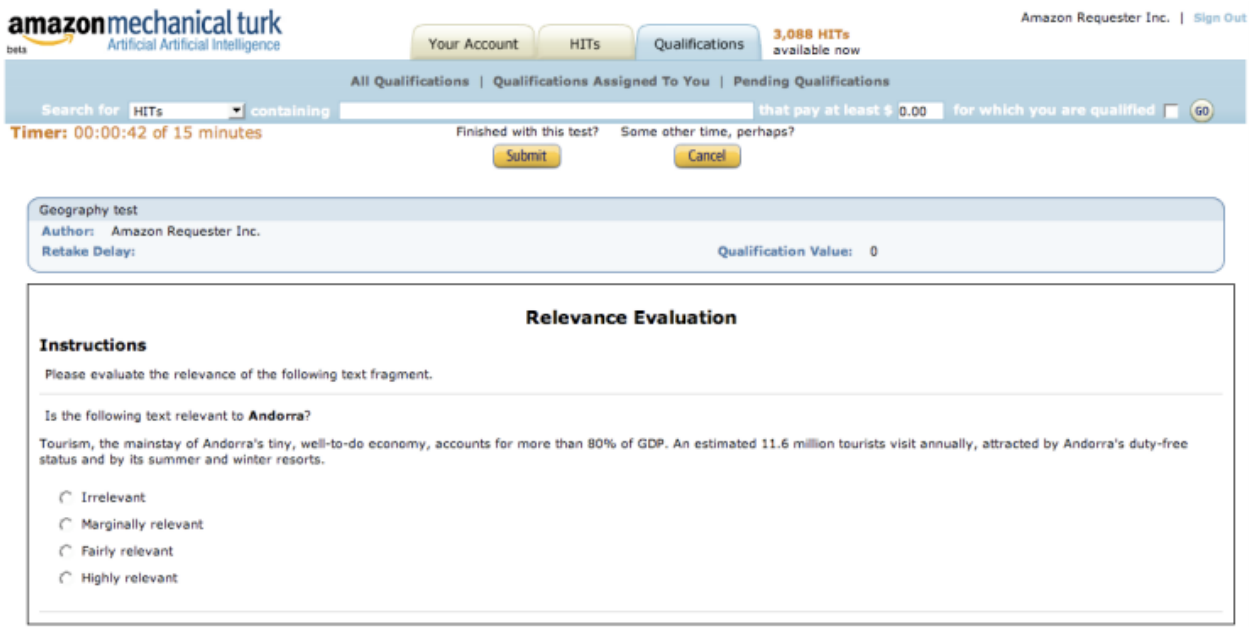


Figure 2. 9: *Evaluation task in MTurk (Alonso et al., 2008)*

The task is designed to be a four-scale relevance assignment. The worker will have to login and once they see a sample of the task assigned, and they can either accept it or reject it. The requester will receive a daily report of the number of tasks completed. Some of the concerns raised in crowdsourcing are related to quality control. The skills and knowledge level of the workers cannot be guaranteed even though the MTurk ensures that workers answer some questions and provide details or examples that support their level of knowledge in a particular topic. The quality of crowdsourcing evaluation was investigated by Kazai (2011) since the satisfaction about the results obtained by different studies varied and some of them had "evidence of cheating and random behaviour among the crowd". This study covered three parameters that are usually used in the experimental design with crowdsourcing which are (1) the pay per label

instead of the pay per task led to a decrease in the return when the pay increases because it attracted less efficient and reliable workers, (2) the worker's qualification is an important parameter, since a qualified worker will produce better quality work and (3) the effort a worker must provide. The study showed that a loaded worker produces less accurate results than a non-loaded one. To reduce the poor judgments that are produced because a worker misinterprets the task's design, Le et al. (2010) studied the effect of training the workers on some gold standard data to determine the reason a worker chose an incorrect answer. They noticed that the workers' answers can become biased which means that if they notice that the distribution of the data leads to a particular label being considered as a correct answer, they tend to pick that answer every time. Therefore, even though crowdsourcing could reduce the time required to evaluate the assessments, the quality control is still an ongoing investigation, and so the work we propose to automate producing the set of qrels could also be an alternative solution for reducing the cost of building a test collection.

2.4 Topic difficulty

The performance of the information retrieval systems differs from one topic to another according to the topic difficulty. Some topics are easier than others. The reason behind topic difficulty and further analysis of the systems' behaviour are discussed in this section. Robust tracks were intended to give more focus on poorly performing topics as the user will no

longer trust a commercial retrieval system when it fails to retrieve relevant information for a hard topic. (Voorhees, 2003). A topic is considered difficult if the median of the AP scores for all the participating systems is below a certain threshold. The first experiments for the Robust Retrieval track were conducted on the TREC-6 to TREC-8 test collections. The topics that were considered difficult are as shown below:

TREC test collection	Difficult topics used in the Robust Retrieval Track
TREC-6	303 307 310 314 320 322 325 330 336 341 344 345 346 347 350
TREC-7	353 354 355 356 362 363 367 372 374 375 378 379 383 389 393 394 397 399
TREC-8	401 404 408 409 414 416 419 426 427 433 435 436 439 442 443 445 448

Table 2. 3: *Number of difficult TREC topics*

These 50 topics were resubmitted to the 78 participating runs and a pool of depth 125 was used. The assessors had to judge the documents as non-relevant, relevant or highly relevant. According to the results they obtained, there were 7 topics that had no highly relevant documents assessed and another 14 topics had less than 5 highly relevant documents. Some of the participating runs used the description field of the topic, others used the title field and the remaining runs used the both fields as queries to retrieve results for a topic. The best results were obtained when using both fields. The worst results were obtained when using the title field only. This track however did not give the reason why the topic was difficult. The degree of ambiguity of the topic usually affects its performance. According to Cronen-Townsend et al. (2002), the clarity score they proposed was related to how

coherent the topic was with the set of the documents in the collection. It is a measure of the dissimilarity between the language model used for the query and the generic one used for all the test collection. When the documents retrieved for a query cover a single topic, the topic has a high coherence and therefore would get a high clarity score, whereas the topic that retrieved a mix of documents that cover several topics would have a low coherence and thus a low clarity score. Hu et al. (2003) aimed at studying the reason why a topic is considered hard and they ran their experiments on TREC-8 adhoc track. They wanted to test if a difficult topic is considered as hard for all the systems participating in the track and whether any documents sets used in the test collection, for example the FT, FBIS, LA FR94 for TREC-8, performed better than the other for that same hard topic. They discovered that some sub-collections or documents sets are more dominant than others, which means that they were able to retrieve more relevant documents for the topic than other sets and that some of the sets were distracting, meaning that they return many non-relevant documents for the topic which affects the average precision of the topic.

2.5 Assessor disagreement

Since building the relevance judgment or qrels is based on human assessment, it will be interesting to look at the notion of assessor disagreement because the concept of relevance differs between human

assessors. How would the disagreement in judgment affect the evaluation of IR system performance?

2.5.1 Early experiments by Cleverdon and Salton

Studies to understand the importance of assessor variability were conducted by Cleverdon (1967) and Lesk and Salton (1968). They came to the conclusion that even though assessors judged documents differently, the system ranking was not affected. The experiments conducted by Salton and then confirmed by Cleverdon, used a set of 1268 abstracts related to documentation and library science with a total of 131500 English text words. Eight persons formulated six queries each, so there were 48 queries in total. After submitting each query, each query author was asked to judge the documents retrieved for the request he had formulated. After forming the first set of relevance assessments, the documents retrieved for each query were passed to another assessor for judgment. The experiments were repeated four times. So at the end there were four different groups of relevance assessments: (1) the ones obtained from the original group of query authors, (2) the ones obtained from non-author judges, (3) the qrels obtained by considering a document is relevant to a given query if it was considered relevant in either the set obtained in (1) or (2) and (4) documents considered relevant only if they were marked relevant in both sets (1) and (2).

All the four sets produced the same ranking of the three different processing methods used, which were as follows: The “word form” method reduced the documents and queries by removing the common words and the final “s” endings then assigned a weight to the remaining word forms. The reduced texts were then matched to obtain the document-query correlation coefficients. As for the “word stem”, in addition to the processing done in the “word form” step, complete suffixes were removed so weights were assigned to stems. The document-query matching process was the same for the first two methods. The third processing method was “thesaurus”. Each stem produced by the previous method was looked up in a thesaurus providing synonym recognition and then the resulting weighted meaning identifiers assigned to the documents and queries were compared. The difference in the output obtained by the sets of judgments was the degree of agreement. The set obtained from using the word stem had a close agreement with the set obtained from the thesaurus.

2.5.2 Additional Experiments on large TREC test collections

Similar experiments were repeated for large test collections in more recent years (Voorhees, 1998) and the results obtained confirmed the previous conclusion which was that assessors’ disagreement does not affect the overall system rankings. In TREC pooling, topic authors are usually the primary assessors of the documents retrieved for that topic. After the

primary assessor was finished with a topic, a new document pool of depth 400 was created for it: 200 documents judged relevant plus 200 randomly selected documents that the primary assessor had judged not relevant. After sorting the documents by their IDs, the new pool was given to two additional assessors (the secondary assessors) who each independently judged the new pool for relevance. The different TREC system rankings were then compared using each set of relevance assessments obtained and this ranking was almost the same. The difference between the minimum and maximum mean average precision (MAP) values was greater than 0.05 for most systems, but the system rankings were highly correlated. The Kendall correlation value for the TREC-4 test collection between the original qrels rankings and the rankings obtained by using the union and intersection of the qrels from different assessors was greater than 0.9. As for the TREC-6 test collection, the Kendall correlation was greater than 0.89. Salton (1968), Cleverdon (1967) and Sanderson (1998) showed that the assessors often agree on the relevance for the top ranked documents but when it comes to low ranked documents, the disagreement becomes obvious. This is in contrast to the findings reported in the previous experiments. The factors that lead to assessor disagreement are still an open problem. A number of studies were made to reassess the document pools obtained by TREC topics.
A study made by Vorhees (2000) showed that only two out of three assessors would usually agree on the same judgment even if they have the

same background. Even though she found out that the agreement between the original TREC assessors and the new judges was only 32.8%, the overall TREC system rankings did not seem to be affected by this low agreement percentage.

2.5.3 Assessor disagreement for graded relevance

Instead of reassessing using binary relevance, Sormunen (2002) used a four-point relevance scale to reassess document pools for 38 TREC-7 and TREC-8 topics and compare agreement between judgments. When the user assigned a (0) as the relevance value, this meant that the document did not contain any information about the topic. A (1) value indicated that the document pointed to the topic and included one sentence about the topic. So the document was "marginally relevant". If the document gave more than just a description about the topic but did not describe it exhaustively, or it covered one or more aspects of that topic, the judge assigned a value of (2). In this case, the document was relevant. When all the themes of the topic were covered in the document, and almost all points of views and facets of the topic were mentioned in several paragraphs, the document was given a value of (3). Now the document was considered highly relevant. From the total number of documents assessed initially by TREC, 48% were assessed relevant and 52% non-relevant, while in the reassessment, 61% of the documents were judged non-relevant and the 39% remaining documents

were assessed as relevant as follows: 20% marginally relevant, 13% relevant and only 6% highly relevant whereas in the TREC assessment, 13% were considered highly relevant, 26% relevant and 36% marginally relevant. Users or new assessors could only find 45% of the documents previously judged by TREC assessors from the pool according to Vakkari and Sormunen (2004). The users were able to identify the highly relevant documents and almost half of the marginal ones. In an investigation of the agreement in judgments between TREC assessors and interactive IR system users conducted by Al-Maskari et al. (2008), the results showed that only 63% of the documents judged relevant by non-TREC judges matched the official TREC judgments.

Carterette and Soboroff (2010) showed that assessor disagreement has an effect on system comparisons for effectiveness measures considering graded relevance levels. They identified several assessor types in poorly trained, autonomous judging environments like crowdsourcing. The assessor types could be unenthusiastic, pessimistic, or lazy. In case of assessor errors, the evaluation measures will be affected and so they proposed strategies to overcome these errors like selecting some documents for reassessment by one or two additional assessors. The documents reassessed were usually the rare ones or the ones that had a low inclusion probability because judging them incorrectly would have more impact on the evaluation than judging documents with high inclusion probability incorrectly.

2.5.4 Shift in assessment by evaluators

Sanderson et al. (2010) examined the drift in assessment between evaluators at different stages. The largest shift was between the highly and partially relevant documents because people's opinions and criteria for making decisions change over time. This was also noticed when the judge had a large number of documents or even when he lost track of the previously judged documents. Furthermore, in most cases, the primary assessors were the topic authors. They had a clear idea about the number of relevant documents each topic submitted to the IR system has, hence when they reach the estimated number of relevant documents, they became more strict in judging the remaining documents. The inconsistency in the assessment resulted in a noticeable impact on the system ranking. A Kendall's tau value of 0.493 was obtained by comparing the 60 systems from the .GOV2 test collection using the first 50% of the relevant documents for each topic from the original, ordered relevance judgments. The authors then randomly partitioned the relevant documents for each topic into two halves and repeated the evaluation for the 60 systems using the different splits of the relevance judgments. The mean Kendall's tau score of the random runs was 0.752.

2.5.5 Other factors influencing assessor disagreement

Webber et al. (2012) studied the relationship between the rank at which a certain document is retrieved and the likelihood that another assessor will disagree with the relevance assessment made previously. The impact of assessment disagreement on the comparative evaluation of automated retrieval systems was found to be minor in almost all previous studies except when using graded relevance or in case of errors like in Sanderson's and Carterette's. Chandar et al. (2013) investigated the relationship between user disagreement and independent factors like readability and cohesiveness. They had three hypotheses to prove: (1) Documents that are more difficult to read will provoke higher levels of assessor disagreement, (2) longer documents provoke more disagreement and (3) less coherent documents also lead to more disagreement between assessors. With the experiments they conducted only the third hypothesis was confirmed. The relationship between the assessor disagreement and the document length was found to be significant. As for the first hypothesis, it failed to be confirmed and actually the reverse was observed: easier documents provoke more disagreement.

2.5.6 Statistics for assessor disagreement

One way of measuring the disagreement between the assessors is by using the kappa statistic, then to determine its influence by assessing the ranking

of different systems with different versions of the judgments. Kappa is intended to show the degree of the agreement between assessors. The calculation is based on the difference between how much agreement is actually present ("observed" agreement) compared to how much agreement would be expected to be present by chance alone ("expected" agreement). Kappa is a measure of this difference, its value ranges between -1 to 1 scale, where 1 is perfect agreement, 0 is exactly what would be expected by chance, and negative values indicate agreement less than chance. "Fleiss' Kappa is a measure of inter-grader reliability or agreement for nominal or binary ratings and is an extended version of Cohen's Kappa. While Cohen's Kappa is only suitable for two assessors, Fleiss' Kappa can be used for more than two assessors" (Fleiss, 1971). Another statistical measure to compute the reliability and agreement between different judges or assessors is Krippendorff's alpha (α). It is widely applicable wherever two or more methods of generating data are applied to the same set of objects, units of analysis, or items and the question is how much the resulting data can be trusted to represent something real. Krippendorff's alpha usually ranges between 0 and 1 where 1 indicates perfect inter-assessor reliability agreement and 0 indicates a level of agreement that can be reached by pure chance and the absence of reliability. An α value >=0.8 suggests that the data is reliable (Krippendorff, 2004).

2.5.7 Summary of the effects of assessor disagreement

In conclusion, even though assessors disagree in judging a document, the overall performance of a system is not affected except when using graded relevance or when there are errors in the assessments as shown in the above studies. Therefore, we can think of new techniques that reduce the human effort and that build the relevance judgment list for test collections automatically which does not involve any assessor disagreement.

2.6 Human Assessment Error

Studying human assessor behavior helps IR researchers in building more accurate information retrieval assessment systems. When a human assessor judges a document for relevance, it is possible for errors to occur. The traditional retrieval metrics are not affected by the errors in assessment (Vorhees, 2000) however newer metrics that sample the documents for judging are very sensitive to assessor error (Carterette et al., 2010; Webber et al., 2010). Bernstein and Zobel (2005), examining the .GOV2 collection, found that identical documents in a test collection were judged differently by assessors because the query was poorly specified. A similar conclusion was drawn by Chen and Karger (2006) who found that the difference in judgments is related to the interpretation of the topic meaning. Scholer et al. (2011) showed that the inconsistency in judgements increased as the distance between the pair of documents also increased. So the time between

judgments and the distance between two documents' matches were associated. The fraction of duplicate documents which were inconsistently judged was of a value which ranged between 15% and 24%.

A study showed that human assessors spent more time to make erroneous judgments than to make correct ones (Smucker and Jethani, 2012). The participants in this study were given a ranked list of documents and they were instructed to save the relevant documents only. The web interface used consisted of two web pages. The first web page showed the summary of 10 retrieved results. The users could click on one of links to read the full document. If the user saved the document, then it was considered relevant. If he did not, the document was judged non-relevant. The number of false negatives (relevant document judged non-relevant) returned by the 48 participants in the study was higher than the number of false positives (non-relevant document judged relevant). False negatives result from not finding the relevant material in the document, and false positives were the result of the final decision being a guess. For longer documents, assessors required more time to judge a document. Taking a longer time to judge a document can also mean that the assessor is finding difficulty in judging the document. When reaching a document rank above 50, the mistakes in judgment increased as well. Another reason for making errors in the assessment is misunderstanding what the topic submitted meant.

2.7 Existing methods to reduce human intervention in building qrels

Since relying on human assessors requires time for building qrels, research has been conducted in the past in order to reduce human intervention in this task. The pooling technique used with the TREC collections was evaluated and some variations on the pooling technique were introduced. Other studies worked on ranking retrieval systems without relevance judgments and approaches were proposed to build pseudo-qrels, which are based on the actual qrels built by human assessors and expanded automatically to add more relevant documents, or on the ranked list of documents retrieved for a certain query, or even to build relevance assessments with no human intervention. We can group the previous work done in three different categories, a first category that focused on the pooling technique and approached the problem as a ranking problem of the different systems that participated in building the test collections, a second category in which studies proposed new evaluation measures to rank the systems and in the third category, searchers proposed new approaches to automatically build the relevance assessments. We provide a detailed description of these studies for each category in the following subsections.

2.7.1 TREC pooling technique and its variations

2.7.1.1The TREC pooling technique reliability

Zobel (1998) showed that TREC results which were obtained from the pooling technique (defined in section 3.2) and which have been judged by human assessors were actually reliable. However, only 50%-70% of the relevant documents we discovered especially for the queries that had a large number of answers. Zobel also stated that it is not correct to say that if two systems have the same effectiveness measure results, which are not significantly different, they are most likely equally good, since the retrieval results might be significantly better for one of them. So the statistical significance is not the most important, but rather we can look at the substantive significance. For example, in one case, he computed the 11-point effectiveness measure for two different systems and found a difference of 0.002 which was statistically significant because for every query the first system outperformed the second slightly, while in another case where the effectiveness measure difference was 0.118, the value was not significant. Even with pooling, the cost of manual judgments can be high. Systems that achieve high recall may actually keep retrieving relevant documents even at a rank higher than 100, but these will not be added to the pool. Therefore, the system effectiveness measurement would change if the pool size were increased. A potential disadvantage of having a measurement depth

exceeding the pool depth was system reinforcement. If one system A retrieved relevant documents at a rank which exceeded the pool depth, but another system B retrieved the same documents by at a rank less than the pool depth, the overall performance of A would be reinforced by the contribution of system B even though a third system C might be implementing a better retrieval technique which was underestimated because it was not reinforced.

Different pool depths were also tested. A pool depth of 100 was evaluated using 11-point recall-precision and precision at depth 100. A pool of size 10 was then evaluated. For some systems the two effectiveness measurements were similar, for others, the difference was significant. For pool depths between 50 and 100, the differences were small. Increasing the pool depth to 1000 had a slight improvement by changing the system ranking by almost 6 positions. Another drawback for the pooling technique noticed by Zobel was that if a system did not have the opportunity to contribute to the pool, its effectiveness may be underestimated. So, in another experiment, he measured the degree to which a certain system contributed to the pool. That was achieved by pooling the results of all the systems and then removing from the pool the documents contributed by the selected run. This experiment was repeated for each system to evaluate its degree of contribution. By neglecting the poorest performing systems, the improvement was of 3.5%. For queries with the most answers (10 queries),

improvement reached 7%. Using the pooling technique, it is impossible to be sure that the most relevant documents were retrieved. Many of the unjudged documents might be relevant. Zobel showed that it is possible to use the most highly ranked retrieved documents to predict with some accuracy how many relevant documents can still be found when going down the rank. Given two pools of depth p-1 and p respectively, n is the number of relevant documents found in the second pool but not in the first, so there were n new arriving relevant documents. Plotting n against p showed the rate at which new relevant documents were coming while the pool was increased and therefore the estimation for the number of relevant documents could be computed following the formula: n= Cps-1 where C and s are constants. The fit of this equation for both pools depth of 100 and 50 seemed to be good. The effect of a new contributing system is also taken into account because it depends on the ranking. Overall, TREC experiments were good in identifying relevant documents but many relevant documents stayed unidentified since all the documents outside the pool were not judged, so it was a weak assumption to say that unjudged documents should be considered non-relevant.

2.7.1.2 Move-to-Front (MTF) technique

The methods proposed by Cormack et al. (1998) aimed to reduce the effort required to judge k documents in case of a large pool depth while maintaining the effectiveness and size of the result set. They proposed

Interactive Searching and Judging, which is an interactive search method that selects the documents to be judged. Four searchers created the set of relevance judgment lists for TREC-6. The system used was the one of the Multitext project. It uses manual Boolean query construction and ranked the documents based on their length. The passages satisfying the query had their search terms highlighted and assessors would record their judgment. The strategy was to formulate a query and keep judging documents until the frequency of the relevant documents dropped, so it relied on human intervention for formulating the query and judging the documents. The Move-to-front (MTF) technique (Cormack et al., 1998) improves the baseline pooling method directly since it selects a different number of documents depending on the system performance. It examines the documents in order of their estimated likelihood of relevance as opposed to TREC pooling. A submission or system run might be considered more effective than others and therefore examined first if it returns more relevant documents. Each document is ranked based on its order in the submission. A document is assigned a maximum priority if it is judged relevant in more than one submission; otherwise its probability is reduced. The MTF strategy had a correlation value of 0.999 with the gold standard system rankings and with only judging half of the number of documents that were judged using the pooling technique; however it still requires human assessors to complete the task.

2.7.1.3 Use of relevance feedback

Sanderson and Joho (2004) studied the effect of pooling on the quality of the qrels generated. They designed several experiments based on previous approaches that used system pooling or relied on one group of searchers to build the relevance judgment list for a test collection. Relevance feedback is a technique used to improve the search result of an information retrieval system. Once the system returns a set of documents, the relevant documents obtained for the query submitted can be fed to the system and the search is repeated again. The relevance feedback can either be blind or automatic or performed manually based on the user selection of documents. Soboroff's iterative relevance feedback technique (2001) used the relevant documents obtained from the pool formed of all TREC runs and then the process was repeated 5 times. Sanderson and Joho used instead the relevant documents retrieved by a single system for relevance feedback and repeated the same experiments done by Soboroff and then computed the correlation using the qrels he obtained and the ones obtained from Soboroff's qrels. Sanderson et al.'s values were very similar to the ones reported by Soboroff. They re-examined Cormack et al.'s (1998) interactive searching and judging (ISJ), because their aim was to test whether the number of searchers used, or the retrieval systems would affect the quality of the qrels obtained. He used the TREC manual runs in which the searchers would create the queries manually and then keep judging documents until

the number of relevant documents found no longer changed. The Kendall's tau correlations obtained for the TREC manual runs using the TREC qrels and then the qrels built by intersecting the TREC qrels and each system's 1000 qrel were found to be > 0.9 for some runs and between 0.8 and 0.9 for others which led the author to conclude that despite the number of searching systems and the searchers participating in building the qrels, the ISJ could be considered reliable. Croft et al. (2009) considered the process of relevance feedback as a machine learning example in IR where the identified relevant and non-relevant documents are considered as training data that can be used to improve the retrieval systems performance. The pseudo-relevance feedback or blind feedback does not involve the user intervention in judging the retrieved documents but rather assumes the top-ranked documents are relevant and uses the terms with the highest weights to expand the initial query and therefore to retrieve more relevant documents.

2.7.1.4 Multi-armed bandits

There are several approaches which have investigated the selection process of the documents to be judged, or what is referred to as document adjudication in Losada et al. (2016). The most recent work done by Losada et al. shows that the technique based on multi-armed bandits to order the documents in the pool and then evaluate them leads to finding the relevant documents more quickly than the traditional pooling technique and

outperforms other methods that tackle the same problem. The multi-armed bandit problem is a reinforcement learning problem (Robbins, 1952) and can be described as follows: given a set of n machines or bandits, where each machine has an unknown probability of distributing a reward, which machine should one choose in order to maximize the total reward over a certain period of time? The user starts first by exploring the machines, where he can pick a random machine and wait for the prize. If the reward obtained was good enough, does the user maintain exploiting the same machine or does he take a risk and explore new machines which could provide better rewards? This is considered to be as an exploitation vs. exploration problem. The machines in this problem can be seen as the runs which return a ranked list of documents for each query. The prize is finding relevant documents.

There are several strategies which can be used to select the next run to examine. These allocation methods can be:

1- Random: a run is randomly selected.

2- ε_n-greedy: a greedy approach always selects the run that has retrieved the most relevant documents in the past. ε_n-greedy acts greedily most of the time but every now and then, it opts for a random run to examine.

3- Upper Confidence Bound (UCB): in the UCB policy, each run is assigned an upper bound confidence index. The leader run at a given round is the one that has the highest accumulated mean of the number of relevant documents retrieved among all previous runs. In order to make sure that the

other runs are indeed inferior to the leader one in the run, an upper confidence bound is defined.

4- Bayesian bandits: each run has a parameter defined as the probability of supplying a relevant document. A Bayesian process assumes that these probabilities are unknown when it first starts and therefore assigns a uniform prior for each run. The initial uniform priors are defined as Beta (1,1). From the prior distribution of each run, one is selected and with the outcome of the run, the probability of supplying a relevant document by that run is revised. Thus, the Beta posterior probability is revised after n pulls over t time: Beta ($S_{i,t}$ + 1,$F_{i,t}$ + 1), where $S_{i,t}$ is the number of 1s or relevant documents in $n_{i,t}$ pulls of run i and $F_{i,t}$ is the number of 0s or non-relevant documents in $n_{i,t}$ pulls of run i. The initial values of these variables (before any pulls) are set to 0. The Bayesian Learning Automaton (Granmo, 2008), which is a parameter-free process that follows the Bayesian process of posterior distributions, had better performance than the UCB and ε_n-greedy strategies. Losada et al. implemented another Bayesian solution which they called MaxMean (MM). MM selects the next run by taking the maximum expectation of the posterior distributions where the expectation of the Beta (α, β) distribution is computed as α/(β +α), while the Bayesian employs random sampling from the posterior distributions to select the next run to examine. This technique was able to retrieve more relevant documents than any of the other pooling techniques suggested.

2.7.1.4 Multi-armed bandits

To further improve their technique for adjudicating documents, Losada et al. defined their environment as non-stationary. In stationary environments, the unknown probabilities of rewarding a run do not change. All prizes or rewards whether they are recent or old are treated equally. This leads to problems, since some runs' outcomes might change over time. This is where it could be more beneficial to use non-stationary (NS) solutions. Recent rewards will be assigned higher weights than old ones. Since the quality of the runs change over time, the MM technique was modified to become non-stationary. In this new implementation, the last relevant document was given more weight for a given run. This MM-NS technique significantly improved the number of relevant documents found with fewer total judgments that had to be made. In chapter 6, we adopt this technique to improve the quality of the qrels we generate. The aim of the thesis was to build the qrels without any human intervention. Looking at the limitations of the approaches we proposed, we decided to use the Losada pooling technique since it allowed retrieving relevant documents faster than any of the previously described pooling techniques and the process to select the documents for adjudication was automated.

2.7.2 Techniques based on new measures

2.7.2.1 The Expected Level of Importance (ELI)

Instead of the traditional TREC pooling technique, Kocabaş and Dinçer (2013) proposed a new pooling strategy based on a new rank-based document criterion which was the expected level of importance (ELI) score. They used TREC 5, 6, 7 and 8 for the experiments. The ELI pooling was similar to the TREC pooling in terms of selecting the top k documents from each result set, except that the top k documents selected were ranked in descending order of their ELI scores, which was an indication of the level of importance of a document. A document with a high ELI score was considered more important to judge or to be added to the pool than a document that was simply among the top k documents of the TREC pooling. The ELI score was computed as following: given n systems' runs, for each topic j and using a dataset of m documents d, a matrix for the topic can be built as shown in figure 2.10 below:

$T_j =$

Result sets	Documents				
	d_1	d_2	d_3	...	d_m
s_{1j}	r_{1j1}	r_{1j2}	r_{1j3}	...	r_{1jm}
s_{2j}	r_{2j1}	r_{2j2}	r_{2j3}	...	r_{2jm}
s_{3j}	r_{3j1}	r_{3j2}	r_{3j3}	...	r_{3jm}
⋮	⋮	⋮	⋮	⋮	⋮
s_{nj}	r_{nj1}	r_{nj2}	r_{nj3}	...	r_{njm}

Figure 2. 10: matrix for topic j (Kocabaş et al., 2013)

Each r_{ijk} represented the inverse rank of the document d_k, so if we take the first row in the matrix, s1j represents the set of documents retrieved in the first run for topic j; r_{1j1} was the inverse rank of the document d_1 in the first run. Hence, if the total number of documents retrieved p was 1000 then the first document retrieved would have a rank of 1000 instead of 1, the second document would have a rank of 999 instead of 2, etc. These ranks were then transformed into weights. When a document was not retrieved in a run, the r_{ijk} value was set to 0. A document that had a higher inverse rank would have a higher degree of relevance to the topic in consideration. The ELI for a document d_k was quantified as follows:

$$ELI(d_k) = \sqrt{d'_k}.d_k = \sqrt{r_{1jk}{}^2 + r_{2jk}{}^2 + \cdots + r_{njk}{}^2}$$

Where d'_k is the transpose vector of d_k. Other potential indicators of document relevance were added to the ELI score. The first was the degree of agreement among runs on the inverse ranks, if a document had the same rank over several runs. The second was the number of runs that retrieved the documents. The ELI score computation formula after using a variance-based approach to express the degree of disagreement on the weights of a document d_k, was as shown below:

$$ELI(d_k) = \sqrt{\frac{d\prime_k.d_k}{var(d_k)}} \quad \text{and } var(d_k) = \frac{1}{n-1}\sum_{i=1}^{n}(r_{ijk} - \bar{d}_k)^2$$

A document was considered to have discriminative power when it was retrieved at different ranks across several runs. The higher the inverse rank variance, the higher its discrimination power is. After defining the scores for the documents and producing the set of qrels, the systems were ranked and the Kendall's tau correlations were computed. The results showed that the ELI pooling strategy can be considered as an alternative to the TREC pooling since for TREC-7 the best tau value achieved was 0.9688 for a judgment limit size of 400 documents and for TREC-8 the best tau value achieved was 0.9578, also for 400 documents.

2.7.2.2 System Similarity Measure

A process related to Soboroff's method, described in section 2.7.1.1 was suggested by Aslam and Savell (2003). A new measure to quantify the similarity between several retrieval systems was proposed, which required quantifying the similarity of the retrieval results. The authors showed a high correlation between the average similarity measure and the one used in Soboroff's method. The hypothesis for their work was that since the best performing systems were poorly identified by Soboroff's method, clearly these systems were performing significantly differently to the others; this is why they could not be judged correctly in the absence of actual qrels which were created by human assessors. Therefore, these systems were evaluated in terms of popularity rather than performance. The popularity of a system

is defined by its average similarity to one other systems. The similarity measure defined was in terms of the common returned documents. Ret_i was the set of documents retrieved by a system i, and the similarity score between two systems was defined as the following:

$$\boldsymbol{SysSimilarity}\,(\boldsymbol{Sys1}, \boldsymbol{Sys2}) = \frac{|\,\boldsymbol{Ret1} \cap \boldsymbol{Ret2}\,|}{|\boldsymbol{Ret1} \cup \boldsymbol{Ret2}\,|}$$

The average similarity score was given by:

$$\text{AvgSysSim } (S_0) = \frac{1}{n-1} \sum_{(S \neq S0)} SysSimilarity(S, S0)$$

Where n is the number of systems. The results obtained when using the average similarity showed a high correlation with the TREC results and the highly performing systems were successfully identified.

2.7.2.3 Reference Count Measure

Wu and Crestani (2003) proposed an automatic method for ranking retrieval systems using "reference counts". For each query, they considered the list of documents retrieved by at least one system and counted the number of systems that retrieved that document, and they called this number the reference count of the document. Once the list was completed, each document had a number of occurrences or a reference count assigned to it. The documents were then ranked in decreasing order of the reference count (RC). The authors also applied a variation of the RC method by assigning

weights to documents depending on their positions in the retrieval list. The documents in higher positions got higher weights. The final list obtained was considered as the newly generated relevance judgment list. The correlation obtained was positive, but, this method, like Soboroff's method, was not good at predicting the performance of the top performing systems.

2.7.3 Other Approaches

2.7.3.1 Ranking systems with no relevance judgments

Several methods for ranking systems without relevance assessments have been tested and proven useful. Soboroff et al. (2001) proposed an idea in which manual relevance assessments are replaced with random sampling from pooled documents. However, Soboroff's method relied on the knowledge of the mean and standard deviation of relevant documents which are not available in practice. They showed that documents returned by multiple runs enhance system ranking accuracy. Relevant documents occur in a pool according to a certain distribution. Instead of having a human assessor judging the documents pooled, the authors suggested the following: For each year, compute the average number of relevant documents found per topic in the pool. (For TREC-8, the average was 5.4% with a standard deviation of 0.048 and for TREC-7, the average was 5.78% with a standard deviation of 0.047). These values can be used for the random selection of documents from the pool.

Pseudo-qrels were created as follows:

- For each topic, draw a percentage value from that year's average and standard deviation as the fraction of the documents to select from the pool (based on previous statistics).
- Select randomly that number of documents to form pseudo-qrels
- Evaluate all runs. Each set of pseudo-qrels produced was considered a trial, so they performed 50 trials.

The choice of the sampling model is the most important component of this methodology. In the TREC experiments, a list of relevant documents is known while in reality they cannot be obtained, so it was necessary to estimate the parameters for the model based on some hand-judged queries. In cases where there were a large number of queries, this task is the same as building the relevance judgment list. This model used a minimal amount of information, since it only needed to be provided with the average number of relevant documents and the standard deviation without any other knowledge about the documents, the topics or the systems. The correlations were not as strong as for the qrels computed manually since the Kendall's tau values for TREC-7 and TREC-8 were 0.369 and 0.459 respectively. This was mainly due to the fact that the top-performing systems were ranked much lower than they should have been. To improve the ranking of the top performing systems, they tested three parameters that had an impact on the ranking: 1) the number of documents to include in the pool (top k), 2) the

number of documents to select as relevant for each topic and 3) how to select documents (as opposed to random). So, they 1) allowed duplicates in the pool, 2) limited the pool depth to build shallow pools by using a pool depth of 10 documents and 3) used exact-fraction sampling by using the true relevant document occurrence rate per topic in choosing the number of documents to select. The greatest improvement was obtained from keeping the duplicate documents in the pool. Documents submitted by more than one system were more likely to be relevant, so when a duplicate document was left in the pool, it had more chance of being selected for the pseudo-qrels. The authors hypothesized that the top systems performed well because they could find rare or unique relevant documents. A relevant document could have been retrieved by only one of the best performing system, yet it was not selected to be included in the pseudo-qrels. In order to prove their hypothesis, they used a smaller pool of depth 10. This shallow pool increased the likelihood of finding relevant documents which were not likely to be found by any other systems. Exact-fraction sampling was shown to be the best form of sampling. For each topic, they chose the exact percentage of the pool that was judged manually as relevant. For example, for topic 401 in TREC-8, 10.8% of the documents were actually relevant, so in each trial, they chose 10.8% of the documents as pseudo-qrels. However, when building a new test collection or in non-TREC test collections, this information does not exist.

2.7.3.2 Use of Machine Learning and Data Fusion technique

A data fusion algorithm is an algorithm that merges two or more ranked lists into a single ranked list in order to provide better system effectiveness measure than any of the systems that produced the individual ranked lists. Nuray and Can (2006) proposed a method based on data fusion to allow ranking retrieval systems in the absence of a relevance judgment list. For the experiments, k systems were selected for fusion. They tested using three different data fusion methods: (1) the rank position (reciprocal rank) method, (2) the Borda count method and (3) the Condorcet method. Using the Condorcet method, the top b documents from each of the k systems were combined. Then the top s% of the merged documents were considered as the "pseudo-qrels". The performance of each retrieval system was evaluated based on the pseudo-qrels obtained and then the systems were ranked. The experiments showed a high consistency with the human based rankings and the Condorcet technique outperformed both the "reference counts" and Soboroff's methods.

Jayasinghe et al. (2014a) investigated in their work the possibility to find relevant documents for the TREC GOV2 test collection which can only be found by manual runs since these runs are highly effective and contribute to the judgments pool with unique relevant documents. The authors fully assessed the documents returned at ranks higher than 50 for several runs

manually. The depth 50 is the depth used to build the GOV2 pools at NIST. The documents returned by the automatic runs at rank higher than 50 were assessed and considered as the initial pool to be expanded automatically by relevant documents which are usually found by the manual runs. A second pool was then formed using the documents retrieved by the automatic runs which were not judged by assessors. The aim was to find the relevant documents in this pool that the manual runs are usually good at finding. The documents in the second pool were ranked according to two different methods and the top-i of them were considered as the most promising documents to be relevant for a given query. The first ranking method adopted was the Borda count which is a voting technique. The document is assigned as a Borda score the sum of the ranks at which it was retrieved over different runs. The second ranking method used a linear SVM classifier that was trained using the judged documents from the automatic runs and tested using the documents judged in the manual runs. The classifier had to label each document in the unjudged pool returned by the automatic runs as either relevant or non-relevant to the query. The documents were ranked according to the score assigned by the classifier which represents the likelihood of the document being relevant. Only the manually judged documents could be evaluated. A combination of the Borda and SVM ranking methods was also applied. The evaluation of the automatically expanded pool was done by comparing the MAP value at different pool depths for all

the runs and then by computing the Kendall's tau correlations between the rankings obtained using the TREC qrels and the union of the documents found in both pools using both ranking methods. A Kendall's tau above 0.9 could be achieved using a pool depth greater than 100. The authors expanded their work and explored the use of data fusion techniques to form the pools (Jayasinghe et al., 2014b) instead of using the methods based on the voting technique. However, they kept the use of the machine learning. In data fusion techniques, the ranked results returned from multiple IR systems are merged to form a new set according to scores computed using a certain approach. In addition to the Borda Count, CombMNZ, CombANZ and Static Judgment Orderings were tested. They all use a voting scheme, but the computation for the final score differs from one technique to another. In the CombMNZ, the fusion score for each document is the total votes obtained by the document multiplied by the number of IR systems that retrieved the document, while in the CombANZ, the score is averaged over the number of systems. Expanding the judged documents retrieved by the automatic systems to find the relevant documents that are usually found by manual runs was tested on TREC-8 and GOV2 test collections. The combination of the Borda Count and machine learning led to finding more relevant documents than using other techniques. The machine learning approach was shown to retrieve more unjudged documents than other methods. The combination of machine learning, and data fusion methods

resulted in finding a large number of relevant documents that are usually returned by manual runs only. Evaluating the complete effectiveness of these techniques was not possible since there are many unjudged retrieved documents, which had to be removed from the evaluation process.

Another retrieval application that requires having relevance judgments is the filtering track that answers the long-term information need of the user by learning its interests. Soboroff and Robertson (2003) presented in their work several approaches to build a filtering test collection and reported the challenges they faced to achieve the task. Filtering systems need relevance judgments during a run while in TREC adhoc test collections, the relevance judgments are produced after the experiments are completed. The challenge for the filtering systems is to have a test collection that has enough relevance judgments to be produced during the run when training and adapting the data, and then expand them in the test phase. That is why these systems used test collections from previous years which have their relevance judgments formed. In the TREC adhoc test collections, the assessors are the topic developers who usually create the topic by exploring the documents. In the filtering track, the assessors have to be provided with the relevance judgments up front and then they perform exhaustive searching to expand the number of relevance judgments by giving them more time to develop each topic. The relevance feedback process was repeated over several iterations until no more relevant documents were

found for a topic. Each new set of relevant documents judged during a given iteration was fed to four different systems each deploying a different retrieval function: PRISE which is an NIST retrieval system implemented using the BM25 model, Cornell's SMART system that uses the Rocchio feedback and the ltc.ntc weighting, the YARI language model and the Bag of Words toolkit developed by McCallum's using the NB and SVM algorithms. After collecting the top 100 documents returned by each of these systems to a given topic, the CombMNZ fusion algorithm was used to merge these results sets and then the top 100 highly ranked documents were judged. The process was repeated over 5 days or until no more relevant documents were found. The described approach for producing more relevance judgments was shown to be useful for the filtering track even though it required a long time to be completed.

2.7.3.3 Examining participating systems

Examining the uniqueness of systems, Spoerri (2007) 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. However, the best run was selected based on the mean average precision and to compute this measure, Spoerri had to refer to the true or actual qrels to evaluate each run. Therefore, if there were n participating teams, there would be n runs selected. Next, five random runs from the n ones were randomly

chosen and each run was selected for exactly five trials. To compare between the five runs or systems, the author computed the overlap between the systems' results for each TREC topic. The total number of relevant documents found was averaged over the 50 topics. This number was computed in the form of a percentage of documents found by a specific number of several systems. The example Spoerri gave to explain how the percentage is computed is as follows: "If system A retrieves 1000 documents for a topic and this result set is compared with the results of four other systems, then we compute the percentage of A's documents that are found by all five systems, the percentage of A's documents found by four systems and so on, ending with the percentage of A's documents that are only found by the system A itself". The systems or runs will be then ranked based on the average percentage computed rather than using the MAP. The correlations obtained using this method outperformed the ones obtained by Wu and Crestani (2003) and Soboroff et al (2001).

In work that involved clustering, Shi et al. (2010) 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.

2.7.3.3 Discrimination power of new documents

In a study which aimed to minimize the judgment effort required to build a test collection that produces a reliable evaluation results, Carterette et al. (2006) provided a new algorithm for selecting the documents to judge based on the average precision (AP) metric. After showing that AP was normally distributed over a set of relevance judgments, a new algorithm was suggested to select the documents for judging. By definition the AP is the average of the precisions at ranks where a relevant document is found. ΔAP is the difference in AP between two different retrieval systems A and B.

If system A was thought better than system B because $\Delta AP > 0$, for documents which have not been judged manually, they considered judging them automatically as follows: if the effect of judging them relevant would be to increase the difference in AP between the two systems, they should indeed be judged relevant; otherwise they be judged non-relevant. In this way new documents were automatically judged until a stopping criterion was reached.

2.7.3.4 Use of query aspects

Efron (2009) used "query aspects" to automatically build a set of qrels. To explain better what an aspect means, consider TREC topic 402 that has

"behavioral genetics" as its title. The same information need might be represented by different aspects such as "behavioral disorders" or "genetics addictions". Several manually derived query aspects were considered as queries and the union of the top 100 documents retrieved for all queries on a TREC topic from a single system was considered to be the set of "pseudo-qrels" or "aspect qrels". Even though building the qrels did not involve any human intervention, since they resulted from combining the top k retrieved documents from all query aspects, but query aspects created for each TREC topic were mostly created manually. The searchers who created the aspects had to search for synonyms for the topic terms then read more about the topic in order formulate new queries. This method does not rely on a system pool, and does not eliminate human intervention, but rather reduces the effort. As a variation on Soboroff's method, Sakai and Lin (2010) randomly sampled 10% of a depth-30 pool which was obtained by using the TREC pooling and which contained duplicate documents and considered the results as relevant. Pseudo-qrels were generated after repeating the process 10 times to produce 10 pseudo-qrels files. They then ranked the runs by how each document set resembles the others, as was suggested by Aslam and Savell (2003). In conclusion, they showed that the simplest method of forming "pseudo-qrels" based on how many systems returned each pooled document, performed as well as any other existing method.

2.7.3.5 Nugget extraction

An interesting method showing the power of constructing a set of "nuggets" (information extracted from the document) in building test collections was presented by Pavlu et al. (2012). Figure 2.11 below shows a list of nuggets extracted for a TREC topic:

Query#	434
Title	Estonia, economy
Description	What is the state of the economy of Estonia?
Narrative	Documents that give concrete economic information such as economic statistics, entering economic unions and treaties, or monetary performance are relevant, as are discussions of economic issues such as transportation or pollution.
Nuggets	In 1993, Estonia's foreign trade more than doubled to around EEK20,000m.
	Estonia's foreign trade deficit declined by 179.1 million kroons when compared with April.
	In other words, Estonia's foreign trade deficit amounts roughly to one third of the export volume.
	In their joint communique, the foreign ministers welcome the political and economic changes in Estonia, Latvia, Lithuania, Poland, and Russia.
	Last year, Prime Minister Mart Laar told you, and I quote, that "It is imperative to realize that Estonia, Latvia and Lithuania cannot be abandoned, not only for their own sake, but for the precedent that deserting our countries would set.
	Last year was successful to Estonia both economically and politically, Laar said while adding that, hopefully, Estonia and the European Union would conclude free trade agreement in 1994.

Figure 2. 11: Example of nuggets for Topic 434

For each query, a sample of documents is judged by an assessor, and from the documents judged relevant, the assessor extracts relevant nuggets of information in the form of sentences. This set of nuggets is then used to infer the relevance of all unjudged documents containing these nuggets. This process is done automatically; however it looks impractical since it relies on human assessors to extract nuggets which is a time consuming task. Additionally, the sample of documents to be judged is fixed. Rajput et al.

(2012) suggested a method to solve these problems by adapting an "Active Learning" principle. The methodology is to find relevant information which immediately leads to relevant documents, hence lead to more relevant information. The framework they proposed uses three independent components:

1. Document selector: decides which document to judge next.
2. Nugget extraction: responsible for deciding which nugget should be considered relevant from the document and determining the weight of the nugget.
3. Text Matching: measures whether the nugget and the document describe the same information or not, using basic Natural Language Processing (NLP) techniques.

This principled method of automatic nugget extraction while assessing a document enhanced the efficiency in building a test collection. It minimised the human effort needed to form qrels, and when evaluated against baselines like the TREC qrels showed that with the nugget-based approach could achieve the same recall as other methods like TREC pooling and pooling with relevance feedback by judging fewer documents as shown in Table 2.4 below.

	Recall				
Adhoc99	60%	70%	80%	90%	100%

Depth-Pooling	417	561	724	962	1302
Relevance Feedback	313	407	516	681	1130
Nuggets	187	245	343	498	892
Web09					
Depth-Pooling	248	291	352	418	487
Relevance Feedback	196	232	263	326	447
Nuggets	129	169	245	325	448
Robust05					
Depth-Pooling	321	402	481	553	600
Relevance Feedback	222	274	328	447	614
Nuggets	193	231	287	378	594

Table 2. 4: Number of documents assessed to achieve a recall percentage among different methods

2.7.3.6 Distance-based technique

Mollà et al. (2013) proposed a document distance-based technique using the following formula:

$$\text{distance measure} = 1 - \text{cosine similarity measure} \quad (2.1)$$

Sets of documents known to be relevant were expanded by all the documents found to have a sufficiently small distance from them. The experiments conducted used a list of references of a sample clinical systematic review in which no negative judgments existed. Based on the

hypothesis that documents in a cluster are usually very similar, the methodology represented below was applied:

For each query (q) of the 50 TREC Topics

Let R_q be one set of known relevant documents for (q)

For each document (d) in the pool of all available documents

Measure the distance between (d) and each document in R_q

Keep a record of the minimum distance obtained

End For

Sort the results in ascending order of distance

Select the top K documents

Add the newly selected documents to R_q ← Set of new qrels

End For

Distance-Based Pseudocode

The total number of documents was set to N=100 and the top K documents selected after sorting was 0.2%. Retaining just small percentages of qrels led to a better improvement over the baselines which were built by using 16 different Terrier ranking algorithms (listed in table 2.5) to generate the runs. Terrier (Ounis et al., 2006) is an open source retrieval system that implements the state-of-art indexing and retrieval functionalities. Retrieval approaches are provided, such as Divergence from Randomness, BM25F, and term dependence proximity models.

BB2	BM25	DFR_BM25	DLH
DPH	DFRee	Hiemstra_LM	DLH13
IFB2	In_expB2	In_expC2	InL2
LemurTF_IDF	LGD	PL2	TF_IDF

Table 2. 5: List of 16 runs from the terrier package

The authors then evaluated the system rankings by 1) using the original qrels, 2) using a subset of the qrels and then 3) using the same subset selected in the previous experiment with the expanded list of documents automatically judged relevant added. The results highly correlated with the ones obtained from the original qrels. Even though this method allows finding more relevant documents for a topic and therefore building pseudo-qrels for it, it requires having a known set of relevant documents for the query to be able to expand it. This approach proved that relevant documents are at a close distance from each other and that the selection of the top K documents found nearby a known relevant qrel was similar to the K-Nearest Neighbour Algorithm which inspired one of our techniques that we describe in section 3.5.

2.7.3.7 Cutoff percentage and exact count

Rajagopal et al. (2014) used two independent approaches to build pseudo relevance judgements. One technique which is completely automated and does not require any human intervention and is based on a "cutoff

percentage" of the number of documents to mark as relevant or non-relevant; the other is called "exact count" and requires previous knowledge of the number of documents judged relevant by the human assessor. The results obtained showed that the approach based on the cutoff percentage gave better Kendall's tau and Pearson correlation values between system rankings based on human-built qrels and pseudo-qrels. Both techniques used the number of occurrences of a document retrieved over all system runs to determine its relevance, whether it is relevant or non-relevant to a topic.

For the "cutoff percentage" approach, their initial hypothesis stated the following: the higher the number of occurrences of a document in the pool of documents found relevant by a range of systems, the higher is the probability of this document being relevant. In their experiment, a variation of the TREC pooling technique was presented, since pseudo relevance judgments were built without any human assessors' involvement. Two cutoff percentages (>50% and >35%) of the document occurrences were studied. A pool of depth 100 was used. The steps followed for TREC-8 are described in the below pseudocode:

```
For each query (q) of the 50 TREC Topics
    Collect the top 1000 retrieval results from all runs
    Pool with depth K =100
    For each document (d) retrieved for (q)
```

Let occurrence (d) ← number of runs that retrieved (d) for (q)

Convert the occurrence(d) to a percentage of systems that retrieved (d)

End For

Order by occurrence(d) in descending order

Let Cutoff be a percentage value

Select all documents with a percentage value > Cutoff to be relevant

End For

Calculate MAP for each run

Rank runs according to MAP

Cutoff Pseudocode

To clarify the step where we convert the occurrence (d) to a percentage, we give this example: for a total of 129 systems, if doc1 occurred in 10 systems, the percentage value is about 7%. As for the cutoff percentage, if we set Cutoff =50% or 35%, for topic 1, doc34 had a percentage value of 64% for example, this means that doc34 will be considered relevant. If the document's percentage is below the cutoff percentage, it will be considered non-relevant.

The results reported by Rajagopal et al. are shown in Table 2.6:

TREC-8 (129 Systems)	Kendall's tau	Pearson	Harmonic Mean
cutoff >50%	0.506	0.739	0.600
cutoff >35%	0.515	0.736	0.605

Table 2. 6: Kendall's tau and Pearson correlation for MAP values for depth 100 using different cutoff percentage for TREC-8

The second approach suggested by Rajagopal et al. used the same steps as described above, except that in the step of setting the Cutoff value, instead

of choosing a cutoff percentage as a threshold value, the threshold used is the exact count of number of relevant documents for a query. For example, if for TREC topic 401, the human assessors judged 40 documents as relevant, the top 40 documents retrieved with the highest 40 percentage values from step 5 will be judged relevant as for the remaining of the documents they will be judged non-relevant. This "exact count" technique depends on knowing the number of relevant documents for each topic, but when building a new test collection, this information is not available. The Cutoff percentage values selected in Rajagopal's et al. technique were somehow done at random, and setting a low value for the cutoff such as 35% would lead to many non-relevant documents considered as relevant and thus this technique can no longer produce reliable qrels that could be used to discriminate between the different systems.

A question that extends from the above experiments based on the "cutoff percentage" is does increasing the cutoff percentage provide better results? The reason behind increasing the cutoff percentage is to minimise the error margin when judging documents as relevant, and therefore a proposed improvement to Mollá's technique is to use the documents resulting from a high cutoff percentage as the set of qrel and then use it to find more relevant documents by computing the distance between documents. This will be examined in chapter 5.

2.8 Conclusion

In conclusion, test collections are widely used in the evaluation of information retrieval systems. However, the effort it takes to build these collections and to judge the pool of documents for Text REtrieval Conference (TREC) collections, the standard test collections used for information retrieval experiments, is time and effort consuming. In the judgment process, errors might occur and assessors frequently disagree in judging the same document. However this does not affect system rankings except when using graded relevance. The techniques that have been proposed so far in order to generate the pseudo-qrels or relevance judgment list built in a semi-automatic way, or to completely automate the generation of the query-based relevance set (qrels) use binary relevance and are based on TREC test collections (Soboroff et al.,2001; Aslam and Savell, 2003; Wu and Crestani, 2003; Nuray and Can, 2006; Spoerri, 2007; Sakai and Lin, 2010; Rajput et al., 2012; Rajagopal et al., 2014). Some of the semi-automatic approaches still require human intervention (although requiring less effort) whether in building new aspects or queries for the TREC topics or using some of the original TREC qrels and then expand them automatically (Cormack et al., 1998; Efron, 2009; Mollà, 2013). Other techniques are based on previous statistics related to the TREC test collection which are needed to make predictions or estimations to automatically generate the qrels. However, this required information may not be available in practice. For example, we may

want to build a non-TREC test collection where there are few participating research groups and we cannot form a pool as large as the one provided by the TREC National Institute for Standards and Technology (NIST). It would be useful to discover a technique that will work using a single system instead of having to use a pool of different systems' runs, especially given that most of the studies that eliminated the use of pools of documents, which were formed by combining the top K documents retrieved from each system for a query, had good outcomes when compared to the baseline results obtained from TREC pooling. The method proposed by Rajagopal et al. (2014) based on a cutoff percentage is independent of the test collection and does not require any human intervention. However this technique is unable to identify the best performing systems. For a lower cutoff percentage of documents (<50% or 35%), we can no longer be sure that the documents judged relevant are actually relevant because this percentage seem to be very tolerant in judging documents as relevant. Many non-relevant documents will be considered relevant using a low cutoff percentage. So can we enhance this technique to expand more relevant documents and obtain a set of qrels which can provide a better correlation with the ones built by human assessors and to identify the best performing systems? We propose a method in this thesis based on Rajagopal's work that proposes a solution for the disadvantages described above. We intend to prove that it can be used on both TREC and non-TREC test collections.

Chapter 3 – Methodology

3.1 Introduction

In the previous chapter, we discussed all the approaches that have aimed to build the relevance assessments automatically or attempted to rank the retrieval systems in the absence of human judgments or with incomplete ones. We showed also that some of these techniques were not fully automated, but rather they require a minimal human effort in some of their aspects, like forming the queries in Efron's work (2009). Others require some previous knowledge about the distribution of the relevant documents in the pool. The question of comparing between the retrieval systems is usually answered by ranking them and then measuring the correlations between the rankings obtained from using the human qrels and the one from the automatically produced qrels. Since there have been no perfect correlations achieved so far, looking for new techniques which correlate better with the human ranking and discriminate between the best systems with minimal or in absence of complete judgments is an interesting field of research especially since previous automatic techniques have failed to discriminate between the very best systems even though they provided positive correlations for the overall systems ranking. Therefore, this chapter sets out the research questions which were based on conclusions drawn from the previous work done and will describe in detail the different

methodologies adopted to answer each of the research questions. We also discuss the test collections and the evaluation metrics used in the experiments.

3.2 Test Collections

Before we describe the experimental design and process for the approaches used in this thesis, we introduce in this section the test collections used in the experiments. Test collections are considered the standard framework for evaluation in information retrieval. A test collection is a list or set of documents, a set of manually constructed topics, and a list of relevance judgments relating a document to a topic, and is built by human assessors. This relevance judgment list is a matrix that shows the topic number, the document ID and the document's binary relevance. If a document is relevant, we assign the value (1), if it is non-relevant, we assign a (0) value. This is known as the Cranfield paradigm, which was first started by Cleverdon in 1967. In his experiments, Cleverdon used 1400 documents and 279 queries. Research papers were used to create queries and the document collection was comprised of the pooled references. Relevance judgments were made by the query providers and augmented by students who actually judged every document.

The Text REtrieval Conference (TREC)[3], co-sponsored by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense, was started in 1992 as part of the TIPSTER Text program. Its purpose was to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. TREC test collections are widely used in evaluation. They consist of a set of documents, a set of 50 topics and a set of relevant assessments or qrels judged by human assessors. For European languages other than English, we use the Cross-Language Evaluation Forum (CLEF) test collections which have the same structure as TREC, and we provide more details about each of these test collections in the next sections.

3.2.1 TREC Test Collections

A TREC workshop consists of a set of tracks, areas of focus in which particular retrieval tasks are defined. The Web track for example explores Web-specific retrieval tasks, including diversity and efficiency tasks, over collections of up to one billion Web pages. The best-known test collections are the ones used for the TREC Ad Hoc track during the first 8 TREC evaluations between 1992 and 1999. In total, these test collections comprise 6 CDs containing 1.89 million documents (mainly, but not exclusively, newswire articles) and relevance judgments for 450 information needs, which are called topics and expressed in detailed text passages. The early

[3] http://trec.nist.gov/overview.html

TREC competitions each consisted of 50 information needs, evaluated over different but overlapping sets of documents. TRECs 6-8 provide 150 information needs over about 528,000 newswire and Foreign Broadcast Information Service articles. Because the test document collections are so large, there are no exhaustive relevance judgments. For each annual TREC evaluation, NIST provides a set of documents and a list of 50 topics. The research groups participating in the workshop run their retrieval systems and return a list of 1000 documents for each topic to NIST. NIST will then select the top 100 documents from each of the submitted runs for each topic, remove the duplicates, then order the documents by their IDs. Any document not found in the pool is considered to be non-relevant (Spärck Jones and van Rijsbergen, 1976). This technique is known as the pooling technique (see figure 3.1).

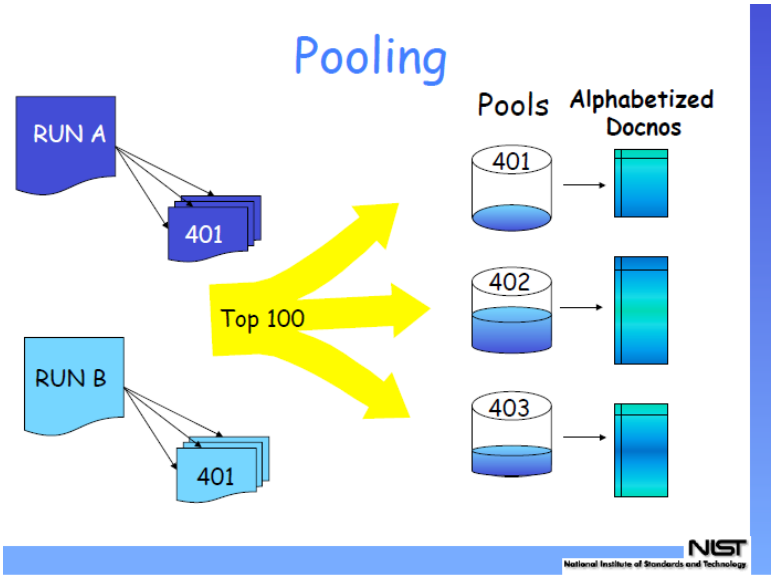


Figure 3. 1: Pooling diagram (Harman, 2010)

The resulting pool of documents for each topic is then given to human assessors, mainly the researchers who formulated the topics, to be judged, which means assigning a binary relevance value to each document. A value 0 indicates that the document is non-relevant to the topic for which it was retrieved, while a value of 1 indicates its relevance to the topic. The participants' run will be then evaluated based on the list of relevance assessments or qrels formed by the human judges. Different evaluation metrics are computed such as the precision, recall and average precision (AP) for each topic, and the mean average precision (MAP) over all the topics. These metrics are described in section 3.3.

The best, average and poor systems are then identified according to their MAP scores and thus their ranks. In our experiments we used three different ad hoc test collections: TREC-6 (Voorhees and Harman, 1997), TREC-7 (Voorhees and Harman, 1998) and TREC-8 (Voorhees and Harman, 1999). The documents provided are news articles from different sources: The Financial Times (FT, 1991-1994), Federal Register (FR, 1994), Congressional Record (CR, 1993), Foreign Broadcast Information Service (FBIS) and the LA Times. Details of three TREC test collections are given in table 3.1.

Test Collection (ad hoc)	Disks	Number of documents	Topics	Number of Participating Systems	Number of relevance assessments
TREC-6	disks 4 & 5 FBIS-2	~ 2 GB of text documents: 556077 docs.	301-350	74	72270
TREC-7	disks 4 & 5 (no CR)	528155 docs	351-400	103	80345
TREC-8	disks 4 & 5 (no CR)	528155 docs	401-450	129	86830

Table 3. 1: Details of the TREC-6, TREC-7 and TREC-8 test collections.

Figure 3.2 shows a sample of the TREC topics, taken from the TREC-8 test collection. Every topic consists of a number such as 401 (the "num" tag), a short query (the "title" tag), a longer query (the "description" tag) and a paragraph giving additional details about the topic (the "narrative" tag).

<top>
<num> Number: 401
<title> foreign minorities, Germany

TREC topic example

<desc> Description:
What language and cultural differences impede the integration
of foreign minorities in Germany?

<narr> Narrative:
A relevant document will focus on the causes of the lack of
integration in a significant way; that is, the mere mention of
immigration difficulties is not relevant. Documents that discuss
immigration problems unrelated to Germany are also not relevant.
</top>

Figure 3. 2: TREC-8 Topic example

The documents are contained in files and they are structured with tags as shown in figure 3.3. They have unique numbers or IDs ("docno", such as FBIS3-17077) and the text tag represents the content of the document. We

process the documents by removing all tags and keeping the content only for the experiments.

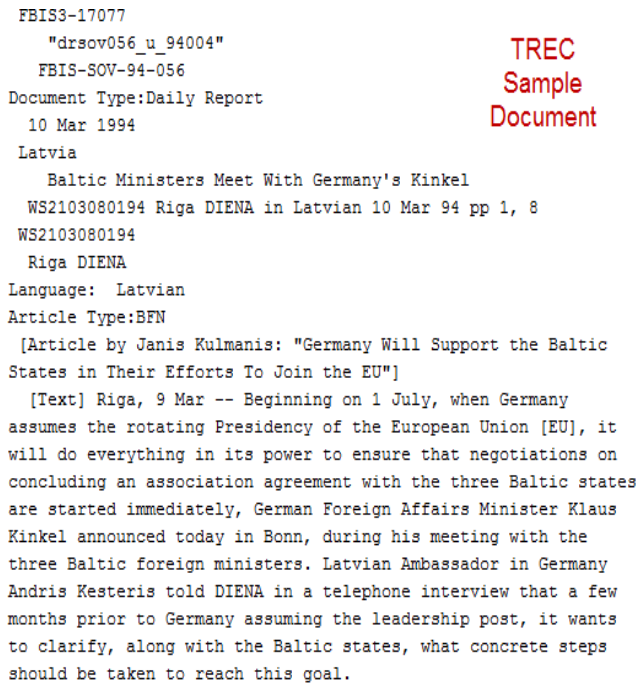


Figure 3. 3: TREC Document

The relevance judgments list for the TREC test collections is ordered by topic number and then by document number followed by a binary relevance value, 0 for a non-relevant document and 1 for a relevant document. Figure 3.4 is a snapshot of the relevance judgments list or qrels for TREC-8.

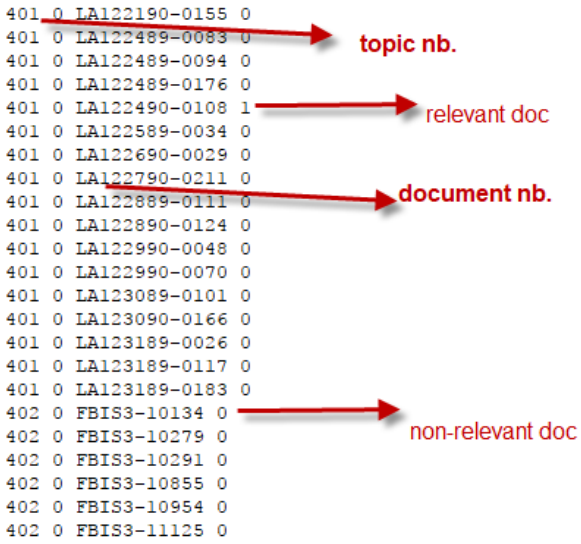


Figure 3. 4: TREC-8 qrels snapshot

The number of relevant documents varies from one topic to another. Some topics could have many relevant documents while others could have very low numbers, which is usually related to how hard a topic is. Because our research focuses on the idea of building the set of qrels automatically and thus finding as many relevant documents as possible for a topic, we provide in table 3.2 below details about the number of documents judged relevant for each topic in all three TREC ad hoc test collections, and in bold are the ones that have less than 20 relevant documents. In the next chapters, we discuss how the proposed techniques would have been affected by the difficulty of a TREC topic.

TREC-6									
Topic Nb.	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**
301	474	311	186	321	234	331	222	341	81
302	77	**312**	**11**	322	34	332	278	342	23
303	**10**	313	107	323	63	333	72	343	290

304	226	314	45	324	162	**334**	**18**	**344**	**5**
305	35	315	67	325	24	335	70	345	39
306	352	316	35	326	48	**336**	**12**	346	106
307	215	**317**	**14**	**327**	**18**	337	98	347	157
308	**4**	318	128	**328**	**9**	**338**	**5**	**348**	**5**
309	**3**	319	187	329	50	**339**	**10**	349	73
310	**13**	**320**	**6**	330	60	340	81	350	69
TREC-7									
Topic Nb.	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**
351	48	**361**	**9**	**371**	**17**	381	28	391	178
352	246	362	39	372	49	382	22	392	105
353	122	**363**	**16**	373	33	383	146	393	71
354	361	364	35	374	204	384	51	**394**	**17**
355	45	365	35	375	80	385	86	395	213
356	**17**	366	99	376	102	**386**	**19**	396	59
357	270	367	189	377	39	387	85	397	27
358	51	368	61	378	98	388	51	398	145
359	28	**369**	**13**	**379**	**16**	389	194	399	102
360	151	370	336	**380**	**7**	390	134	400	125
TREC-8									
Topic Nb.	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**	**Topic Nb.**	**Nb. Relev Docs**
401	300	411	27	421	83	431	130	**441**	**17**
402	80	412	123	422	152	432	28	442	94
403	21	413	69	423	21	**433**	**13**	443	102
404	142	414	39	424	171	434	347	**444**	**17**
405	38	415	136	425	162	435	117	445	62
406	**13**	416	42	426	202	436	180	446	162
407	68	417	75	427	50	437	72	**447**	**16**
408	118	418	116	428	118	438	173	448	46
409	22	**419**	**16**	**429**	**11**	439	219	449	67
410	65	420	33	**430**	**6**	440	54	450	293

Table 3. 2: Details of the number of relevant documents for each TREC topic

In more recent years, NIST has done evaluations on larger document collections, including the 25 million page .GOV2 web page collection (Manning, 2008). While this collection contains less than a full terabyte of

data (~426 GB), it is considerably larger than the collections used in previous TREC tracks. For TREC 2004, the collection was distributed by the Commonwealth Scientific and Industrial Research Organisation (CSIRO) on a single hard drive. Later, the University of Glasgow took over the distribution of the WT2G, WT10G, .GOV (~ 18 GB) and .GOV2 Web Research Collections. The data is crawled from Web sites in the .gov domain. A web crawler is a program that looks for webpages in an automated manner. They follow links on those webpages and then move from link to link to bring up-to-date data about the webpages back to the server where the data is stored. For example, Google's web crawler is called "Googlebot". One of the largest web test collections is the ClueWeb09[4] dataset which consists of about 1 billion web pages (size 25 TB) in ten different languages.

In our experiments, due to the resource limitations of the server capacity, memory and processing unit, we did not use web test collections but rather the ad hoc test collections TREC-6, TREC-7 and TREC-8, we provided some details about these large test collections, so we can discuss in later chapters how we anticipate the performance of our proposed techniques applied to the ad hoc test collections to be.

3.2.2 CLEF Test Collections

The Cross-Language Evaluation Forum (CLEF) is a series of evaluation activities that were supported by the Information Society Technologies

[4] https://lemurproject.org/clueweb09/index.php

program of the European Union. The CLEF consortium supports all information retrieval tasks operating on European languages in both monolingual and cross-language contexts. It uses the same methodology for evaluation as the TREC test collections. A CLEF test collection consists of a set of documents, topics and relevance assessments. We selected the CLEF 2003 (Braschler, 2003) campaigns for our experiments and we tested the approaches on the French and Finnish test collection from the year 2003. The CLEF collection has more than 7 European languages and it can be used for cross-lingual, bilingual and monolingual information retrieval tasks. Some details about the test collections used are shown in table 3.3.

	Articles resource	**Number of documents**	**Topics**	**Number of runs**	**Number of relevance assessments**
French 2003	Le Monde 94 SDA 94 SDA 95	42615	141 – 200	36	20358
Finnish 2003	Aamulehti (daily morning newspaper) 94-95	55344	141 – 200	13	15605

Table 3. 3: Details of the CLEF 2002- 2003 test collections

The structure of the topics, the documents and the relevance assessments for the CLEF collections is similar to that of the TREC collections.

3.2 Search engine evaluation

Web search engines are another application of IR systems. The Google and Yahoo web search engines crawl many terabytes of data every day and make this data available for users in just milliseconds as a response to the queries they submit from all over the world. In designing a search engine several features are taken into consideration: effective ranking algorithms, evaluation and user interaction. The performance of a search engine in terms of the response time, query throughput and indexing speed are the key features for deploying the system in large-scale environments. In web search engines, the age of the result retrieved ("recency and freshness") and the scalability – whether the system continues to work as the amount of data and the number of users grow – are also factors included in the design. A diagram presenting the design of a search engine and the core issues it accounts for are shown in Figure 3.5.

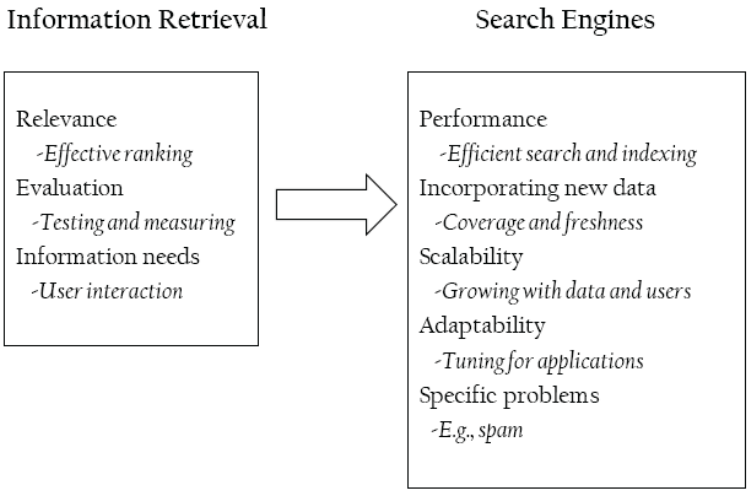


Figure 3. 5: Search engine design and core IR issues (Croft et al., 2009).

The architecture of a search engine is meant to achieve two goals: (1) effectiveness: retrieving as many relevant documents as possible and (2) efficiency: process the queries as quickly as possible. The query process in a search engine has 3 main stages:

1. User interaction: the interface presented to the user to submit his or her query.
2. The ranking component which is the core of a search engine. It takes the transformed query entered by the user and generates a ranked list of documents using scores based on a retrieval model. Ranking should be efficient and effective.
3. The evaluation component measures and monitors the efficiency and effectiveness of the search engine. The results obtained will be used to improve the ranking component. Evaluation is not performed in real time, but it is very important in the lifecycle of the query process.

As described in section 3.2, for the ranking analysis, test collections are usually used because they provide a controlled environment for testing. A basic requirement for evaluation is to be able to compare the results of different techniques, so the experiments have to be repeatable and settings should be fixed. Test collections consisting of a set of documents, queries and relevance judgments list are assembled for this purpose.

In order to rank different retrieval systems, or to determine which search engine (SE) is better to use for a current application, we need to quantify

the results produced by the system. The metric used in performance evaluation should be associated with the notion of relevance of the results with respect to the user (Baeza-Yates et al., 2011).

3.2.1 Recall and Precision based measures

The first effectiveness measures introduced in the Cranfield studies were precision and recall. Recall measures how well a search engine is doing in retrieving a relevant set of documents for a query while precision measures how well it rejects non-relevant documents. Recall (R) may be defined as the fraction of relevant documents retrieved:

$$Recall = \frac{number\ of\ relevant\ documents\ retrieved}{number\ of\ relevant\ documents} = R\ (retrieved \mid relevant)$$

In many search applications, especially web search engines, users tend to look at the top part of the ranked result list, typically the 10 hits on the first page, rarely visiting another result page. In such cases, recall is no longer enough. Not only does the user want relevant documents, but he also expects them to be at the top of the retrieved list. So, we define Precision (P) which is the fraction of the retrieved documents which are relevant:

$$Precision = \frac{number\ of\ retrieved\ relevant\ documents}{number\ of\ retrieved\ documents} = P\ (relevant \mid retrieved)$$

Another retrieval effectiveness measure based on recall and precision is the F measure. It is used for evaluating classification performance. It summarizes effectiveness in a single value and is defined as the harmonic mean of the precision (P) and recall (R):

$$F = \frac{1}{\frac{1}{2(\frac{1}{R}+\frac{1}{P})}} = \frac{2RP}{(R+P)}$$

The use of the harmonic mean rather than the arithmetic mean is to emphasize the importance of small values.

Because the retrieval models or search engine (SE) produce a ranked list of documents, recall and precision can be computed at a particular rank to allow comparing different SE or ranking functions. So, for example, P@10 and R@10 are the precision and recall computed for the top 10 ranked documents respectively.

These two metrics can be measured at different ranks of the retrieval result and therefore we use p@5, r@5 to indicate the precision at rank 5 or the recall at rank 5. Another method to summarize the ranking is by computing the average-precision (AP) values from the rank positions where a relevant document was retrieved. This metric is based on the ranking of relevant documents, but its value depends on highly ranked relevant documents. So, it is mostly used when we want to evaluate which SE returns more relevant documents, among the top-ranked documents. However, it shows the effectiveness of a system ranking for a single query. A single query is not enough to evaluate and compare ranking algorithms. We should compute the average effectiveness over a whole set of queries. This average measure is called the Mean Average Precision (MAP). MAP is a single-figure measure of quality among all recall levels (Manning, 2008). MAP is the most

commonly used metric for system evaluation. It can be computed using equation 3.1 below:

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk}) \quad (3.1)$$

Where Q is the set of all queries and R_{jk} is the set of ranked retrieval results from the top result until reaching document d_k. A worked example is shown below:

Consider a set of two queries Q1 and Q2, each submitted to an retrieval system. The ranked retrieval result RR1 related to Q1 retrieved relevant documents at ranks: 1, 3, 6 and 10, while RR2 related to Q2 retrieved relevant documents at ranks: 3 and 15. The average precision AP1 for query Q1 is:

$$AP1 = \frac{1}{4}\left(\frac{1}{1}+\frac{2}{3}+\frac{3}{6}+\frac{4}{10}\right) = 0.641$$

As for the average precision AP2 for Q2:

$$AP2 = \frac{1}{2}\left(\frac{1}{3}+\frac{2}{15}\right) = 0.233$$

The mean average precision for the retrieval system is:

$$MAP = \frac{1}{2}\,(AP1 + AP2) = \frac{1}{2}\,(0.641 + 0.233) = 0.437$$

3.2.2 Binary Preference (bpref)

The MAP score is an effective and robust measure in case of complete judgments. Because the main interest in this research is to build the qrels automatically in the absence of judgments, or with an incomplete set of

judgments, we used other metrics which have been shown to be more robust than the MAP. These measures do not take into consideration whether documents are actually judged relevant by human assessors or whether they are merely assumed to be relevant, or assumed to be non-relevant because they were unjudged. One of the measures designed to deal with the problem of incomplete judgements is “bpref” (Buckley et al., 2004) which uses binary relevance judgments. This measure defines a preference relation, where any relevant document is preferred over any non-relevant document for a given topic. The formula used to compute the bpref is shown in equation 3.2.

$$bpref = \frac{1}{R} \sum_r 1 - \frac{|n\ ranked\ higher\ than\ r|}{R} \qquad (3.2)$$

Thus, for a topic with R relevant documents, bpref is defined over document pairs in which one member of the pair is not relevant and the other is relevant. The initial definition of bpref uses all pairs (r,n) where r is one of the known R relevant documents for the topic, and n is a member of the first R judged nonrelevant documents in the document ranking being evaluated. Since in practice the number of documents judged relevant is small, a variation on bpref is made. The bpref-10 "fix" to this problem is to compute the bpref-10 score using at least 10+R nonrelevant documents, as shown in equation 3.3.

$$bpref - 10 = \frac{1}{R} \sum_r 1 - \frac{|n\ ranked\ higher\ than\ r|}{10+R} \qquad (3.3)$$

3.2.3 Inferred Average Precision (infAP)

A variation to the AP that can work in the case of imperfect relevance judgments is the inferred AP (infAP) which was proposed by Yilmaz and Aslam (2008). This metric is shown to give better estimates of the gold standard average precision than the bpref metric suggested by Buckley and Voorhees (2004). The formula for computing the infAP is shown in equation 3.4 below:

$$E[precision\ at\ rank\ K] = \frac{1}{k}.1 + \frac{(k-1)}{k}\left(\frac{|d100|}{k-1}.\frac{|rel|+\in}{(|rel|+|nonrel|)+2\in}\right) \quad (3.4)$$

where d100 is the set of documents within the depth-100 pool. The "rel" are the documents among the depth-100 pool judged relevant, and the "nonrel" are the documents judged non-relevant in the pool. Since it is possible to have no documents sampled above the rank "k" which would cause division by zero, Lidstone smoothing (Chen and Goodman, 1998) was employed by using a small value $\in$ set to 0.00001 in the trec_eval[5] package used for evaluation.

3.2.4 Correlation metrics: Kendall's tau and Spearman coefficient

The precision and recall measure the effectiveness of the retrieval system or how well it performs in finding relevant documents at high ranks for the topic submitted. But when it comes to comparing several systems, it is usually done by comparing the rankings of the systems based on a certain measure. For example, after computing the MAP score for each of the IRS,

[5] http://trec.nist.gov/trec_eval/index.html

the systems are ranked in descending order of their MAP scores. When producing a new set of qrels, the MAP scores will change and therefore the ranking of the systems could also change. The generated qrels are considered reliable if they are able to produce a similar ranking as the one obtained using the human assessments. Measuring how close the two rankings are can be achieved using two metrics: Kendall's tau and the Spearman coefficient.

3.2.4.1 Kendall's tau

Kendall's tau (1945) is used to measure the similarity of the ordering of the ranked documents in two ranked sets. Given two documents d_j and d_k with their respective positions $s_{1,j}$ and $s_{1,k}$ in the ranked set R1 and $s_{2,j}$ and $s_{2,k}$ in the ranked set R2, we compute the difference between the two positions in each ranking $s_{1,k}$ - $s_{1,j}$ and $s_{2,k}$ - $s_{2,j}$. If the two differences have the same sign the pair of documents $[d_j,d_k]$ is called a concordant pair, otherwise it is called a discordant pair.

For a list of ranked documents, Kendall's tau is defined as following:

$$\tau(R1, R2) = P(R1 = R2) - P(R1 \neq R2)$$

Where P(R1=R2) is the proportion of the concordant pairs of documents and P($(R1 \neq R2)$ is the proportion of the discordant pairs of documents.

If we let $\Delta(R1, R2)$ be the number of discordant document pairs in R1 and R2, K be the size of the ranked set, then $K(K-1) - \Delta(R1, R2)$ will represent the

number of concordant document pairs in R1 and R2, therefore the Kendall's tau formula becomes:

$$\tau(R1, R2) = 1 - \frac{2 \times \Delta(R1,R2)}{K(K-1)} \quad (3.5)$$

The Kendall's tau value ranges between -1 and 1. A value of 1 indicates that all the document pairs are concordant. A -1 value means that all the document pairs are discordant. According to Voorhees (1998), a value greater than 0.8 for the Kendall's tau between the ranking of the retrieval systems is considered reliable.

3.2.4.2 Spearman coefficient

The Spearman coefficient (1904) is based on the differences between the positions of the same document in two different rankings. Consider 10 example documents retrieved by two distinct rankings R_1 and R_2. Let $s_{1,j}$ and $s_{2,j}$ be the document position in these two rankings, as follows:

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$	$(s_{1,j} - s_{2,j})^2$
d_{123}	1	2	-1	1
d_{84}	2	3	-1	1
d_{56}	3	1	+2	4
d_6	4	5	-1	1
d_8	5	4	+1	1
d_9	6	7	-1	1
d_{511}	7	8	-1	1
d_{129}	8	10	-2	4
d_{187}	9	6	+3	9
d_{25}	10	9	+1	1
Sum of Square Distances				24

Figure 3. 6: Example of documents with their position in two different rankings (Baeza-Yates and Ribeiro-Neto, 2011)

If $s_{1,j}$ is the position of document d_j in the ranking R1 and $s_{2,j}$ is the position of the same document d_j in the ranking R2, the Spearman coefficient is defined as:

$$S(R1, R2) = 1 - \frac{6 \times \sum_{j=1}^{K} (s_{1,j} - s_{2,j})^2}{K \times (K^2 - 1)} \qquad (3.6)$$

Where K is the size of the ranked sets R1 and R2.

The value of the Spearman coefficient is in the range [-1, +1] where the value of 1 indicates that the rankings are exactly the same, a value of -1 indicates that the rankings are the exact opposites of each other and a value of 0 indicates that the rankings are not related.

The Pearson correlation coefficient is a measure of the linear correlation between two variables X and Y, giving a value between +1 and −1 inclusive, where 1 is a total positive correlation, 0 is no correlation, and −1 is a total negative correlation. Pearson's correlation coefficient when applied to a population is commonly represented by letter Γ referred to as the population correlation coefficient or the population Pearson correlation coefficient:

$$\Gamma_{X,Y} = \frac{cov(X, Y)}{\sigma_X \sigma_Y}$$

where the cov(X,Y) is the covariance between X and Y and the σ_X is the standard deviation of X. When using ranked data, the Spearman and Person coefficients are identical.

Normally, if comparing a new system with the baseline, we would want the new system to be better. In this case, we want the rankings produced by the automatic techniques we proposed to form the qrels to be a replicate of the ones obtained from using the human-built qrels, or as close as possible. Therefore, we hope that the difference between them would not be significant (p-value > 0.05).

3.2.5 Intrinsic evaluation using Precision and Recall

In addition to the traditional evaluation for the retrieval systems, and measuring the correlations between the rankings to measure the reliability of the qrels, we propose an intrinsic evaluation for the generated qrels. The intrinsic evaluation is based on the precision and recall metrics. The known relevant documents are the ones judged relevant by human assessors. The qrels are ordered alphabetically by document ID. We computed the precision and recall measures for the generated qrels.

The formula used for the precision metric is:

$$Precision = \frac{d_{AH}}{d_A} \qquad (3.7)$$

Where d_{AH} is the number of documents judged relevant automatically using the new technique and by the human assessors and d_A is the number of documents judged relevant automatically by new technique.

As for the recall metric, the formula used is:

$$Recall = \frac{d_{AH}}{d_H} \qquad (3.8)$$

Where d_{AH} is also the number of documents judged relevant automatically using the new technique and by the human assessors and d_H is the number of documents judged relevant by human assessors. Such an evaluation provides more details about the average precision and recall for the automatically produced qrels when compared to the human-built ones.

3.3 Terrier IR Platform

Terrier (Ounis et al., 2006) is an open source search engine that implements state-of-the-art indexing and retrieval functionalities. It can be used to run information retrieval experiments. After indexing the test collection documents, the researcher can choose a retrieval model and use it to retrieve documents for each TREC topic. Then, the evaluation of the retrieval results could also be done automatically by simply pointing to the qrel set. It provides the list of metrics found in the trec_eval package, which is a tool developed by the researchers at NIST and used by the TREC community to evaluate the ad hoc retrieval results. It implements different evaluation metrics like precision, recall, MAP, infAP, bpref and others. All these metrics are described in section 3.6. During retrieval, several parameters can be tuned, such as determining which tags are to be processed in the topic and each of the documents of a TREC collection and which tags are to be skipped. The latest version of the Terrier platform supports several retrieval weighting models. New models were introduced, called field-based retrieval models because they do not only assign weights to terms in case they were

present in the document but they also take into consideration the frequency of that term and the field or tag in which it was found. If a term is found in the title of the document, this means that the document is more likely to be relevant to that term than if it occurred once in the body of the document. We used version 4.2 in our experiments. The table below summarizes the weighting and field-based weighting models.

Weighting-models	**Field-based Weighting models**
BM25, DFR_BM25, BB2, PL2, LGD, LF2, DLH, DLH13, IFB2, InL2, In_expB2, In_expC2, DFRee, DPH, Hiemstra_LM, Tf, TF_IDF, LemurTF_IDF, DFIC, DFIZ, DFReeKLIM, DirichletLM, InB2, Js.KLs, XSqrA_M	PL2F, BM25F, ML2, MDL2

Table 3. 4*: Terrier Platform retrieval models*

We used Terrier weighting retrieval models in the keyphrase extraction technique (KP Technique, chapter 4), and in the AQML technique described in chapter 6 when using the CLEF test collection since we did not have access to the initial runs as in the case of the TREC test collections. As for the other techniques, we used the TREC runs provided with the test collections in order to form the pool, select the documents for training and for forming the qrels.

3.4 Using automatic keyphrases as aspects to queries

To answer the first research question

Q1. Can we use keyphrases describing a topic as queries to retrieve more qrels?

which was based on the work done by Efron (2009), we proposed a methodology based on automatic keyphrase (KP) extraction from the documents. Each keyphrase was assigned a score which was computed depending on the distance between the keyphrase vector and the original topic vector. The keyphrases with the highest scores were then used as new queries or new aspects to the information need and submitted to the retrieval systems.

3.4.1 Tools Used

Before we explain the steps followed in this KP method, we will go over the tools we used in the experiments.

3.4.1.2 Keyphrase Extraction Algorithm (KEA)

The Keyphrase Extraction Algorithm, KEA (Witten et al., 2005) is an extraction algorithm based on the Naïve Bayes machine learning algorithm. The implementation is available online[6]. KEA uses a stemmer and a stopword list depending on the language used. The algorithm has to be trained first by creating a model to identify the keyphrases using documents where the keyphrases are actually known. The second step would be to extract the keyphrases from a new document using the trained model. The training and extraction processes are illustrated in Figure 3.7.

[6] http://www.nzdl.org/

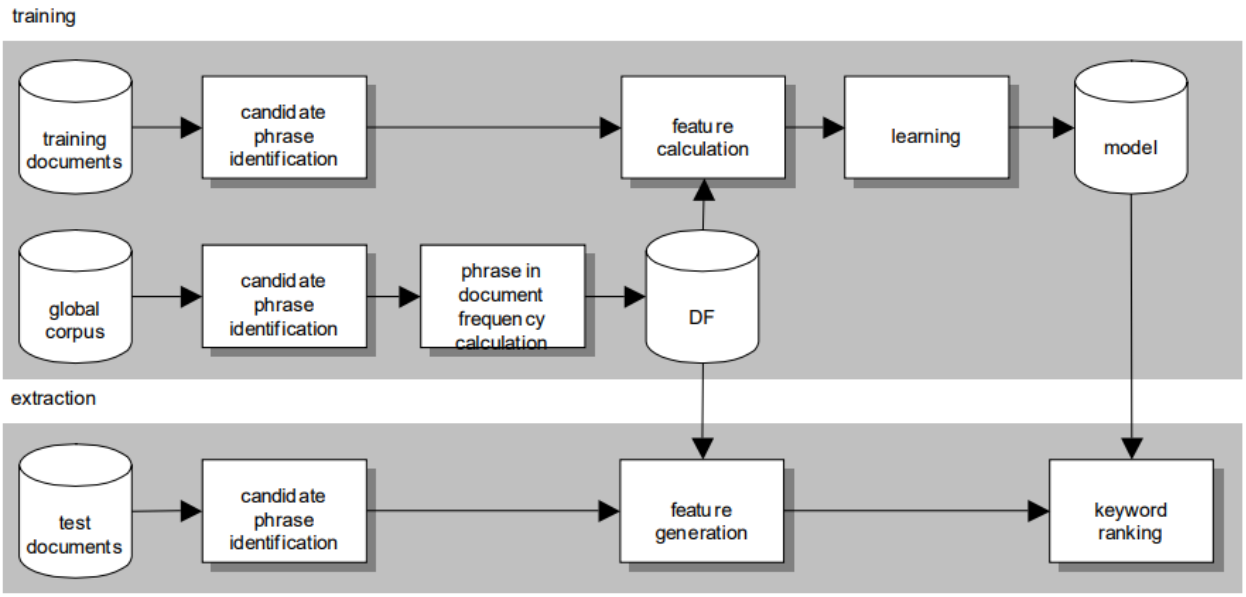


Figure 3. 7: Training and extraction process

To choose the candidate phrase, any punctuation, brackets or numbers are removed from the input text. The only tokens left are strictly alphabetic. Second, candidate phrases with limited length, usually set to three words, are selected. A candidate phrase cannot start or end with a stopword or a proper name. This step is called the phrase identification. Stemming and case-folding are then applied to the candidate phrases. For each candidate phrase, the tf.idf features are calculated. These features measure the frequency of the phrase in a document compared to how rarely it is used in the document collection as a whole, and the first occurrence of the phrase in the document, the distance or the number of words in the document required till reaching or that preceded the phrase. After training the model, the extraction begins, and the probability of the candidate phrases being keyphrases is calculated based on the computed features. The candidate

phrases are ranked according to their probability values, and the first "r" keyphrases are returned. We chose the KEA tool since it is language independent and it includes the option to be used without a controlled vocabulary, therefore choosing the keyphrases from the text.

3.4.2 Experimental Design

The experiments described in this thesis were tested on the TREC-7 and TREC-8 test collections. We submitted each of the topics to 12 Terrier retrieval weighting models. The top k documents retrieved by all weighting models for each topic are collected and considered as a relevant set (S) of documents for that topic, based on the hypothesis that if a document is retrieved by more than one system it is more likely to be relevant to the topic. Then, we extract keyphrases from these documents using the KEA algorithm. The number of keyphrases to extract from each document, and the number of tokens or words that each keyphrase consists of, are parameters which were set empirically for each test collection. A score was then computed for each keyphrase according to its similarity to the initial topic in question. Keyphrases with a score greater than or equal to an "ε" value are considered the new aspects to the initial TREC topic and will be submitted to the BM25 retrieval model. The union of all the documents retrieved for all the submitted keyphrases will be putatively relevant. The evaluation of the qrels was done by computing the correlation metrics Kendall's tau and Spearman rho between the TREC systems ranking

obtained using the human-built qrels and the ranking obtained using the newly generated qrels. The ranking of the systems was done based on the mean average precision (MAP) value. An algorithmic representation of the above described technique can be found below:

```
For each RetModel in the set of the 12 selected models
    Submit 50 TREC Topics
End For
Let ε be ← score threshold
For each query (q) of the 50 TREC Topics
    Let Aspects (q) be ← an empty set
    Let Sq be ← the set of the Top K documents retrieved by all models for (q)
    For each document (d) in Sq
        Extract x keyphrases from (d)
        For each Keyphrase KP in x
          Compute a score (KP) = degree_of_similarity (KP,q)
          If score (KP) >= ε
           Add KP to Aspects (q)
        End For
    End For
    Submit Aspects(q) to BM25
    Qrels q ← union of all results obtained from Aspects (q)
End For
```

KP Technique Pseudocode

The retrieval models, the parameters used in the experiments and the results obtained are detailed in Chapter 4.

3.5 Machine Learning techniques

In the methodology described in section 3.4, we had to determine a score threshold for each keyphrase and we could not standardize the number of keyphrases which should be extracted from the documents using different test collections, and the number of terms each keyphrase should consist of. Therefore, even though this technique seems to provide high correlations, we wanted to come up with a more solid technique which could be used on any test collection. The work done by Rajagopal et al. (2014) and Mollà (2013) inspired us to investigate how well the k-nearest neighbour algorithm would perform in generating the qrels. This idea led to expanding the work and testing different machine learning algorithms: unsupervised K-Means algorithm and supervised machine learning using the Naïve Bayes classifier and the Support Vector Machines in order to answer the second research question

Q2. Is it possible to use machine learning techniques to expand an initial set of presumed relevant documents and produce more qrels?

We describe the process for each of those approaches in sections 3.5.1 to 3.5.3.

3.5.1 K-Nearest neighbour

3.5.1.1 Introduction

The K-nearest neighbour (K-NN) algorithm is used for classification. It requires training samples as input. The output is a class membership. A test object is classified based on its distance from the majority of the "k" neighbours. When "k" is set to 1, the object is classified based on its closest neighbour. In the approach suggested in this thesis, we set k=1. We next describe the training set selection and the classification process we followed.

3.5.1.2 Experimental design

To apply the k-means algorithm, we needed a training set of documents for each of the topics. Since we wanted to make the technique fully automatic, our goal was to form this training set without human intervention. Because we could not be sure of the selected documents' relevance, we used a set that had a high probability of being relevant. In their work, Rajagopal et al. used a cutoff percentage of the number of occurrences of the documents in the pool and considered them relevant. We selected a higher cutoff percentage to support our hypothesis. The documents which are retrieved by several systems for the same topic are more likely to be relevant to that topic. Hence, we picked the documents which were retrieved by more than S% of the systems for each topic and we used them as training sets for the K-NN. The S% was defined as the minimum percentage of systems ensuring that each topic has at least one document selected in the training set.

After forming the training sets, we wanted to classify the remainder of the documents retrieved in the pool for each topic. In his paper, Mollà showed that relevant documents are at a close distance from each other. The cosine similarity measure between the vectors representing the documents was used. Therefore, for each topic, we measured the distance between the vectors of the document retrieved in the pool and each of the documents which was selected for training and presumed relevant to the topic. When the distance between that document vector and any of the vectors in the training set was less than a threshold "ε", the document was considered relevant to the topic, otherwise non-relevant. When the process was completed, we evaluated the list of qrels by computing the MAP scores for the retrieval systems using the newly generated qrels, ranking the systems and measuring the Kendall's tau and Spearman rho correlations between the initial ranking produced by the human relevance assessments and the one computed based on the automatically generated qrels. Our technique resulted in higher correlations than Rajagopal's et al., and had a lower false positive rate. The algorithmic representation for the K-NN technique is a combination of the Distance-Based pseudocode described in section 2.7.3.6 and Cutoff pseudocode detailed in section 2.7.3.7.

For each query (q) of the 50 TREC Topics

Collect the top 1000 retrieval results from all runs

P_q ← Pool with depth K =100

For each document (d) retrieved for (q)

Let occurrence (d) ← number of runs that retrieved (d) for (q)

Convert the occurrence(d) to a percentage of systems that retrieved (d)

End For

Order by occurrence(d) in descending order

End for

Let Cutoff be ← minimum percentage of systems so that (q) has at least one (d)

For each query (q) of the 50 TREC Topics

Let $Qrels_q$ ← empty set

Let R_q be ← set of documents where % occurrenc(d) >= Cutoff for topic (q)

Let ε be ← threshold for distance value

For each document (d) in P_q

Measure the distance between ($\vec{d}$) and each document vector in R_q

If distance <= ε

Add (d) to $Qrels_q$ ← consider relevant

End For

End For

Compute MAP for each run

Rank runs according to MAP

Compute Correlations

NN Pseudocode

3.5.2 Unsupervised K-means

3.5.2.1 Introduction

Clustering is the process of separating the documents given as input data without any labels into different groups or clusters according to some predefined criteria (Baeza-Yates, 2011). Since there is no information given about, or labelling of the input data, the clustering is called unsupervised. One of the most frequently used unsupervised algorithms is the K-means. The K-means clustering classifies the data into K clusters where each cluster is represented by a center point, called a centroid. The documents are distributed to the K clusters according to the distance of each document to the centroid of that cluster. The centroids are recomputed every time a new document is added to the cluster. The process is repeated until the centroids no longer change. The initial centroids are picked randomly at the start. It is possible though to select seed centroids to start each cluster and in that case the K-means becomes semi-supervised.

3.5.2.2 Experimental design

The experiments we designed include both unsupervised and semi-supervised implementations of K-means clustering.

3.5.2.2.1 Unsupervised K-means

Our goal of unsupervised learning was to obtain a set of 50 clusters for each of the 50 topics. Therefore, we grouped all the documents from the 50 pools for each topic and then we classified the documents into 50 clusters using

the unsupervised K-means where the centroids were initially picked randomly. Once the clusters were formed and the computed centroids no longer changed, we measured the cosine similarity between the vectors of the centroids of each cluster and the initial TREC topic. Since the number of centroid terms is much greater than the number of topic terms, we selected the "k" most frequent terms from the centroid for the distance computation. Unfortunately, this did not allow a successful match between each cluster and each topic. Several clusters' centroids were at a close distance from the same topic, while for some topics we could not find any matching cluster. Hence, we tried feeding the K-means algorithm the initial centroids instead of letting the algorithm pick them automatically.

3.5.2.2.2 Semi-supervised K-Means

Since the K-means algorithm provides the possibility to initialize the centroids with a known set of seed inputs, the optimal centroid seeds would be the centroid of a few known relevant qrels. Thus, we can expect that relevant documents would form a cluster for each topic since it was shown that relevant documents are at a close distance from each other (Mollà, 2013). Because we wanted to keep our technique automated without the use of actual qrels, we selected the centroid of the documents which have a high probability of being relevant, these being the documents retrieved by most of the systems. We then applied the K-means clustering with K set to 50 clusters. Only the documents classified and added to the clusters were

considered relevant to the topic. In both these experiments, we used the K-Means module implemented in the scikit-learn[7] Python library. The evaluation of the automatically generated qrels was done by computing the MAP scores for the TREC systems based on the new qrels, ranking the systems and then measuring the correlations between the gold standard ranking and the new ranking. Even though the correlations were positive, the MAP scores were very low. It was not possible to use an unsupervised classifier to group the documents into clusters that match exactly the 50 TREC topics, and since the semi-supervised algorithm provided positive correlations, we expanded the experiments to use supervised machine learning classifiers: Naïve Bayes and Support Vector Machines.

3.5.3 Supervised Machine Learning

3.5.3.1 Introduction

In a supervised machine learning environment, the classes or categories for classification are known. For each category, we have a set of known or labelled documents which we can use as training models for the classifiers. After the classifiers are trained and validated using the known data, the classifier is used to classify the remaining unlabelled documents. The documents are represented as vectors of terms weighted by tf.idf. Each term in the document's vector is called a feature, a separate variable. The classifier predicts the class to which each document should be assigned.

[7] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

3.5.3.2 Naïve Bayes classifier

In the case of a probabilistic classifier, a probability score is computed for each document-class pair according to the features in the document vector's representation. The highest probability score for the document-class pair determines the class which the document belongs to. In a Naïve Bayes classifier based on the classic probabilistic model, a document d_j is represented by a vector of binary weights indicating the presence or absence of a term as follows:

$$\vec{d}_j = (w_{1,j}, w_{2,j}, \dots\dots\dots\ w_{t,j})$$

Where $w_{i,j} = 1$, if the term "i" occurs in the document d_j and $w_{1,j} = 0$ otherwise. The classifier then assigns a score $S\left(d_j, c_p\right)$ which is the ratio:

$$S\left(d_j, c_p\right) = \frac{P\left(c_p \left|\vec{d}_j\right.\right)}{P\left(\bar{c}_p \left|\vec{d}_j\right.\right)} \quad (3.9)$$

Where $P\left(c_p \left|\vec{d}_j\right.\right)$ is the probability that the document d_j belongs to the class c_p and $P\left(\bar{c}_p \left|\vec{d}_j\right.\right)$ is the probability that the document d_j does not belong to the class c_p.

By applying the Bayes theorem, which is defined in the equation below, for two given events A and B:

$$P(A|B) = \frac{P(B\ |A\ ).P(A)}{P(B)}$$

- $P(A \mid B)$ is the conditional probability for the occurrence of A given than B is true.

- $P(B\,|A)$ is the conditional probability for the occurrence of B given than A is true.
- P(A) and P (B) are the marginal probabilities of A and B with the assumption that the two events A and B are independent.

We obtain:

$$S\left(d_j\,, c_p\right) \sim \frac{P(\vec{d}_j|c_p\,)}{P(\vec{d}_j|\bar{c}_p)} \qquad (3.10)$$

The estimate of the two probabilities is based on the naïve assumption that the terms in the document vector are independent of each other, the document is treated as a bag-of-words and therefore, each of the probabilities has the representation shown in the equations (3.11) and (3.12):

$$P\left(\vec{d}_j\middle|c_p\,\right) = \prod_{k_i \in \vec{d}_j} P\,(k_i\,, c_p\,) \times \prod_{k_i \notin \vec{d}_j} P\,(\overline{k_\iota}\,, c_p\,) \qquad (3.11)$$

$$\text{And } P\left(\vec{d}_j\middle|\bar{c}_p\right) = \prod_{k_i \in \vec{d}_j} P\,(k_i\,, \overline{c_p}\,) \times \prod_{k_i \notin \vec{d}_j} P\,(\overline{k_\iota}\,, \overline{c_p}\,) \qquad (3.12)$$

Where $P\left(k_i\,, c_p\right)$ and $P(\overline{k_\iota}\,, c_p)$ are the probabilities for the occurrence and non-occurrence of the term k_i in the document that belongs to the class c_p; and the $P\left(k_i\,, \bar{c}_p\right)$ and $P(\overline{k_\iota}\,, \bar{c}_p)$ are the probabilities for the occurrence and non-occurrence of the term k_i in the document that does not belong to the class c_p.

According to the equations, only the terms that occur in the document will contribute to the score computation since this model uses the binary

representation for the weights of the terms. This is known as the Binary Independence Naïve Bayes Classifier.

The term frequencies can actually improve the quality of the results when considered in the computation of the probability scores. This variation to the binary Naïve Bayes classifier is known as the Multinomial Naïve Bayes classifier which we use in our experiments. The equations which include the frequencies become:

$$P(\vec{d}_j|c_p) = F_j! \times \prod_{k_i \in d_j} \frac{[P(k_i|c_p)]^{f_{i,j}}}{f_{i,j}!} \quad (3.13)$$

Where $f_{i,j}$ is the frequency of the term i is the document j and F_j is the total number of terms in the document d_j; it measures the document length.

The term probabilities can be estimated from the training set as the following:

$$P(k_i|c_p) = \frac{\sum_{d_j \in D_t} f_{i,j} P(c_p|d_j)}{\sum_{\forall k_i} \sum_{d_j \in D_t} f_{i,j} P(c_p|d_j)} \quad (3.14)$$

Where D_t is the set of documents used for training. To compute the probability that the document d_j belongs to the class c_p, we apply Bayes theorem:

$$P(c_p | \vec{d_j}) = \frac{P(c_p) \times P(\vec{d_j}|c_p)}{P(\vec{d_j})} \quad (3.15)$$

The prior class probability can be computed as follows:

$$P(c_p) = \frac{\sum_{d_j \in D_t} P(c_p \mid d_j)}{N_t} = \frac{n_p}{N_t} \qquad (3.16)$$

Where N_t is the size of the training set.

The prior document probability can be computed using the equation in (3.17):

$$P(\vec{d}_j) = \sum_{p=1}^{L} P_{prior}(\overrightarrow{d_j} \mid c_p) \times P(c_p) \quad (3.17)$$

Where L is the total number of classes. With the substitution of the equations 3.16, 3.17 and 3.13 in equation 3.15, the classifier estimates the probability for each document-class pair and assigns the document to the class with the highest probability value. All probabilities have the value in the range of {0,1}.

3.5.3.3 Support Vector Machines (SVM)

SVM classifiers were introduced by Cortes and Vapnik (1995). They are a set of methods which allow data to be classified by finding a hyperplane that separates the elements of two classes. The elements or documents belonging to a class A will be grouped in one region, and the documents belonging to class B will be grouped in another region. The SVM finds the best hyperplane that separates the two regions in a way that a new document will be classified based on its distance to the hyperplane previously learnt. The documents are represented as vectors or points in a

two-dimensional space and therefore the hyperplane will be a line as shown in figure 3.8.

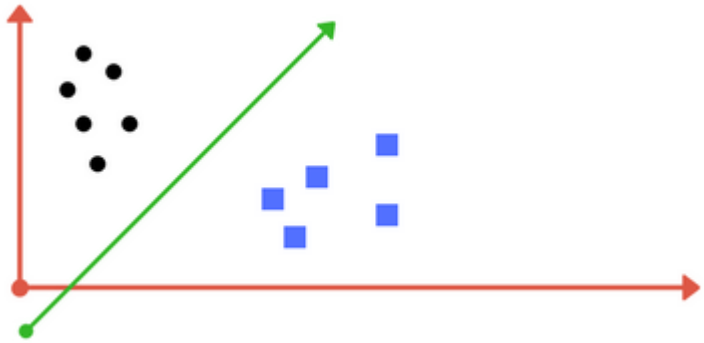


Figure 3. 8: Documents belonging to two different classes and the hyperplane learnt to separate between them[8].

There could be several hyperplanes which separate the classes, however the classifier looks for the best one which maximizes the margin or the distance to the nearest point. Thus, finding the hyperplane becomes more difficult, such as for the data shown in figure 3.9.

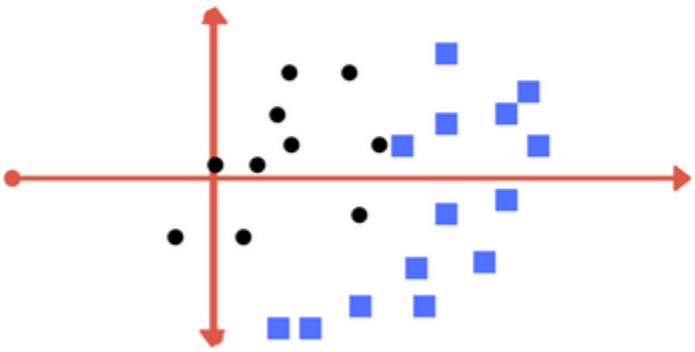


Figure 3. 9: Regions for the two classes are not clearly divided

[8] Figures 3.8, 3.9 and 3.10 have the following reference: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

The optimal hyperplane could be linear or non-linear (as shown in figure 3.10) and to learn this hyperplane, the SVM uses kernel functions which take a low dimension input space which is not easily separable and converts it to a high dimension space which is separable linearly. There are linear kernels, polynomial kernels and Radial Basis Function (RBF) kernels. In our experiments, we used the SVM with linear kernels, the simplest form of SVM and the one which is usually recommended for text classification.

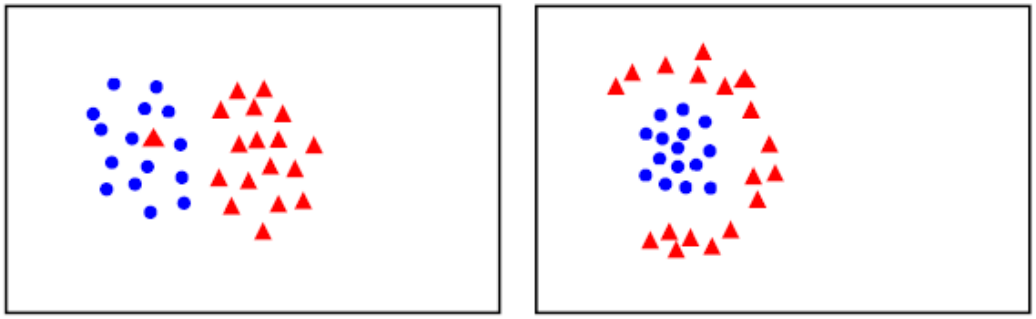


Figure 3. 10: A non-linearly separable data

The linear classifier has the form: $f(x) = w^{\top}x + b$ where w is the normal to the line, also known as the weight vector and b is the bias. The training data is used to learn "w" which will be used for classifying the new data, in our case, the new documents.

If we consider:

- H_w: a hyperplane that separates documents in classes c_a and c_b
- m_a: distance of H_w to the closest document in class c_a
- m_b: distance of H_w to the closest document in class c_b

- $m_a + m_b$: margin m of the SVM

The decision hyperplane is the one which maximizes the margin m. Given a training set

T={… $[c_j, \vec{z_j}]$ …} where c_j is the class associated with a point Z that has a vector representation $\vec{z_j}$, the SVM optimization problem which maximizes m where $m = \frac{2}{|\vec{w}|}$ is subject to the following:

$$\vec{w}\vec{z_j} + b \geq +1 \; if \; c_j = c_a$$

$$\vec{w}\vec{z_j} + b \leq -1 \; if \; c_j = c_b$$

The vectors $\vec{w}$ which make the equations equal to either 1 or -1 are referred to as support vectors. The classification of a document d_j represented by a vector $\vec{z_j}$ is decided by:

$$f(\vec{z_j}) = sign(\vec{w}\vec{z_j} + b)$$

If $f(\vec{z_j})$ has a positive sign, this means that d_j belongs to class c_a, otherwise it belongs to class c_b.

3.5.3.4 Experimental design

With supervised machine learning algorithms, the classifiers are given a set of training data as input. In our experiments, we used two approaches with supervised classifiers. Since the aim of our work is to generate the set of qrels without human intervention, the training set was also selected

automatically. In a first approach of this ML technique, the documents which formed the pool for each topic were classified as either relevant or non-relevant. Therefore, the number of categories or classes used was two. We refer to this approach as the **two-class ML.** For each of the classes, a set of documents was automatically selected as a training set. The selection process went as described in the steps below, for each of the 50 topics:

1. For each document in the pool, count the number of systems which retrieved that document for the topic. This number represents the occurrences of each document.
2. Order the documents in decreasing order of their number of occurrences.
3. Select the documents which were retrieved by more than S% of the systems as the training set for the relevant class. "S%" is defined as the minimum number of system required to ensure that each topic has at least one document selected as relevant.
4. Select the same number of the documents which had the lowest number of occurrences and use them as the training set for the non-relevant class.

The selection in step (3) gives a set with a high probability of being relevant, while the documents selected in step (4) have a high probability of being non-relevant. We then train the supervised machine learning classifier. We repeated the experiments twice, first using the Multinomial Naïve Bayes (NB)

classifier and secondly using an SVM. After the classifiers had been trained, we classified the remainder of the documents in the pool of each topic into one of the two classes: relevant (1) or non-relevant (0). The process was repeated for all the topics. The two-class ML algorithmic representation is described below:

//Step 1. Count Occurrences

For each query (q) of the 50 TREC Topics

Collect the top 1000 retrieval results from all runs

P_q ← Pool with depth K =100

For each document (d) retrieved for (q)

Let occurrence (d) ← number of runs that retrieved (d) for (q)

Convert the occurrence(d) to a percentage of systems that retrieved (d)

End For

OS_q ← Set of documents after ordering by occurrence(d) in descending order

End For

Let Cutoff be ← minimum percentage of systems so that each (q) has at least one (d)

//Step 2. Form Training sets

For each query (q) of the 50 TREC Topics

Let R_q ← empty training set for relevant documents

Let NR_q ← empty training set for non-relevant documents

Count ← 0

For each document (d) in OS_q

If occurrence% (d) >=Cutoff

Add (d) to R_q

Count ← Count + 1

End for

Start from the end of OS_q

While count > 0

Add (d) to NR_q

Count ← Count - 1

End while

//Step 3. Train classifiers

Train NB classifier (OR SVM) using R_q *and* NR_q

//Step 4. Generate qrels

CS_q *← Classify each document in the set* $P_q - (R_q + NR_q)$ *as either relevant or non-relevant*

$Qrels_q \leftarrow CS_q + R_q + NR_q$

End For

Two-Class ML Pseudocode

Once the classification was completed for all the topics, we obtained a list of automatically generated qrels which we evaluated using both intrinsic and extrinsic evaluations. The intrinsic evaluation was based on computing the precision and recall for the generated qrels and comparing them to the gold standard, the human-built qrels. The extrinsic evaluation was done by ranking all the retrieval systems which participated in building the automatic qrels and measuring the correlations between the ranking produced with the automatic qrels and the one obtained from the real or actual qrels based on human assessors. The details of the evaluation metrics are provided in section 3.6.

The second approach which we refer to as the 50-class ML approach, also involved using the supervised machine learning classifiers NB and SVM and it was conducted as follows:

1. For each document in the pool, count the number of systems which retrieved it for the same topic. This number represents the occurrences of each document.
2. For each topic, select the documents which were retrieved by more than S% of the systems.
3. The documents selected in step 2 are then grouped into a class that has the topic number as a label. For example, if for the topic with number 401, the documents selected were doc1, doc16 and doc25, we create a class with a label 401 and we label doc1, doc16 and doc25 with the value 401.
4. The classification problem here requires the classifier to assign a document to one of the 50 classes where each class represents a TREC topic.

After the classifiers NB and SVM were trained using the training sets for the 50 topics, we put together in one pool all the documents that were retrieved for all the topics excluding the ones used for training. The trained classifiers then predict the label or the topic number of each document in the pool. We strictly used the documents in the pools to perform the classification, since we know their actual relevance and therefore we can better evaluate the obtained qrels. When the classification was completed, all the documents

which were labelled with a topic number were considered relevant to the topic, while all the remaining retrieved documents for the topic were considered non-relevant. The algorithmic representation for the 50-class ML approach is defined below:

```
//Step 1. Count Occurrences: Same as two-class ML, page 132
//Step 2. Form Training sets
For each query (q) of the 50 TREC Topics
    Let Rq ← empty set
    For each document (d) in OSq
      If occurrence% (d) >=Cutoff
            Add (d) to Rq
     End for
End for
//Step 3. Train classifiers
Train NB classifier (OR SVM) over 50 classes {R1, R2, R3…R50}
//Step 4. Generate qrels
CS ← Classify each document in all Pi – {R1 ... R50} into one of the Ri sets
Qrels q ← CSq + Rq
```

50-Class ML Pseudocode

This process led to generating a set of automatic qrels and the same intrinsic and extrinsic evaluation processes were followed with this approach as well. The correlations we achieved with the two approaches outperformed the existing methods which attempted to rank the retrieval systems in the absence of or with an incomplete set of qrels.

3.6 Conclusion

In this methodology chapter, we described the different techniques we implemented to answer each of the research questions. The KP technique described in section 3.4 uses keyphrases to describe a TREC topic in alternating ways and was implemented to answer the research question Q1 (see chapter 4). In section 3.5, we explained how the nearest neighbour, the K-means and the supervised machine learning techniques were deployed to answer the second research question. Each of those techniques was shown to provide better results than the previous existing approaches but still failed to discriminate between the best systems. This is why we had to use a small number of actual known qrels or relevant documents as seeds to start the process of building the pseudo-qrels and answer research question Q3 (see chapter 6). In this last set of experiments, we were dealing with an incomplete judgments list and therefore we investigated how good the correlations using the bpref and infAP measures would be when compared to the ones obtained using the MAP scores, answering research question Q4 (see chapter 6). Because this last variation seemed to work best on English test collections, we tested its performance with other languages as raised by research question Q5 (see chapter 6).

The evaluation of the results for each of the techniques proposed to answer the research questions was based on measuring the correlations between the overall rankings or sub rankings of the systems which are discussed in

details in chapters 4, 5 and 6. We also performed an intrinsic evaluation by measuring the precision and recall for the generated qrels. Finally, a statistical test was conducted to interpret the significance of the difference obtained in the correlations results. We also discussed how the hard topics could have affected the results since they only have a few relevant documents.

Chapter 4 - Using automatic keyphrases as aspects of queries

4.1 Introduction

In this chapter, we present our first method called the KP technique to automatically produce the set of qrels for a test collection based on keyphrase (KP) extraction from documents. It is an attempt to answer the first research question of this thesis:

Q1. Can we use keyphrases describing a topic as queries to retrieve more qrels?

This method is inspired by previous work done by Efron (2009) in which he showed how an information need can be expressed through different queries which he named "aspects". An example of an aspect is given in section 4.2. He then showed how using these aspects can result in pseudo-qrels that produced similar system rankings to the human-built ones. Human effort was required, though, to develop these aspects. In an attempt to reduce this effort, we thought of extracting keyphrases (KPs) automatically from documents considered presumably relevant because a relevant document contains relevant information and therefore it could lead to finding more relevant documents and then using them as queries to retrieve more relevant documents. The full process used by Efron to generate the pseudo-qrels, the experiments and the evaluation he performed are detailed in

section 4.2. As for our technique, we selected a set of documents with a high probability of being relevant to a TREC topic, then we used the Keyphrase Algorithm Extraction (KEA) open source tool to extract keyphrases from these documents and then based on a score computed for each keyphrase, we selected the most suitable to be new queries for the TREC topic. We submitted the new queries to the BM25 model implemented in Terrier and then pooled the documents obtained from all the queries to form the pseudo-qrels. The process and an overview of the different tools used are given in section 4.3. In section 4.4, we evaluate the outcomes of the KP technique. In section 4.5 we report the results we obtained when using a dataset with keyphrases manually annotated in order to examine whether the human-generated query aspects produced were more effective than the automatically generated ones. and then in section 4.6 we provide a conclusion for the chapter.

4.2 Origin of the Work

An information need can be expressed using any of several different queries. When one of these queries is submitted to an information retrieval system, a ranked list of documents is returned as an answer to the query. The union of the top returned documents for all the queries can be considered as the query-based relevance set (qrels) for the initial information need. This assumption was tested by Efron (2009) through the experiments he conducted using TREC test collections which we described in the previous

chapter. Each of the TREC test collections has a set of 50 topics. For each topic, Efron (2009) developed a number of aspects for the topic. An aspect was described by Efron as follows: consider TREC topic 402 that has "behavioral genetics" as its title, an aspect of the topic could be "behavioral disorders" or "genetics addictions". The person formulating the aspect did some research about the initial information need. Four aspects were developed for each topic. Two aspects were made manually by the researchers and the remaining two were developed automatically using query expansion of the manual ones previously created.

Query expansion is a process which involves reformulating the initial query submitted to an information retrieval system or expanding it by adding terms which seem relevant to the topic. It can be performed either manually or automatically. Manual query expansion requires the user to select additional terms or phrases which are related to his information need and which are added to the query resulting in a new one. The terms and phrases are suggested by the information retrieval system and are usually synonyms of the terms which were initially used in the query (Manning et al., 2008). Automatic query expansion does not require the user to select the terms to expand the query, but rather extracts the terms from the top k ranked documents retrieved for the query, which are presumed relevant, and then computes the weights for the terms, using the tf.idf weighting with the vector space model, or their probability if the weighting model used for

retrieval was probabilistic, and then adds the "x" terms with the highest weight to the initial query.

Efron used automatic query expansion. The four resulting aspects were then submitted as queries to the Okapi BM25 retrieval system and the union of the top 100 documents retrieved for all four aspects were considered putatively relevant, forming the pseudo-qrels automatically without human effort. Although the qrels formed from aspects did not require human intervention, there was still some effort required to build the aspects of queries for the TREC set of topics. The only manual effort required in Efron's technique was building the manual aspects, which was not very easy as it required conducting research about the topic in question and forming aspects that could relate to the original query. That motivated us to devise a new technique that could reduce the human effort in forming the aspects as the results of the obtained qrels from Efrons' technique seemed really encouraging. Thus, we proposed using keyphrases (KPs) extracted from documents as new queries as surrogates for the manual aspects which were developed for the initial TREC topic. The detailed steps we followed in the experiments are described next.

4.3 Experiments using KPs

4.3.1 Keyphrase Extraction Algorithm (KEA)

The Keyphrase Extraction Algorithm (KEA) (Witten et al., 2005), which was discussed in detail in section 3.4.1.2, is an open source tool which allows the extraction of phrases consisting of several words or tokens. The accuracy of the keyphrases extracted automatically by the KEA was 99% when compared to a set of manually extracted keyphrases according to Witten et al. (2005) when the training set consisted of 20 to 50 documents and the number of KPs to extract was set between 5 and 15. The extraction process is a two-step procedure that starts with building the training model, defining the set of parameters and then applying the model to the list of documents we wish to extract the KPs from. The parameters which can be tuned when building the model and then when extracting the KPs are defined in table 4.1 below along with their default values and the values we have used or tested with.

Parameters defined when *building the model from the training set*		
Name	**Default value**	**Value(s) used**
y: minimum phrase length	1	2 and 3
x: maximum phrase length	5	3 and 5
o: minimum number of occurrences	2	1

t: stemmer	SremovalStemmer (Lovins Stemmer)	Kept the default
Parameter defined when *extracting the keyphrases from the input documents*		
n: the number of keyphrases to be outputted	5	5, 10, 15, 20 and 25

Table 4. 1: List of parameters defined for KEA

The minimum and maximum number of tokens in a keyphrase were chosen according to the original TREC topic size. Most of the topics had 3 to 5 words in TREC-8 for example, so we maintained the same length for the keyphrases. Increasing the minimum number of occurrences to more than 1 will not result in sufficient number of extracted keyphrases because we can rarely find the same keyphrase with all 5 words repeated in a document. In the coming two sections, we explain how these parameters were set, how the training models were built, and which documents were used for both the training and extraction phases.

4.3.2 Using Terrier retrieval models

The aim of the approach we propose is to automatically generate the pseudo-qrels with a minimum of human intervention and if possible without any. Therefore, the selection of the documents from which we could extract KPs is essential because those KPs should hold some relevant information about the topic or the information need in question and subsequently lead to more relevant information. Due to the assumption made initially by Soborrof

et al. (2001) which states that a document retrieved by several information retrieval systems is most likely to be relevant to the topic, we used the top k documents retrieved by a set of retrieval models. We could not use the TREC runs to get these documents, because it is very rare to find one or more documents retrieved by 100% of the runs, thus we deployed the retrieval models implemented in the Terrier platform and we selected twelve of the implemented retrieval models listed in section 3.3 to retrieve documents for the 50 topics in each TREC test collection. We used both probabilistic and vector space models for the retrieval: BB2, BM25, DFR_BM25, DLH, DLH13, IFB2, In_expB2, In_expC2, LemurTF_IDF, LGD, PL2, TF_IDF.

4.3.3 Building the training models

The best scenario for building a training model for extracting KPs from documents would be using a set of documents which have their KPs manually assigned by a human annotator. Since we did not have this set, we built the training models automatically. After running all twelve Terrier retrieval models, we selected the top document retrieved by each of these models for each topic and we automatically created for it a set of KPs: the title of the TREC topic in full, then the set of bigrams and unigrams in the title field. The document set resulting from all the 50 topics was used as a training set for building the training models and it contained around 300 documents. We built three different models: a first model that had a

minimum of two tokens in the KP and a maximum of three, a second which had the minimum set to two, the maximum to five and in the third one the minimum number of words was defined as three with the maximum kept to five. The choice of the minimum and maximum values was not done randomly. We used TREC-7 and TREC-8 test collections for running the experiments so we could compare our results with Efron's. If we observe the topic titles for each of those collections, we notice that the range of the number of words in the topic title is between two and five, but a few of them have only one word in the topic title. For all the three training models, the minimum number of occurrences of each KP was set to 1.

4.3.4 Applying the extraction model

After building the training models, we extracted the KPs from a set of presumably relevant documents because we hypothesized that these KPs could hold some additional relevant information to the topic and therefore they could lead to finding more relevant documents. We applied the algorithm described in section 3.4.2; the different values fed to the algorithm are detailed here in this section. As a start, we used the top k documents retrieved by all twelve Terrier retrieval models listed in section 4.3.2 above. These documents have a high probability of being relevant to the topic. We experimented with different values for K: 5, 10, 15 and 20 documents. These documents without the ones used for building the training model formed the set S_q defined in the algorithm. Another parameter that

was provided was "x", the number of keyphrases to extract from each document. We tested the extraction of 5, 10, 15, 20 and 25 KPs per document. In the case where a text document was too short to have 10 or more KPs extracted, the algorithm would not extract more than the possible number that can be found in that document. Therefore, 20 and 25 KPs could be extracted only from long documents. The steps we explained are graphically represented in diagram 4.1 below with the example of using the top 10 common documents retrieved by all Terrier models.

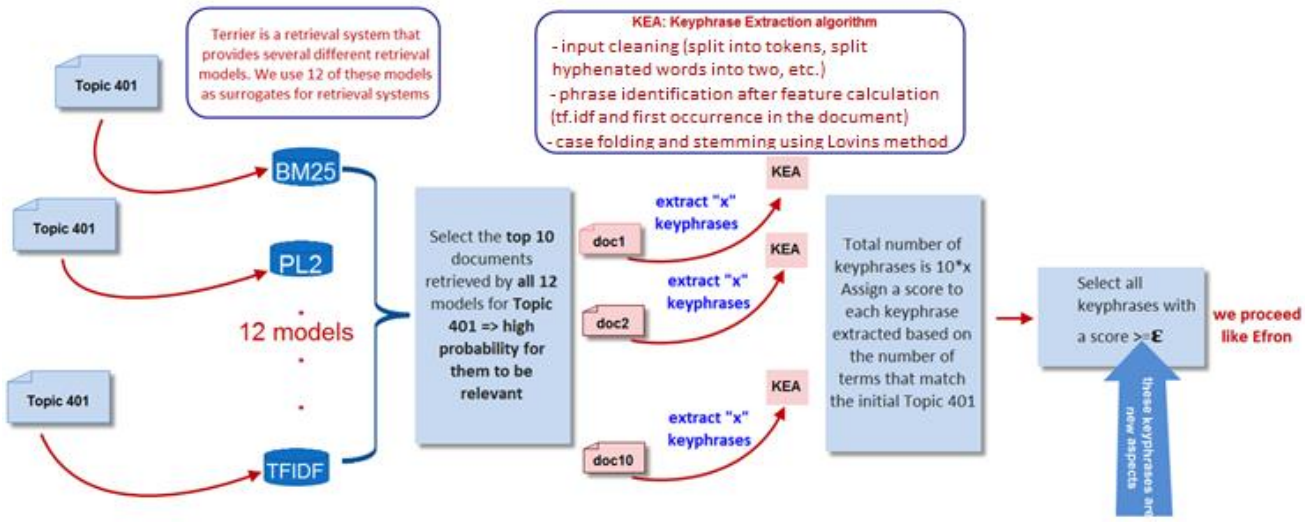


Figure 4. 1: Keyphrases based approach diagram

The combination of the parameters and models we tested on both TREC-7 and TREC-8 is shown in figure 4.2.

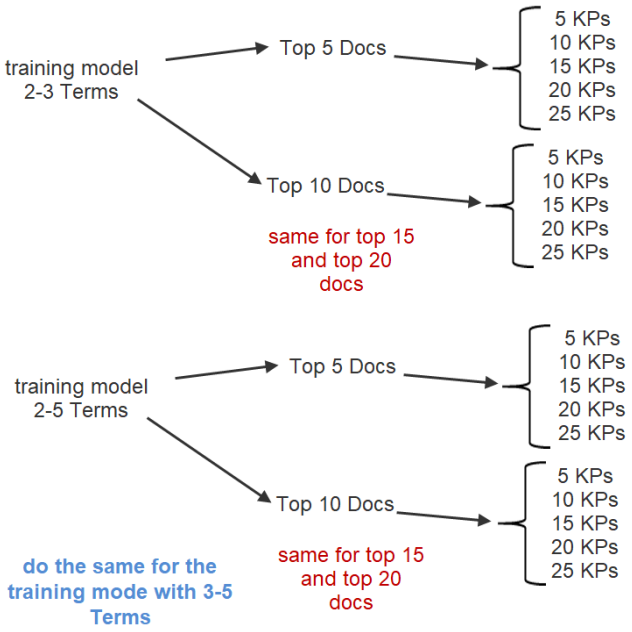


Figure 4. 2: Different combinations for the parameters and the top "K" documents

4.3.5 Potential queries from keyphrases

Having extracted the keyphrases with a certain number of terms, from a given number of documents, the next step is to select which of the KPs can be considered as a new query to the TREC topic. We thus assign a score for each KP which is defined as *degree_of_similarity (KP,q)* in the KP pseudocode. The score for each KP is computed as the number of matching terms between the KP and the initial TREC topic title and description fields. For each TREC topic, the title field represents a short query while the description field provides more explanation about the topic or what a relevant document could include. For example, for TREC-8, the title field for topic 401 is: foreign minorities, Germany while the description field is: What language and cultural differences impede the integration of foreign

minorities in Germany? We ignored the stopwords when calculating the scores for each KP. We tried to compute the score in two different ways: first, by counting each term either in the title or in the description field, so if the term was found in either the title or the description we count it as one; the second was to check for each term in both the title and the description fields and therefore if it existed in both, its count would be two. This second score computation approach resulted in forming qrels that gave higher correlations between the KP system rankings and the TREC standard rankings as reported in tables 4.3 and 4.4 this could be due to the fact that if a term is found in both the title and the description, it is more important in the query than the term that appears once in either one of them, and thus assigning to it more weight, would increase the score of the KP which has it, therefore increasing its probability to be selected as a potential query. While the correlations between the rankings obtained based on the MAP scores computed from the KP pseudo-qrels and the TREC rankings based on the first approach to compute the scores for the KPs were slightly better in most cases, their highest value was still less than the best one we found when using the second score computation. In the above experiments, each KP could have at most 5 terms leading to a maximum score of 5 if all the terms in the KP could be found in the title and the description fields combined. So, the value for the score threshold ε was determined as follows. If there were no KPs with score 5, we looked for KPs with a score 4, if not for a score 3,

then 2 then 1. A KP with only one matching word could lead to some noise in the results, but we had to keep it since the KPs extracted for a few topics did not have more than one matching term. We tried to use a semantic matching technique between the extracted KPs and the original queries, using Wordnet synset[9] and then a greedy algorithm that compares each word in the KP with every query's word's synset but the results were not encouraging, so we decided not to report them. In table 4.2 below, we show an example of some of the KPs extracted for TREC-7 and TREC-8 topics.

Initial TREC topic title	**Keyphrases selected as new queries**
TREC-7	
Topic 351: Falkland petroleum exploration	- falkland islands territorial - falkland islands councillor - ownership of the Falkland islands
Topic 400: Amazon rain forest	- south americas amazon
	- americas amazon basin
	- greatest rain forest
TREC-8	
Topic 404: Ireland, peace talks	- peace and a political settlement - longterm intentions in Ireland - northern ireland talks
Topic 446: tourists, violence	- ensure tourists safety - attack on tourists - acts of violence

Table 4. 2: Example of keyphrases extracted for TREC topics

[9] http://www.nltk.org/howto/wordnet.html

For each topic, the candidate new queries were then submitted to the Terrier BM25 retrieval model. The retrieval result obtained was a ranked list of 1000 documents, where the top retrieved document was considered the most relevant to the query. We use the term qrels to refer to the documents judged by human assessors while the phrase pseudo-qrels refers to those automatically generated by the expansion of qrels. To form the pseudo-qrels for each topic, we proceeded in a similar manner to Efron (2009). We defined the pseudo-qrels for each topic as the union of the top 100 documents retrieved for all the candidate queries we submitted.

4.4 Evaluation of results

The evaluation process of the automatically generated qrels using the KP extraction technique was the same as the one adopted by Efron. In his experiments, Efron evaluated the automatic retrieval runs for TREC-7 and TREC-8. The automatic runs are the runs with retrieval results obtained from using a query automatically built without any human effort (Voorhees et al., 1997). How the query was built by each system is not detailed in the overview provided by NIST for each test collection. According to the description of each test collection, there are 86 automatic runs for TREC-7 and 116 for TREC-8. For each of the TREC runs, we computed the Mean Average Precision (MAP) score using the human-built qrels and then we computed a new MAP score using the newly generated qrels. Afterwards, the systems were ranked according to their two computed MAP scores and the

two rankings were compared using the correlation metrics Kendall's tau and Spearman coefficient.

The complete Kendall's tau (τ) and Spearman's rho (ρ) correlations obtained with the different set of parameters we tested are given in table 4.3 for TREC-7 and table 4.4 for TREC-8.

Extraction Models applied	Number of KPs extracted	Top K documents used for extraction							
		5		10		15		20	
		τ	ρ	τ	ρ	τ	ρ	τ	ρ
2-3 Terms	5	0.5595	0.7600	0.5266	0.7316	0.5189	0.7254	0.5291	0.7354
	10	0.5622	0.7604	0.5174	0.7183	0.5050	0.7107	0.5111	0.7140
	15	0.5524	0.7446	0.5163	0.7158	0.5070	0.7062	0.4975	0.6963
	20	0.5331	0.7217	0.4875	0.6792	0.4902	0.6815	0.4826	0.6724
	25	0.5301	0.7221	0.4740	0.6631	0.4830	0.6737	0.4822	0.6707
2-5 Terms	5	0.5159	0.7200	0.4947	0.6886	0.5080	0.7117	0.5304	0.7407
	10	0.5561	0.7567	0.5448	0.7469	0.5750	0.7813	0.5745	0.7789
	15	0.5539	0.7581	0.5786	0.7828	0.5566	0.7642	0.5551	0.7646
	20	0.5311	0.7320	0.5616	0.7645	0.5541	0.7612	0.5507	0.7550
	25	0.5343	0.7338	0.5541	0.7594	0.5572	0.7656	0.5593	0.7656
3-5 Terms	5	0.5377	0.7309	0.4761	0.6689	0.5302	0.7349	0.5394	0.7454
	10	**0.6131**	**0.8144**	0.5824	0.7856	0.5790	0.7845	0.5574	0.7641
	15	0.5995	0.7947	0.5729	0.7726	0.5258	0.7270	0.5222	0.7204
	20	0.6027	0.7994	0.5509	0.7498	0.5325	0.7328	0.5275	0.7237
	25	0.5536	0.7527	0.5266	0.7185	0.5034	0.6977	0.5177	0.7097

Table 4. 3: TREC-7 Kendall's tau (τ) and Spearman's rho (ρ) correlations based on MAP scores between TREC rankings and rankings obtained using pseudo-qrels produced with different KEA parameter combinations

The extraction model that includes extracting 3 to 5 terms for each KP provides the best Kendall's tau and Spearman's rho values when applied to

the top 5, 10 and 15 documents, while the model with fewer terms, the 2 to 5 model, has the highest correlations when the top 20 documents are used for the KP extraction. These results are very similar over all the models, where the Kendall's tau value is above 0.5377 and reaches its maximum of 0.6131 with the 3 to 5 model applied to the top 5 most common documents to extract 10 KPs. The Spearman's rho values range between 0.7527 and 0.8144. Correlations outside these ranges are observed in a few cases, when extracting 20 or 25 KPs from more than 10 common documents using the model built with 2 to 3 terms and then again when extracting 5 KPs with the remaining two models. The reason for such low correlations in the first case could be interpreted as follows: extracting too many KPs which do not add any new information might lead to more noise in the results rather than enhancing the correlations. As for the last case, the cause could be the lack of information extracted from the documents and therefore for some of the topics not many new queries were formulated.

Looking at table 4.4, we can see that the only model that could provide a Kendall's tau above 0.5 and a Spearman's rho above 0.7 was the one using 3 to 5 terms while all the others showed low correlations. The reason for these values could be related to the fact that KEA works best when the indexing is done using a controlled vocabulary as mentioned by the authors (Witten et al., 2005). But with TREC test collections, we could not provide a vocabulary for indexing since we were not working in a specific domain. We

were dealing with free text extracted from newspaper articles about a variety of topics.

Extraction Models applied	Number of KPs extracted	Top K documents used for extraction							
		5		10		15		20	
		τ	ρ	τ	P	τ	ρ	τ	ρ
2-3 Terms	5	0.4618	0.6344	0.4826	0.6564	0.4773	0.6541	0.4723	0.6501
	10	0.4673	0.6377	0.4623	0.6349	0.4567	0.6333	0.4434	0.6212
	15	0.4671	0.6368	0.4296	0.6017	0.4430	0.6204	0.4461	0.6231
	20	0.4563	0.6277	0.4377	0.6127	0.4569	0.6361	0.4574	0.6371
	25	0.4575	0.6282	0.4419	0.6153	0.4600	0.6407	0.4605	0.6432
2-5 Terms	5	0.4782	0.6523	0.4840	0.6599	0.4861	0.6636	0.4794	0.6581
	10	0.4819	0.6544	0.4870	0.6601	0.4967	0.6773	0.4925	0.6751
	15	0.4876	0.6627	0.4534	0.6287	0.4594	0.6392	0.4662	0.6447
	20	0.4685	0.6398	0.4299	0.6006	0.4527	0.6278	0.4658	0.6447
	25	0.4648	0.6375	0.4315	0.6030	0.4514	0.6283	0.4685	0.6475
3-5 Terms	5	0.5374	0.7193	**0.5527**	**0.7300**	0.5278	0.7166	0.5167	0.7050
	10	0.5316	0.7114	0.4923	0.6753	0.5111	0.6977	0.5077	0.6956
	15	0.5218	0.7005	0.4780	0.6637	0.4870	0.6721	0.5000	0.6819
	20	0.5167	0.6948	0.4851	0.6697	0.4979	0.6821	0.4967	0.6813
	25	0.5144	0.6906	0.4880	0.6720	0.4978	0.6818	0.5016	0.6850

Table 4. 4: TREC-8 Kendall's tau (τ) and Spearman's rho (ρ) correlations based on MAP scores between TREC rankings and rankings obtained using pseudo-qrels produced with different KEA parameter combinations

Table 4.5 compares the best Spearman's rho and Kendall's tau values we obtained using the keyphrase extraction technique and the correlations obtained by Efron.

	Efron's aspects qrels		Keyphrases generated qrels	
Test Collection	**Kendall's tau**	**Spearman**	**Kendall's tau**	**Spearman**
TREC-7	**0.867**	**0.974**	0.613	0.814
TREC-8	**0.77**	**0.92**	0.552	0.730

Table 4. 5: Kendall's tau and Spearman correlations for TREC-7 and TREC-8 automatic runs

The correlation values reported in table 4.5 resulted from using the extraction model built with 3 to 5 terms and when extracting 10 KPs from the top 5 common documents for TREC-7. The same model seems to work best for TREC-8 only when extracting 5 KPs from each of the top 10 common documents retrieved for each topic. The topics for TREC-8 are known to be more difficult than the ones for TREC-7, and this could be the reason why more documents were needed to find more relevant information. For all the correlations values obtained, the p-value was less than 0.05 which means that the results are not found randomly.

The qrels are considered to be reliable when the Kendall's tau is above 0.8 (Voorhees, 2001): "evaluation schemes that produce correlations of at least .9 should be considered equivalent since it is not possible to be more precise than this. Correlations less than .8 generally reflect noticeable changes in the rankings". The KP automatic technique failed to achieve a tau value above 0.8 when we considered the top 100 pooled documents from all the new queries as relevant. To study further the quality of the pseudo-qrels we produced, we measured the actual relevance of each of the documents we presumed relevant. So, we referred to the TREC human judged qrels and we

marked the real binary relevance for every document in the pseudo-qrels produced. We skipped any document that was not judged by human assessors and therefore did not exist in the TREC qrels. Then, we computed a MAP score for the automatic runs in TREC-7 and TREC-8 based on the qrels with their real relevance, we ranked the runs and measured the Kendall's tau and Spearman correlations between this new ranking obtained and the TREC ranking produced from using the full set of human-built qrels. Spearman's rho with this manually marked set of relevance assessments for TREC-8 was 0.8404 while Kendall's tau was 0.6684, which are both better than the correlations that we obtained considering all the documents relevant in our pseudo-qrels. When the stopwords were included in the computation of the KP scores, we obtained a 0.8871 Spearman's rho and a 0.7323 Kendall's tau value. These last results seem close to the ones obtained by Efron as shown in Table 4.5. The logical explanation would be that considering all the documents as relevant especially those which are initially unjudged had some noticeable impact on the correlations. As for TREC-7, there were no changes in the Spearman and Kendall's tau values when computing the correlations using the real relevance of the pooled documents. A different combination of the parameters could affect the results obtained from one test collection to another because it depends on how the KPs are being extracted and which ones are being chosen as potential queries.

4.5 Using DUC2001 as training set

In this section, we will describe the experiments that were conducted on the same TREC test collections, but this time using a dataset that was manually annotated with keyphrases.

4.5.1 DUC2001 Description

The Document Understanding Conferences[10] is a conference series run by the National Institute of Standards and Technology (NIST) which aims to develop and improve summarization systems and their evaluation. The DUC2001 dataset is usually used for summarization, which consisted of 308 news articles collected from TREC-9. Wan and Xiao (2008) manually annotated the documents in this dataset with keyphrases. So, for example, for the document AP830325-0143, the keyphrases manually assigned were: 987-foot tanker exxon valdez, oil spill, major environmental catastrophe, cleanup equipment, crude oil. This dataset seemed convenient for us to use in building the training model for keyphrases extraction since we were also working with news articles test collections.

4.5.2 Experiments to extract keyphrases

We started by building the training model, and tried different combinations for the minimum and maximum number of terms. We built a model with 2 to 4 terms since manually extracted keyphrases in the annotated DUC dataset consisted of up to 4 terms. We also tried the model with 3 to 5 terms, the

[10] https://duc.nist.gov/

one that worked best in our previous experiments in section 4.3.3, but the best results were obtained with the default model parameters which are a minimum of one term and a maximum of 5. The Porter Stemmer (Porter, 1980) seemed to work better than the default Lovins stemmer (1968) defined in KEA and the minimum number of occurrences was set to 1. After building the model, we proceeded as described in section 4.3.4. After running twelve Terrier retrieval models, we found the 5, 10, 15 and 20 documents most commonly retrieved by the models, and we extracted in turn 5, 10, 15, 20 and 25 KPs from each of these documents. We then computed a score for each keyphrase by counting the number of matching words between the KP and the initial TREC topic title and description fields. The KPs with the highest score were then used as queries and submitted to the Terrier BM25 retrieval model. Each TREC topic had several new queries submitted. We obtained the union of the top 100 ranked documents retrieved for each of these queries and we considered them to be relevant to the topic, and thus could form the pseudo-qrels for the test collection.

4.5.3 Evaluation of results

In order to evaluate the pseudo-qrels obtained from using the KPs extracted based on the model built from the DUC annotated dataset, we computed the MAP scores for the automatic TREC-7 and TREC-8 runs using the KP produced pseudo-qrels, then we ranked the runs according to their MAP scores and finally we measured the Spearman's rho and Kendall's tau

correlations between the new ranks obtained and the ranks according to TREC qrels. The correlations for TREC-7 are presented in table 4.6 below.

Number of KPs extracted	Top K documents used for extraction							
	5		10		15		20	
	τ	ρ	τ	ρ	τ	ρ	τ	ρ
5	0.5019	0.6968	0.4836	0.6726	0.4996	0.6913	0.4928	0.6829
10	0.4868	0.6660	0.4849	0.6599	0.4755	0.6590	0.5024	0.6882
15	0.5132	0.7031	0.5062	0.6880	0.4698	0.6597	0.4691	0.6581
20	**0.5183**	**0.7111**	0.5165	0.7011	0.5010	0.6912	0.5017	0.6964
25	0.5088	0.7043	0.5063	0.6933	0.5131	0.7080	0.5146	**0.7129**

Table 4. 6: TREC-7 Kendall's tau (τ) and Spearman's rho (ρ) correlations based on MAP scores between TREC rankings and rankings obtained using pseudo-qrels produced after training KEA with the DUC dataset

The best correlations were achieved when extracting 20 KPs from the 5 most commonly retrieved documents. Even though we got a slightly better Spearman's rho value when extracting 25 KPs from the top 20 documents, the improvement was not worth the effort. The best correlations were still less than the correlations observed with the automatic KP extraction technique (as reported earlier in table 4.5). We thought that using manually annotated keyphrases to train the model would actually lead to better correlations, but the findings we obtained reject this hypothesis. We list in the table 4.7 below some keyphrases which were extracted using the DUC model as a comparison to the KPs we obtained in table 4.2 in section 4.3.5.

Original TREC topic title	Keyphrases selected as new queries (DUC model)	Keyphrases selected as new queries (automatic model)
TREC-7		
Topic 351: Falkland petroleum exploration	- falkland islands - falkland islanders - ownership of the Falkland islands - waters adjacent to falkland	- falkland islands territorial - falkland islands councillor - ownership of the Falkland islands
Topic 400: Amazon rain forest	- amazon rain forest	- south americas amazon
	- amazon rain	- americas amazon basin
		- greatest rain forest

Table 4. 7: Example of keyphrases extracted for TREC-7 topics after training with the DUC dataset

Most of the KPs extracted were still the same irrespective of the training model used to extract them. The main reason for that could be related to the uncontrolled vocabulary in our case. KEA is usually used with a controlled vocabulary for agriculture or medicine, but here with news articles, we did not have any domain specific vocabulary that we could use. Looking at the results we got from the experiments on TREC-8 in table 4.8, the correlations have low values when compared to the ones in table 4.4 which were obtained from using the KEA with an automatically built training model. The best Spearman's rho we could achieve was 0.6755 when extracting 25 KPs from the top 20 documents, and a Kendall's tau of 0.5028 when extracting 20 KPs from the top 20 documents. The p-value for all the reported correlation values was less than 0.05.

Number of KPs extracted	Top K documents used for extraction							
	5		10		15		20	
	τ	ρ	τ	ρ	τ	ρ	τ	ρ
5	0.4286	0.6044	0.4432	0.6197	0.4404	0.6196	0.4663	0.6436
10	0.4608	0.6394	0.4701	0.6469	0.4566	0.6328	0.4852	0.6615
15	0.4664	0.6437	0.4839	0.6607	0.4815	0.6586	**0.5028**	0.6806
20	0.4619	0.6399	0.4784	0.6573	0.4841	0.6603	0.4975	0.6731
25	0.4757	0.6552	0.4920	0.6693	0.4928	0.6696	0.4970	**0.6755**

Table 4. 8: TREC-8 Kendall's tau (τ) and Spearman's rho (ρ) correlations based on MAP scores between TREC rankings and rankings obtained using pseudo-qrels produced after training KEA with the DUC dataset

Similar to TREC-7, the KPs extracted for the example topics 404 and 446 when using the manually annotated DUC dataset were the same as the ones extracted using the automatically built training model as shown in table 4.9 below:

Original TREC topic title	Keyphrases selected as new queries (DUC model)	Keyphrases selected as new queries (automatic model)
TREC-8		
Topic 404: Ireland, peace talks	- peace and a political settlement - longterm intentions in Ireland - northern ireland talks - peace in Northern Ireland	- peace and a political settlement - longterm intentions in Ireland - northern ireland talks
Topic 446: tourists, violence	- ensure tourists safety - attack on tourists - French tourists - German tourists - tourists to Egypt - tourists return	- ensure tourists safety - attack on tourists - acts of violence

Table 4. 9: Example of keyphrases extracted for TREC-8 topics after training with the DUC dataset

4.6 Conclusion

In this chapter, we described an approach that is based on keyphrase extraction from the common documents retrieved by twelve Terrier retrieval models for a given topic. We used the KPs which had the highest number of common terms with the TREC title and topic fields as new candidate queries to test our assumption that such queries could bring more relevant information and therefore retrieve more relevant documents. Each of these queries was submitted to a single information retrieval system which is the BM25. Forming the pseudo-qrels for the initial TREC topic was done by taking the union of the top k retrieved documents from the different ranked lists which were returned in response to each new query. These automatically generated qrels produced a ranking for the TREC automatic runs which positively correlated with the ranking resulting from using the actual qrels, however it was still not enough to say that the difference between the two rankings is not noticeable. We also tried to use a dataset manually annotated with keyphrases to build the training model and therefore to obtain better quality KPs which could enhance the correlations, however the results we got did not show this. On the contrary, the correlations were lower than those we obtained when using the automatically built models. The limitation of the KP technique lies in the number of parameters which have to be tuned from one test collection to another. Not having a domain specific test collection where we can use a

controlled vocabulary to train the KEA and extract KPs might also be a factor, as Witten et al. (2005) stated that their algorithm works better with a controlled vocabulary. We could not find a theoretical way to optimize the number of documents from which to extract the KPs, or the number of terms which a KP should have. Our settings were all determined empirically. The correlations achieved between the automatic TREC runs rankings based on the human-built qrels and the pseudo-qrels – the ones generated automatically - were positive correlations. However, there were several parameters that would affect the outcome of the KP technique such as the number of documents to use for the KP extraction, the number of KPs to extract from each document, the number of terms each KP should consist of, and which KP could be considered as a potential query for the TREC topic. One could also see this KP technique was similar to an automatic query expansion technique since we were choosing a KP that has as many common words with the initial topic as possible but expanded with new terms. Due to these limitations, we decided to use a more robust technique which could be applied to any test collection and which could provide higher correlations between the rankings of the retrieval systems. We thus investigated the use of machine learning algorithms to build the pseudo-qrels. Chapter 5 describes the experiments related to the supervised and unsupervised machine learning based approaches we tried to answer the research question Q2, whether it is possible to produce pseudo-qrels which could be

considered good enough to replace manually created qrels, and hence to reduce the cost and effort to build a test collection.

Chapter 5 – Machine Learning Techniques

5.1 Introduction

The previous chapter discussed a keyphrase-based (KP) approach to generate the pseudo-qrels for a test collection from a list of documents presumed relevant. Due to the shortcomings of the technique which were highlighted in section 4.6, we wanted to come up with a more robust technique that could be applied to any test collection with fewer parameters to be tuned.

In this chapter, we describe a set of techniques based on different machine learning algorithms to automatically produce or to expand the set of pseudo-qrels for a test collection. The aim of implementing these techniques, testing them on TREC ad hoc test collections and evaluating the results we obtained is to answer to the third research question of this thesis:

Q2. Is it possible to use machine learning techniques to expand an initial set of presumed relevant documents and produce more qrels?

The supervised machine learning algorithms can be used in classification tasks. They are usually trained with some initial input or training data and then they are used to predict the class or the label of some unseen or test data while the unsupervised machine learning algorithms have no details about the data, they are not trained, but they are automatically used to form

clusters or groups of data sharing some characteristics. In the case of using supervised machine learning, we trained the classifiers with some documents presumed relevant and then we used the classifiers to classify the documents into binary or multiple classes which yielded eventually to forming the pseudo-qrels. The classification process and the number of classes used are detailed in future sections. The details about the experiments conducted using the first approach, the K-nearest neighbour explained in section 3.5.1 and the results obtained are discussed in section 5.2. We then explore the use of the most common and simplest unsupervised machine learning algorithm, the K-means in section 5.3 and in section 5.4 we discuss all the experiments led on TREC test collections using the third approach which we call the ML technique, and which uses the supervised machine learning classifiers, the Naïve Bayes (NB) and Support Vector Machines (SVM). We perform an intrinsic evaluation based on the precision and recall metric for these techniques. We also evaluate them through an extrinsic evaluation that measures the correlations between the system rankings produced based on MAP scores from using the pseudo-qrels and the gold standard rankings based on the manual human assessments. The section also includes the details of the experiments and results obtained from using the doc2vec representation of the documents as opposed to the tf.idf representation. Even though the ML technique outperforms several of the previous techniques from the literature which aimed at ranking systems

with reduced human-built qrels or in absence of them, it still has some limitations which we discuss at the end of section 5.4. We conclude the chapter in section 5.5 by introducing an enhancement to the ML technique: the use of actual qrels.

5.2 Nearest neighbour

5.2.1 Introduction

The K-nearest neighbour (K-NN) algorithm is a non-parametric classification algorithm. It does not make any assumption about the data distribution. Given a set of classes with different objects belonging to each class, the algorithm must classify a new object into one of the defined classes by determining the nearest distance between the new object and the k nearest neighbours of a class. When "k" is set to 1, the object is classified based on its closest neighbour in each class. In the approach we suggested, we set k=1.

The methodology proposed in this section is based on automatically selecting a set of documents with a high probability of being relevant to a topic and then using them to find more relevant documents based on the nearest neighbour algorithm. This nearest neighbour (NN) technique does not require any human intervention and has no prior knowledge of the test collection's original qrels.

5.2.2 Origin of the work

Rajagopal et al. (2014) used two independent approaches to build the pseudo relevance judgements: one which was completely automated and did not require any human intervention and was based on a "cutoff percentage" of the number of documents to mark as relevant or non-relevant. The second was called "exact count" and it required previous knowledge of the number of documents judged relevant by the human assessors for each topic. Both approaches were discussed in detail in section 2.7.3.7 and the one based on the algorithmic representation for the automatic selection of the cutoff percentage was also provided.

Mollá (2013) used a distance-based measure to expand positive judgments only – to find more relevant documents. He had to start with some known relevant documents or qrels (at least one) for each topic and then apply a distance measure to find more documents. The distance measure was based on the cosine similarity measure between two document vectors as explained in section 2.7.3.6.

The problem with Rajagopal's automatic technique was the choice of the cutoff percentage value. It was selected at random without any significant justification. While for Mollà, his technique required at least one known qrel to be able to expand the positive judgments. So, both these techniques inspired us to propose a new methodology: Selecting a cutoff percentage that guarantees having presumably relevant documents for a topic and then

using the distance-based approach to automatically form the qrels for a topic; this algorithm is similar to the k-nearest neighbour. This idea directed us towards expanding the work and testing different machine learning algorithms: the unsupervised K-Means algorithm and supervised machine learning using the Naïve Bayes classifier and Support Vector Machines. The remainder of the chapter describes in detail each of the algorithms we studied along with the experiments we conducted and the results we obtained.

5.2.3 Experimental design

The main idea of the nearest neighbour technique is to automatically select a few documents which could be relevant to each topic and then use a distance measure to find more relevant documents for that topic and therefore generate the pseudo-qrels for the test collection. The details of the document selection and the distance computation are described below.

Because we cannot be sure of the selected documents' relevance, we used a set that has a high probability of being relevant. In their work, Rajagopal used a cutoff percentage of the number of occurrences of the documents in the pool and considered those relevant. In our version, we selected a cutoff percentage (S) defined as the minimum percentage that ensures selecting at least one document for each topic as shown in the NN pseudocode in section 3.5.1.2.

When the process was completed, we evaluated the list of qrels by computing the MAP scores for the retrieval systems using the newly generated qrels, ranking the systems and measuring the Kendall's tau and Spearman rho correlations between the initial ranking produced by the human relevance assessments and the one computed based on the automatically generated qrels. Our technique resulted in higher correlations than Rajagopal's and had a lower false positive rate. The described process is shown in figure 5.1 below.

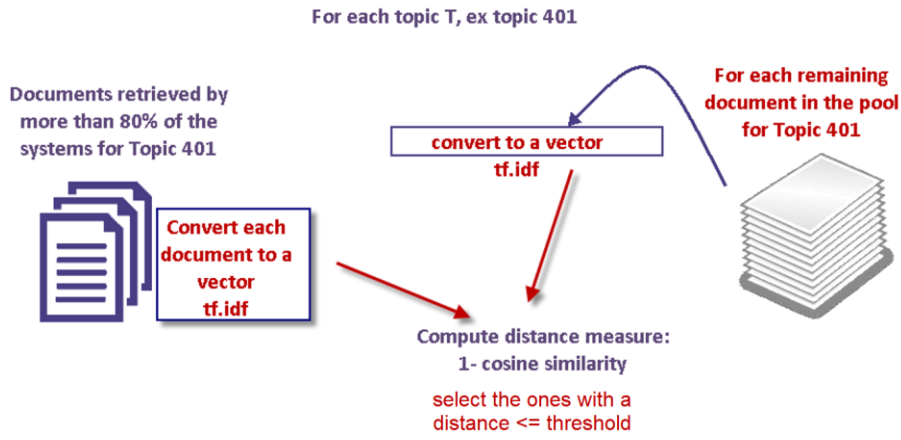


Figure 5. 1: Nearest Neighbour technique

5.2.4 Experiments and results

The experiments were conducted on the TREC-7 and TREC-8 test collections. Since the experiments conducted by Rajagopal et al. included only TREC-8, we replicated their experiments on TREC-7 first. Therefore, we explored both 35% and 50% cutoff percentages. For the nearest neighbour technique, we selected a value of 90 for the (S) percentage which forms the set of

documents with a high probability of being relevant. Then, we proceeded as explained above. TREC-7 consists of 103 runs. We tested for different distance threshold values ε which were obtained by applying equation (2.1) in section 2.7.3.6 and based on which we judged a document as either relevant or non-relevant. The correlations using the Kendall's tau and Pearson's r for each ε value are reported in table 5.1 below:

Threshold (ε)	Kendall's tau	Pearson
0.5	0.4814	0.6463
0.4	0.4946	0.6531
0.3	0.5114	0.6768
0.2	0.4810	0.6602
0.15	0.4637	0.6342

Table 5. 1: TREC-7 Kendall's tau and Pearson's r coefficients for different values of the distance measure threshold

The best correlations were achieved with a threshold value ε=0.3. The correlations which can be obtained by applying Rajagopal's automatic technique with the two cutoff percentages are less than the values obtained when applying the nearest neighbour technique. Table 5.2 summarizes the correlations between the ranking obtained after applying each of the approaches and the ranking based on the actual set of qrels.

	Rajagopal cutoff >35%	Rajagopal cutoff >50%	Distance threshold ε=0.3
Kendall's tau	0.4428	0.4391	0.5114
Pearson	0.5174	0.5178	0.6768

Table 5. 2: TREC-7 Kendall's tau and Pearson coefficients using Rajagopal's technique vs. the nearest neighbour technique

In the experiments conducted on TREC-8, the (S) percentage was found to be 80%. There are 129 runs in TREC-8. When using different cutoff percentages, we computed the percentage of actual relevant documents retrieved because in reality not all the documents retrieved in the cutoff set were judged relevant by the human assessors. With a cutoff percentage of 80%, almost 24% of the documents considered relevant were actually judged relevant by human assessors while with a cutoff percentage of 50%, only 11.9% of the documents considered relevant were actually relevant. Therefore, we used this set (S) in the remainder of the experiment to expand the first set of qrels generated and judge more documents as relevant using the same distance measure.

We ran each topic, formed a pool of depth 100, counted the number of runs which retrieved each document in the pool for the same topic, then ordered the pooled documents according to the count calculated in descending order. The documents with a percentage of occurrences greater than 80% were considered relevant and thus for each of the remaining documents retrieved, we measured the distance between its tf.idf vector representation and each vector representing the documents collected from the 80% cutoff. The documents which were at a close distance to one of the documents from the 80% set were considered relevant to the topic which the nearest document belongs to. We then computed the MAP scores for the 129 runs, using the automatically generated qrels and we rank the runs. Afterwards, we

computed the correlation measures Kendall's tau and Pearson's r coefficients between the ranking we obtained and the gold standard ranking obtained from the actual set of qrels. We tested for different distance thresholds, as we did for TREC-7 above. The different correlation results obtained are shown in table 5.3.

Threshold (ε)	Kendall's tau	Pearson	Harmonic Mean
0.5	0.4451	0.7017	0.5446
0.4	0.5033	0.7654	0.6072
0.3	0.5032	0.7804	0.6118
0.2	0.4879	0.7814	0.6007
0.15	0.4809	0.7786	0.5945

Table 5. 3: TREC-8 Kendall's tau and Pearson's r coefficients for different values of the distance measure threshold

The results showed that the best Kendall's tau value was obtained for ε=0.4 while the best Pearson's r value was for ε=0.2. For an overall comparison between the results using the harmonic mean which is the most suitable for the average of rates (in our case that's Kendall's tau and Pearson's r) of the two measures, the best value was achieved for ε=0.3.

The harmonic mean equation we used is shown in equation 5.1 below:

$$Harmonic\ Mean = \frac{2}{\frac{1}{\tau}+\frac{1}{\rho}} \quad (5.1)$$

where τ represents the Kendall's tau value at a given distance threshold and ρ is the Pearson coefficient at the same distance threshold. The Pearson's r coefficient showed better results than the ones Rajagopal et al. obtained with the automatic cutoff percentages approach, but the τ values they obtained were slightly better than the ones we got.

	Rajagopal cutoff >35%	Rajagopal cutoff >50%	Distance threshold ε=0.3
Kendalls' tau	0.515	0.506	0.5032
Pearson	0.736	0.739	0.7804

Table 5. 4: TREC-8 Kendall's tau and Pearson's r coefficients using Rajagopal's technique vs. the nearest neighbour technique

For a more detailed evaluation of the generated pseudo-qrels, we performed an intrinsic evaluation. We computed the precision and recall measures at different ranks (@5, @10, and @20… @ 100, @ 20 ... @ 1000) using the formulas (3.7) for the precision and (3.8) for the recall, as explained in section 3.2.5 We also computed the precision and recall for the qrels generated by Rajagopal et al.'s technique for a cutoff percentage >50%. Figure 5.2 plots the precision values at different ranks for Rajagopal et al.'s technique using the 50% cutoff percentage and the nearest neighbour technique using a distance threshold of 0.2. As can be seen our technique outperforms the values obtained by Rajagopal et al.'s at almost every rank except at rank 5 where the precision is really close (0.1 – Rajagopal and 0.08 using the NN technique), we also provide the actual precision values in table 5.5 below the figure.

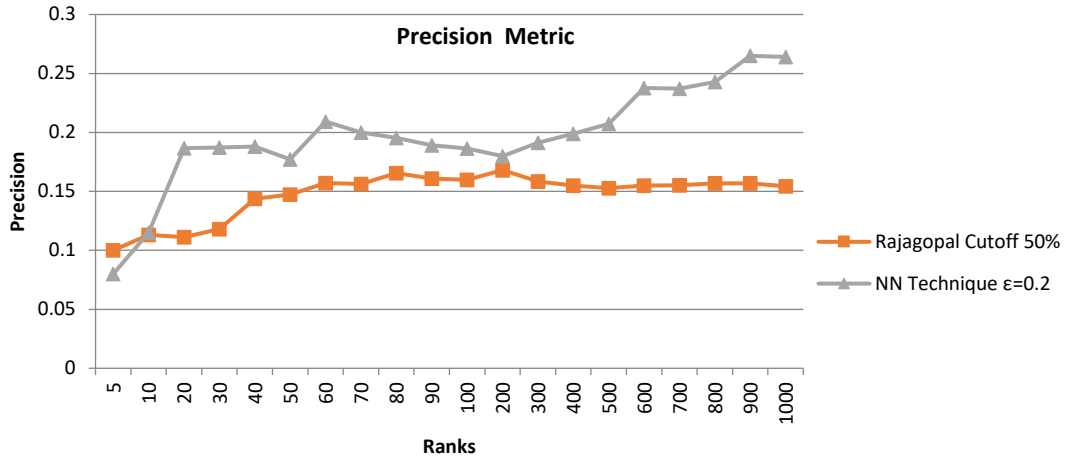


Figure 5. 2: Precision metric at different ranks for both techniques: Rajagopal et al.'s using a cutoff percentage of 50 and the NN technique using a distance threshold of 0.2

	Rajagopal Cutoff 50%	**NN technique, ε=0.2**
P@5	0.1	0.08
P@10	0.113	0.115
P@20	0.1112	0.1867
P@30	0.118	0.1872
P@40	0.1437	0.1881
P@50	0.1474	0.1773
P@60	0.1572	0.2092
P@70	0.1562	0.2
P@80	0.1655	0.1952
P@90	0.1608	0.1892
P@100	0.1598	0.1864
P@1000	0.1544	0.264

Table 5. 5: Precision metric at different ranks for both techniques: Rajagopal et al.'s using a cutoff percentage of 50 and the NN technique using a distance threshold of 0.2

For the recall, the cutoff of 50% provides better recall values than the NN technique using a distance threshold of 0.2. But if we increase the distance threshold to 0.5, our method can achieve similar or even better scores at

some ranks as the plot in Figure 5.3 shows. The raw recall values are listed in table 5.6 below.

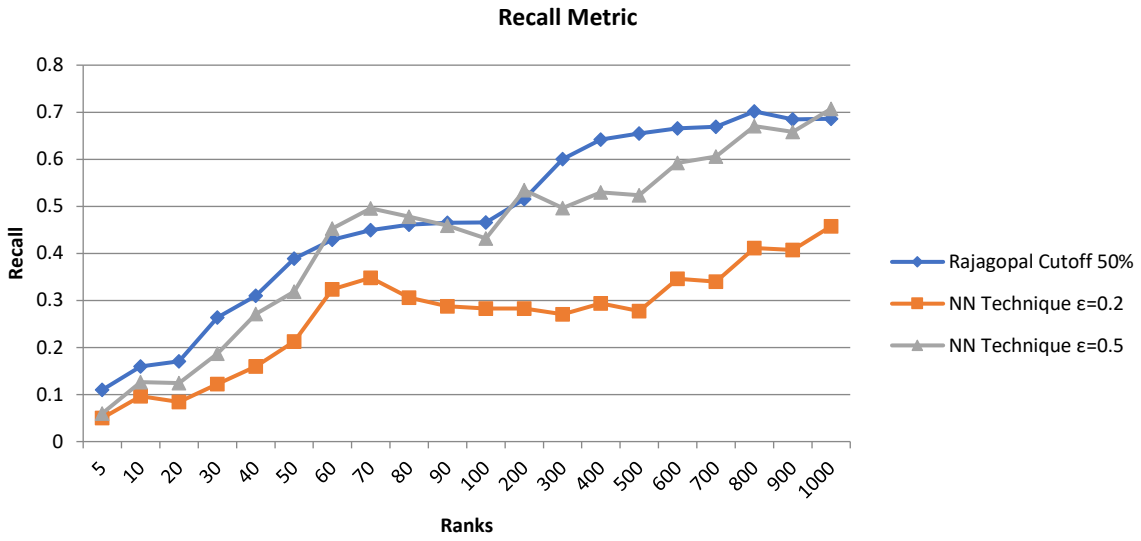


Figure 5. 3: Recall metric at different ranks for two techniques: Rajagopal et al.'s using a cutoff percentage of 50 and the NN technique using a distance threshold of 0.2 and of 0.5.

	Rajagopal Cutoff 50%	NN technique, ε=0.2	NN technique, ε=0.5
R@5	0.11	0.05	0.06
R@10	0.16	0.0967	0.1267
R@20	0.1703	0.084	0.1247
R@30	0.2637	0.122	0.1867
R@40	0.31	0.1599	0.2715
R@50	0.3889	0.2125	0.319
R@60	0.4287	0.3236	0.4525
R@70	0.4492	0.3478	0.4959
R@80	0.4611	0.306	0.4778
R@90	0.465	0.2878	0.459
R@100	0.466	0.2829	0.7071
R@1000	0.6858	0.4575	0.06

Table 5. 6: Recall metric at different ranks for two techniques: Rajagopal et al.'s using a cutoff percentage of 50 and the NN technique using a distance threshold of 0.2 and of 0.5.

5.2.5 System subrankings evaluation

A test collection is usually used to evaluate a single retrieval system by determining how well the system can retrieve the relevant documents at high ranks. Therefore, a good system will have a high recall value. The test collection will also provide the ability to compare between different systems and their ability to find not only relevant documents, but also rare ones. Thus, it can be used as well to compare between different retrieval systems and find the best ones. A limitation of an automatic technique could be the measurement of recall since building the qrels doesn't involve any human intervention. Therefore, a motivation for automatically producing qrels could be to compare between the systems by ranking them and being able to discriminate between the different groups of systems: the best ones, the average and the poor ones. That is why we divided the TREC-8 systems into three subsections based on the retrieval effectiveness value, the MAP value: the top third of the systems were considered to be best performing systems, the middle third were the average performing systems and the bottom third were the poor performing systems. Grouping the systems into different groups was done to show whether our approaches perform better for a specific subset of systems than the others. We then computed Kendall's tau for each group of systems based on the results achieved by Rajagopal et al.'s cutoff >50% approach and our nearest neighbour method with a threshold value $\varepsilon=0.3$. Based on the results described in table 5.7, both

techniques fail to discriminate between the best performing systems, but they can better identify the poor systems. A positive correlation exists between the average system rankings, but nevertheless these correlations indicate a major change in the system ranking as the tau value is below 0.8.

Methods	Best Performing Systems		Average Performing Systems		Poor Performing Systems	
	Kendall's tau	Pearson	Kendall's tau	Pearson	Kendall's tau	Pearson
Cutoff >50% (Rajagopal's)	-0.2313	-0.8111	0.3842	0.5919	0.7799	0.9169
Cutoff >=80% and ε=0.3	-0.2174	-0.8128	0.3324	0.5066	0.7773	0.9435

Table 5. 7: TREC-8 Kendall's tau and Pearson correlations for the three groups of systems using a 50% cutoff percentage and the nearest neighbour technique.

The negative correlations we observed between the best systems indicates that the rankings produced by the NN technique is almost the opposite of the ones obtained from using the human-built qrels and this could be affected by the hard topics found in each of the TREC test collections. The best systems are usually good at finding most of the relevant documents and the rare ones because they usually perform differently from other systems. Hard topics have only a few numbers of documents judged relevant. When we were automatically selecting the relevant set, we could have been falsely presuming documents as relevant and therefore a topic which only has 5 relevant documents according to the initial qrels would have many more selected by our technique, thus making the best systems behave like any other system. Since the newly devised approach based on

the nearest neighbour was able to generate pseudo-qrels which improve the correlations between the different TREC runs, we decided to investigate further the possibility of using machine learning techniques to further improve the correlations and most importantly to generate pseudo-qrels which can discriminate between the best performing systems. We began with the unsupervised machine learning algorithms, and we studied the possibility of using the simplest and most commonly used algorithm the K-Means in the task of producing automatic qrels.

5.3 Unsupervised K-means

5.3.1 Introduction

In this section, we describe the experiments conducted with both the unsupervised and semi-supervised K-Means clustering algorithm described in section 3.5.2. We deduced that it was not possible to use an unsupervised algorithm to achieve the task of producing pseudo-qrels.

5.3.2 Experimental design

We describe next the experimental design we followed to automatically generate a set of pseudo-qrels for a TREC test collection using the K-means algorithm with both random seed selection (unsupervised) and assigned initial seeds (semi-supervised).

5.3.2.1 Unsupervised K-means

For each TREC test collection, after applying the pooling technique to retrieve the top 100 documents retrieved by all the TREC runs for each of the 50 topics, we merged all the documents in the 50 different pools into one pool, so we could cluster the documents into 50 clusters, each representing a topic. We ran the K-means algorithm which was started initially by picking 50 different centroids randomly or 50 different documents. The K-means class found in the scikit-learn package[11] represents each document using a tf.idf vector after the tokens or terms in the document have been stemmed using the Porter Stemmer. The K-means class implemented in the scikit-learn package has several parameters which can be tuned. We list the parameters used and we give a short description of how they affect the output we obtained in table 5.8.

K-means parameter	Value	Description
n_clusters	Set to 50	The number of clusters to form
init	(1) kmeans++ (default) (2) ndarray	Defines the method of selecting the initial centroids. (1) The kmeans++ is a fast way to randomly select the initial seeds. (2) ndarray allows the user to assign the initial centroids.
n_init	Default is 10	Number of times the K-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia. K-means aims to choose centroids which minimize the inertia or the within-cluster sum of squares criterion. Inertia is a measure of how internally coherent clusters are.
max_iter	Default is 300	The maximum number of iterations of the K-means

Table 5. 8: K-means parameters used

[11] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Once the algorithm had finished, we matched the terms of each centroid vector to the terms of each TREC topic vector also weighted by the tf.idf and computed the cosine similarity between the two vectors. The terms of the centroids are many more than the topic terms which can be at most 20 terms. Thus, we chose the 50 most frequent terms from each centroid to perform the comparison. However, when applying the above steps, it was not possible to assign each cluster to a single topic. Several clusters were similar to the same topic. We also tried to increase the number of clusters to 75 but we still could not match between all the topics and the clusters formed from the random seeds. For some topics we could not find any matching cluster, and for others, several clusters had a high cosine correlation with the same topic. We applied the algorithm to both TREC-7 and TREC-8.

These results were not surprising since the initial centroids were picked randomly and there was a chance of picking documents as centroids for different clusters which were relevant to the same topic. Since the K-means algorithm provides the possibility of initializing the centroids to a known set, we thought we could solve the problem of not having 50 clusters to represent the 50 topics by initializing the centroids using known documents for each of the topics. Moreover, in the classification we proposed, each document was assigned to one topic only even though the same document could have been retrieved and considered relevant for more than one topic.

Hence, we tried feeding the K-means the initial seeds or initial centroids instead of letting the algorithm pick them automatically. We will now describe the use of a semi-supervised version of the K-means algorithm.

5.3.2.2 Semi-supervised K-Means

A semi-supervised variation of the K-means can be run by changing the "init" parameter to "ndarray". To ensure having 50 clusters for the 50 TREC topics, the optimal centroid seeds should be the centroid of a few known relevant qrels. Thus, we could expect that relevant documents would form a cluster for each topic since it was shown that relevant documents are at a close distance to each other (Mollà, 2013). Because we wanted to keep our technique automated without the use of actual qrels, we selected the centroid of the documents which have a high probability of being relevant, these being the documents retrieved by most of the systems which we defined as (Ri) in our nearest neighbour approach – the documents retrieved by S% of the systems, where S is the minimum percentage required to have at least one document selected for each topic. We applied the semi-supervised K-means. Only the documents classified and added to the clusters were considered relevant to the topic. Even after feeding the algorithm with initial centroids and being able to automatically produce a set of qrels, the MAP scores computed were very low for TREC-7 and TREC-8. The best MAP value was of 0.1172 for TREC-8 and 0.0345 for the TREC-7. Even with such low MAP scores, we evaluated the system ranking by

measuring the correlations between the gold standard ranking using the human-built qrels and the new ranking obtained from the K-means pseudo-qrels.

	Kendall's tau	Spearman rho
TREC-7	0.1940	0.2768
TREC-8	0.3833	0.5226

Table 5. 9: TREC-7 and TREC-8 correlations between the systems ranking

The correlation results using Kendall's tau and Spearman's rho are low. This led us to conclude that it is not possible to use an unsupervised algorithm to form exactly 50 distinct clusters for the distinct TREC topics. Since the semi-supervised algorithm provided low but positive correlations, we expanded the study to train classifiers with input data by using two supervised machine learning classifiers: Naïve Bayes and Support Vector Machines.

5.4 Supervised Machine Learning

5.4.1 Introduction

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. Two classifiers are often used for text classification: The Naïve Bayes (NB) and Support Vector Machines (SVM) which we described in chapter 3, section 5.3. Since the aim was to produce pseudo-qrels, without any human intervention if

possible, the initial training set we picked for the classifiers was also automatically selected by using the documents retrieved by the S% of the runs as detailed in section 5.2.2 of the nearest neighbour technique.

We defined the problem of generating qrels as a classification problem. We have tested two different approaches which we called the two-class technique (see Two-Class ML Pseudocode, section 3.5.3.4) and the 50-class ML technique (50-Class ML Pseudocode, section 3.5.3.4).

5.4.2 NB and SVM Technical specifications

Before we proceed to the experimental design, we will go through some technical details about the tools used to train and run both the NB and SVM classifiers. We also refer to the scikit-learn package.

5.4.2.1 The NB technical specification

Since the data used as input and ouptut is represented as vectors and therefore the features are discrete, the Multinomial NB class[12] is most suitable for the text classification problem since it is suitable for classification with discrete features such as words count. The Multinomial NB has several parameters which could be tuned. We only tuned the "alpha" parameter which is defined as the Laplace/Lidstone smoothing parameter initially set to 1. When it has a value of 0, this indicates that there is no smoothing applied. It is an additive smoothing parameter which is added to a feature value to avoid a zero count. In our experiments, we tested with the default

[12] http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB

value of alpha and then we tested again with values of 0.1, 0.2, …, 0.9. We noticed that with a value of 0.1, there was a slight improvement in the correlations in some cases as described later.

5.4.2.2 The SVM technical specification

SVM are a set of supervised learning methods used for classification, regression and outlier detection. Our problem is a classification task. The scikit-learn package provides three different classes for the SVM classification[13]: SVC, NuSVC and LinearSVC. In our experiments, we used the LinearSVC[14] as it implements the "one-vs.-the-rest" approach which trains one classifier per class and returns a confidence score of the predicted output. We kept all the parameters' default values.

5.4.3 Experimental design

5.4.3.1 The two-class approach

The output of this approach is a list of documents classified into one of the two categories: Relevant or Non-Relevant. This list represents the set of pseudo-qrels for the test collection. In order to form this set, we proceeded as follows: we started by applying the pooling technique and then counting the number of runs which retrieved each of the documents in the pool. The documents which were retrieved by more than S% of the runs or systems were considered as the training set for relevant documents and called the

[13] http://scikit-learn.org/stable/modules/svm.html#svm-classification
[14] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC

"RelSet" as they have a high probability of being relevant. As for the "NonRelSet", or the training set to be used for the non-relevant documents, the same S% of documents as was picked for the "RelSet" was selected, but this time we took the documents which had the lowest number of occurrences in the pool. We tested the NB and SVM classifiers. After the classifiers, NB and SVM, were trained using the RelSet and NonRelSet, the remainder of the documents in the initially formed pool were classified as either relevant or non-relevant. The classified list along with the training sets were considered the newly generated qrels.

5.4.3.2 The 50-class approach

In this second approach, instead of selecting two different document sets, the highest S% and the lowest, we created only one training set for all the topics. We started first by selecting, for each topic, the documents retrieved by S% of the TREC runs. We assigned the topic id (e.g. 401 from TREC8) as a label for each document in this set. For example, if the documents $\{d_1, d_5, d_{20}\}$ were retrieved for topic 1, we would create a class labelled "1" that had the 3 documents as its training set. We repeated this process for all the 50 topics. We trained the NB and SVM classifiers using the labelled documents. Next, we used the trained classifiers to predict the topic ID, or label for each remaining unlabelled document in the pool of documents retrieved for all the topics. In this second approach, the number of relevant documents could be expanded because the classifier could predict that a document belonged 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 were also considered relevant to the topic. The pseudo-qrels were formed by considering the labelled documents as relevant to the topic and all the initially retrieved documents which were labelled with another topic ID were considered non-relevant.

To evaluate both approaches, we computed the mean average precision (MAP) of the systems using the trec_eval package. Then, we measured the Kendall's tau and Spearman correlations between the TREC runs ranking using the ML pseudo-qrels and the ranking obtained from using the human-built qrels. We compared our results to the scores reported from previous studies.

5.4.4 Experiments and results

We tested both approaches described in the previous section on TREC test collections. We used TREC-6, TREC-7 and TREC-8 to be able to compare our results with previous studies. TREC-6 had 74 participating runs, 103 runs participated in TREC-7, and TREC-8 had 129 participating runs. The S% of the systems which guarantees that each topic has some presumably relevant documents returned for training purposes was found to be 80% for TREC-7 and TREC-8, while for TREC-6 we used a value equal to 75%.

Using the two-class approach which classifies the documents as relevant or non-relevant, the NB seems to give better correlation results for TREC-6,

while the SVM using a linear kernel works better for TREC-7 and TREC-8. We show the results obtained in table 5.10.

	Two-class ML technique			
	SVM (Linear Kernel)		**Naïve Bayes (Multinomial) alpha=1.0**	
	Kendall's tau	Spearman	Kendall's tau	Spearman
TREC-6	0.5266	0.7009	0.5408	0.7024
TREC-7	0.4482	0.5577	0.4328	0.5402
TREC-8	0.6095	0.7369	0.6037	0.7266

Table 5. 10: Correlation measures based on MAP scores for TREC-6, 7 and 8 using the two-class ML technique

For the smoothing parameter alpha value of 0.1, we saw better correlations than the ones reported in the above table for TREC-7, as shown in table 5.11, which means that there is almost no smoothing applied, and some features could have a count of value 0, so we can conclude that when some words are discarded, the quality of the qrels seems to be better.

	Two-class ML technique	
	Naïve Bayes (Multinomial) – alpha 0.1	
	Kendall's tau	Spearman
TREC-6	0.4669	0.6362
TREC-7	0.4491	0.5587
TREC-8	0.6001	0.7229

Table 5. 11: Correlation measures based on MAP scores for TREC-6, 7 and 8 using the two-class ML technique, with NB alpha set to 0.1

The second ML approach based on the 50-class classification allows more documents to be discovered for a topic which may not have been part of the pool judged by human assessors, not all the documents which were automatically classified for a topic were actually judged by human assessors.

This second approach seems to provide better correlation results when compared with the two-class 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 5.12.

	50-class ML technique					
	SVM (Linear Kernel)		**Naïve Bayes (Multinomial) alpha=1.0**		**Naïve Bayes (Multinomial) alpha=0.1**	
	Kendall's tau	Spearman	Kendall's tau	Spearman	Kendall's tau	Spearman
TREC-6	0.5712	0.7405	0.5864	0.74312	0.5887	0.7491
TREC-7	0.4116	0.5284	0.5128	0.6151	0.5661	0.6702
TREC-8	0.4494	0.5938	0.5144	0.6479	0.5330	0.6654

Table 5. 12: Correlation measures based on MAP scores for TREC-6, 7 and 8 using the 50-class ML technique

The correlation scores achieved from the 50-class classification by topic using the NB classifier for TREC-6 and TREC-7 outperform the first classification technique which classifies the retrieved documents as either relevant or non-relevant, however the two-class approach works better for TREC-8. This is due to the fact that the classification by topic is leading to more documents in the qrels sets not just the ones found in the pool for each topic. Now we compare our best 50-class ML technique with the previous methods which were discussed in the literature review chapter (chapter 2). The Spearman correlation values based on MAP scores were reported by the authors for TREC-6 and TREC-7.

	RS	RC	CB	Single %	ASS	ASSBC	50-class ML NB technique
TREC-6	0.436	0.384	0.717	0.618	0.630	**0.854**	**0.749**
TREC-7	0.411	0.382	0.453	0.550	0.585	0.631	**0.670**

Table 5. 13: Spearman correlations based on MAP values for TREC-6 and TREC-7 using automatic methods to produce pseudo-qrels

The second column (RS) in the table shows the Spearman correlation for Soboroff's (2001) technique that uses the random sampling of documents. The third column (RC) reports the results obtained using the reference count technique by Wu and Crestani (2003). The results reported by Nuray and Can (2006) based on a data fusion technique (CB) shows an improvement over most of the techniques for TREC-6 and better than the first two for TREC-7. The "Single%" value in column 5 was recorded by Spoerri's (2007), and the average system similarity (ASS) was proposed by Aslam (2003). The ASS based on clustering (ASSBC) suggested by Shi et al. (2010) provided much better correlations than all the previous approaches. The last column shows data from our ML technique. These correlations are better than all of the previous automatic approaches except for the ASSBC on TREC-6, but the ML technique requires less effort than ASSBC which requires clustering the systems and selecting a representative document from each cluster. For TREC-8, the best Spearman's rho was 0.7363 using the SVM. The previous methods were not tested on TREC-8 except by Soboroff et al. (2001) and he reported an average of 0.5 for Kendall's tau, while we

obtained a value of 0.6095. All these automatic techniques show positive correlations between the overall systems rankings and we succeed at obtaining better scores for the correlations, but the overall rankings are not enough to evaluate the qrels. We perform an intrinsic evaluation and then we do a more detailed evaluation to see how well the generated qrels can discriminate between the best systems.

5.4.5 Intrinsic evaluation

The experiments described so far constitute an extrinsic evaluation of the automatically generated qrels, where we evaluated 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 evaluated the performance of the systems using the generated qrels when compared to the qrels built by the human assessors using the recall and precision metrics described in section 5.2.3. We computed the recall metric at different ranks (@5, @10, and @20… @ 100, @ 20 ... @ 1000) using equation (5.3). We then computed the precision metric at the same ranks using equation (5.2). 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 formula (5.4) below, where p is the precision at rank (i) and r is the recall at rank (i).

$$\mathrm{F} = \frac{2}{\frac{1}{p}+\frac{1}{r}} \qquad (5.4)$$

The values obtained for our experiments which used supervised machine learning algorithms are plotted in Figure 5.4 and the raw values are given in table 5.14 below the figure.

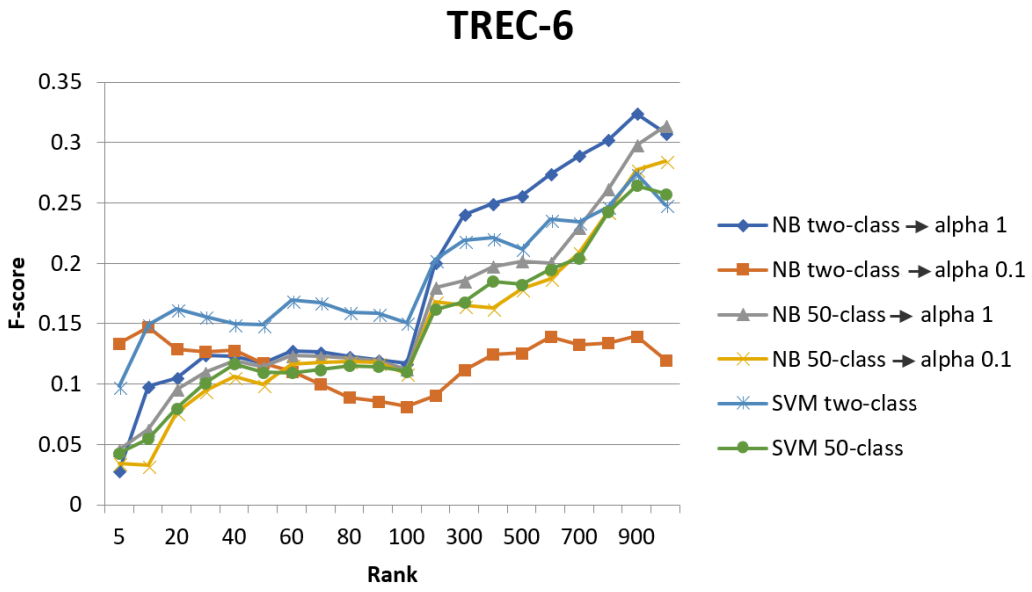


Figure 5. 4: F-score values at different ranks obtained for all the ML technique classification approaches for TREC-6

Rank	NB two-class, alpha=1	NB two-class, alpha=0.1	NB 50-class, alpha=1	NB 50-clas, alpha=0.1	SVM two-class	SVM 50-class
5	0.0282	0.1339	0.0457	0.0343	0.0982	0.0429
10	0.0982	0.1472	0.0625	0.0328	0.1493	0.0547
20	0.1051	0.1289	0.0955	0.0765	0.1620	0.0799
30	0.1238	0.1268	0.1097	0.0944	0.1559	0.1008
40	0.1229	0.1281	0.1193	0.1059	0.1497	0.1167
50	0.1170	0.1174	0.1141	0.0996	0.1492	0.1097
60	0.1274	0.1103	0.1236	0.1165	0.1695	0.1093
70	0.1265	0.0999	0.1228	0.1182	0.1675	0.1118
80	0.1229	0.0890	0.1218	0.1187	0.1596	0.1150
90	0.1200	0.0859	0.1196	0.1176	0.1585	0.1144
100	0.1169	0.0815	0.1122	0.1088	0.1510	0.1105
1000	0.3078	0.1201	0.3138	0.2849	0.2484	0.2576

Table 5. 14: F-score values at different ranks obtained for all the ML technique classification approaches for TREC-6

Figure 5.4 shows the different F-scores for the TREC-6 test collection using the ML technique with the two-class and the 50-class approaches and the different classifiers. The best F-score was achieved using the NB classifier with the 50-class approach, but it did not achieve a high recall score. This is expected since the technique has no human intervention and therefore we cannot be sure of the actual relevance of the documents used for training and thus that of the resulting classified documents. The best F-score for both TREC-7 and TREC-8 was obtained using the NB 50-class approach as shown in figures 5.5 and 5.6, followed by the exact values in tables 5.15 and 5.16, and the maximum value was 0.2693 for TREC-7 and 0.2743 for TREC-8. These scores indicate low precision and recall.

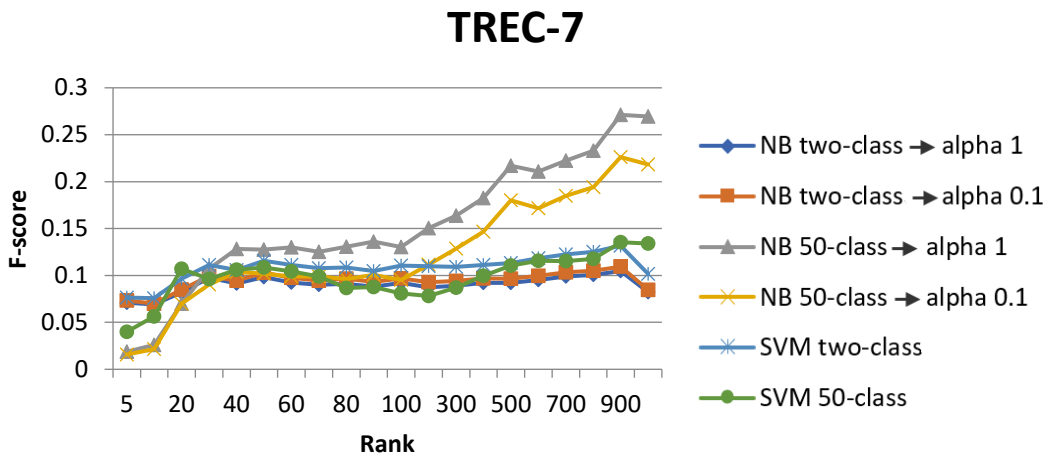


Figure 5. 5: F-score values at different ranks obtained for all the ML technique classification approaches for TREC-7

Rank	NB two-class, alpha=1	NB two-class, alpha=0.1	NB 50-class, alpha=1	NB 50-class, alpha=0.1	SVM two-class	SVM 50-class
5	0.0710	0.0732	0.0187	0.0157	0.0764	0.0400
10	0.0691	0.0700	0.0260	0.0215	0.0758	0.0562
20	0.0820	0.0847	0.0698	0.0700	0.0965	0.1071
30	0.0975	0.0992	0.1071	0.0905	0.1110	0.0961
40	0.0912	0.0943	0.1282	0.1036	0.1057	0.1061
50	0.0983	0.1029	0.1276	0.1021	0.1157	0.1086
60	0.0925	0.0974	0.1298	0.0990	0.1111	0.1042
70	0.0898	0.0946	0.1250	0.0990	0.1076	0.0988
80	0.0911	0.0966	0.1308	0.0960	0.1084	0.0868
90	0.0878	0.0930	0.1360	0.1004	0.1048	0.0875
100	0.0921	0.0967	0.1302	0.0952	0.1104	0.0809
1000	0.0820	0.0847	0.2693	0.2182	0.1015	0.1339

Table 5. 15: F-score values at different ranks obtained for all the ML technique classification approaches for TREC-7

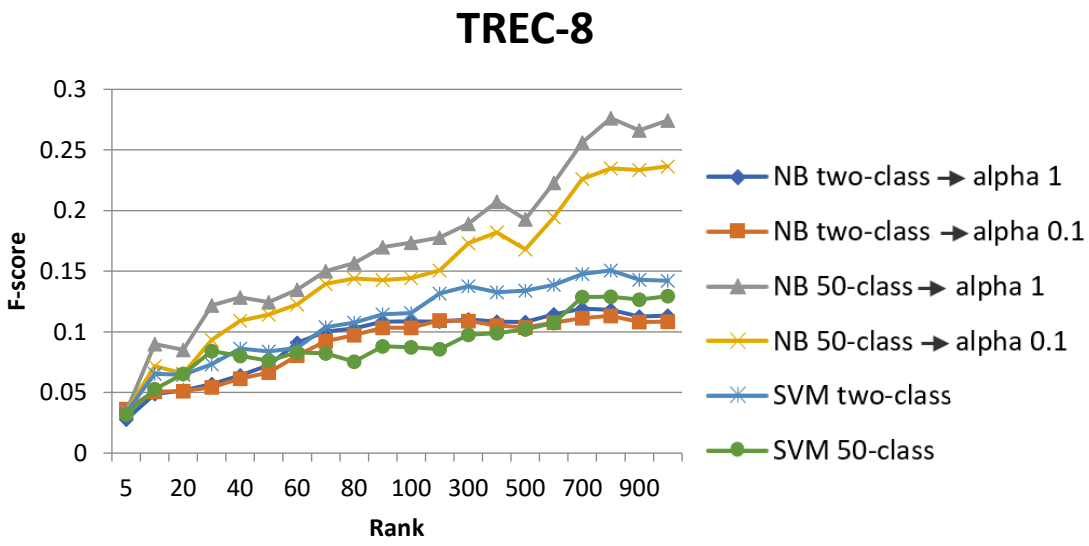


Figure 5. 6: F-score values at different ranks obtained for all the ML technique classification approaches for TREC-8

Rank	NB two-class, alpha=1	NB two-class, alpha=0.1	NB 50-class, alpha=1	NB 50-class, alpha=0.1	SVM two-class	SVM 50-class
5	0.0277	0.0365	0.0343	0.0343	0.0336	0.0320
10	0.0486	0.0505	0.0900	0.0716	0.0655	0.0528
20	0.0517	0.0511	0.0853	0.0657	0.0645	0.0653
30	0.0569	0.0542	0.1218	0.0934	0.0732	0.0840
40	0.0641	0.0613	0.1284	0.1092	0.0861	0.0802
50	0.0726	0.0664	0.1246	0.1142	0.0838	0.0761
60	0.0911	0.0803	0.1347	0.1225	0.0869	0.0830
70	0.1003	0.0924	0.1501	0.1396	0.1038	0.0823
80	0.1029	0.0972	0.1567	0.1441	0.1076	0.0754
90	0.1085	0.1034	0.1700	0.1426	0.1146	0.0881
100	0.1088	0.1033	0.1736	0.1442	0.1156	0.0872
1000	0.1133	0.1084	0.2743	0.2365	0.1421	0.1295

Table 5. 16: F-score values at different ranks obtained for all the ML technique classification approaches for TREC-8

5.4.6 Using doc2vec document representation

5.4.6.1 Introduction

The motivation behind Le and Mikolov's work (2014) is driven by the fact that the bag-of-words representation of a paragraph does not consider the semantics of words or their position in a sentence. They proposed an unsupervised algorithm, the "paragraph vector", which provides a representation for text documents based on a fixed-length set of features. Similar to the algorithm "word2vec" (Mikolov et al., 2013) which gives a semantic representation of a word given the other words in context, in the paragraph vector algorithm, also known as "doc2vec", every paragraph is mapped to a unique vector, which is the sum of the word vectors for each of the individual words in the paragraph.

5.4.6.2 Experiments

The experiments conducted on TREC test collections using the two-class and 50-class approaches in the ML technique were repeated, except that this time we used a doc2vec vector representation of the documents instead of the tf.idf. The "doc2vec" module implemented in the gensim[15] library allows setting different parameters such as the vector size, the window size, the minimum count which ignores all words with a frequency less than this number, the number of iterations and other parameters. 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 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 labelled using the trained doc2vec model and both the NB and SVM classifiers. The test data in our case was much larger than the training data. The number of documents (from the S% cutoff) used to train both the classifiers and the doc2vec model was very low compared to the number of documents for which a label must be predicted. Yet, the results were not far from the ones we achieved using the tf.idf. We summarize in table 5.17 the Spearman correlation coefficients computed for the different test collections using both classifiers and the two different classification approaches. The

[15] https://radimrehurek.com/gensim/

parameters used for the doc2vec model were the following: min_count=1, window=10, size=100, sample=1e-4, negative=5, workers=8.

	Using SVM		Using NB, alpha 1.0	
	50-class	**Two-class**	**50-class**	**Two-class**
TREC-6	0.6257	0.6813	0.7555	0.7550
TREC-7	0.6175	0.5594	0.6116	0.5158
TREC-8	0.7293	0.6334	0.7081	0.6839

Table 5. 17: Spearman correlation based on MAP values for TREC-6, 7 and 8 using doc2vec document representation

There was a slight improvement in the Spearman correlation values using the two-class approach and the NB on TREC-6 and TREC-7, while with the 50-class approach, using the SVM classifier, we noticed a minor improvement for TREC-7 and TREC-8. We did not expect to have better correlations since the doc2vec model works best when it has been trained with a large set of data. The test data is usually less than the training data but in our case, it was the other way round, we only have a few documents in the training set and we had to apply the classification for a higher number of documents. Since there was not any significant improvement in the correlation values using the doc2vec representation, we retained the tf.idf representation of the documents for the remainder of this thesis.

5.4.7 System subrankings

All previous automatic techniques have failed to discriminate between the best systems. Here we show how well the qrels generated using the ML technique can discriminate between three subsets of the systems: those with the best, average, and poor performance. We computed the MAP values of all the systems using the human-built qrels, then we ranked the systems according to the MAP scores and we ordered them from the highest rank (rank 1) to the lowest. We selected the first third of our set of systems as the best systems, the second third as the average systems and the last third as the poor systems. Now, using automatically produced qrels from the ML technique, we computed the MAP scores for the TREC systems and we ranked them. We measured the Spearman correlations between the TREC ranking of the best systems and the ranking we obtained, then we did the same for the average and poor systems. For each test collection, there were six sets of automatically generated qrels:

- Using the first two-class approach that classifies the documents in two categories: "relevant" and "non-relevant"
    - a first set was produced using the SVM
    - another set using the NB with the default value for the smoothing parameter alpha = 1.0
    - a third set also using the NB but with alpha set to 0.1

- Using the second approach which we called the 50-class approach, we also obtained three different sets of pseudo-qrels
    - using SVM
    - using NB with alpha 1.0
    - using NB with alpha 0.1

Figures 5.7, 5.8 and 5.9 show the Spearman correlation between the best systems, the average ones and the poor ones for TREC-6, TREC-7 and TREC-8 respectively. For most of the Spearman correlations for the best and average systems in TREC-6, the p-value was greater than 0.05. However, for the poor systems it was less than 0.05. For the TREC-7 and TREC-8, the p-value was less than 0.05 for most of the reported values.

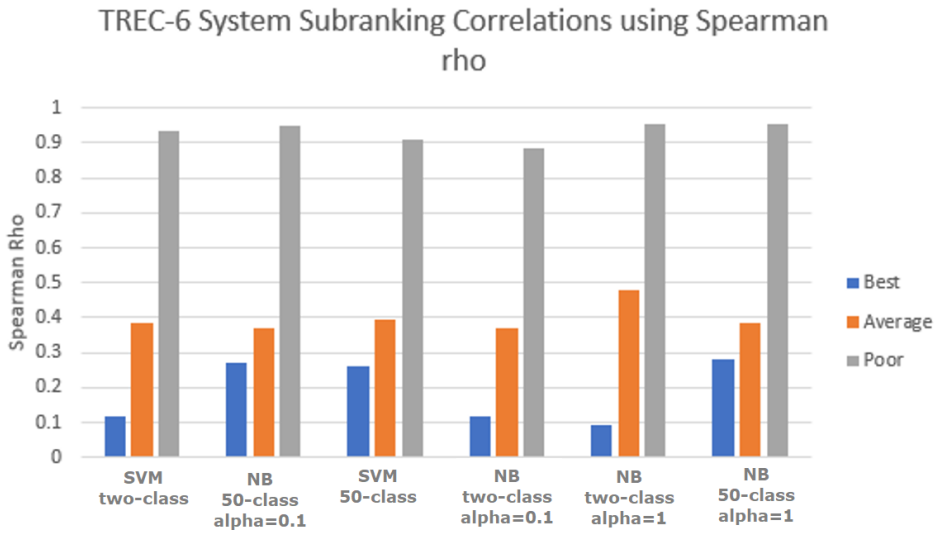


Figure 5. 7: Spearman correlations between the system subrankings using the ML technique for TREC-6

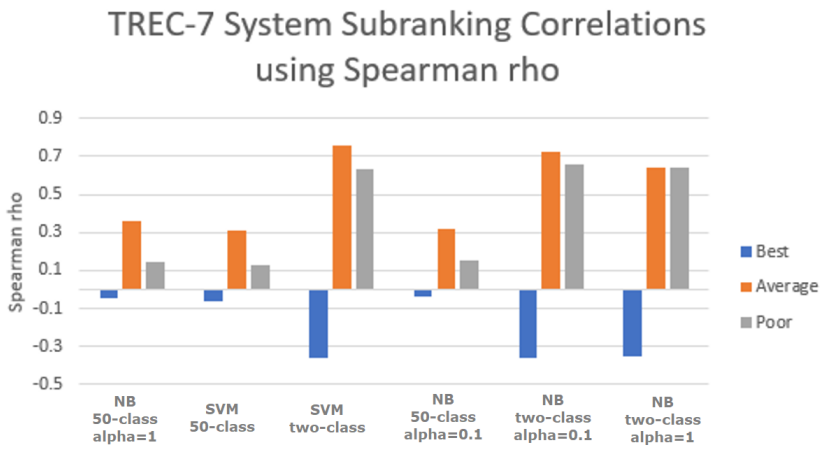


Figure 5. 8: Spearman correlations between the system subrankings using the ML technique for TREC-7

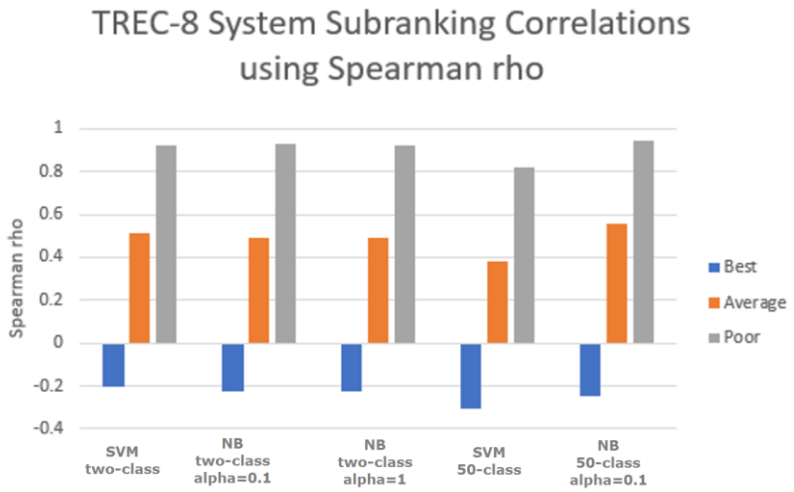


Figure 5. 9: Spearman correlations between the system subrankings using the ML technique for TREC-8

The Spearman correlations between the rankings of the best systems using the ML technique and the TREC rankings in TREC-7 and TREC-8 were negative which means that they were to some extent ranked in reverse order. This was somehow expected because the best systems are usually the ones able to find more relevant documents than the other systems or even

find rare relevant documents. On the one hand, we performed the evaluation by measuring the correlations between the systems rankings, and on the other we performed a statistical test for the MAP scores computed for the different TREC runs using each of the ML approaches as it became important to interpret how meaningful the results we obtained in experiments are. The importance of such tests was highlighted by Sakai (2016) in the review work he conducted on hundreds of papers submitted to SIGIR conferences and which lacked statistical testing. The p-value which is an important indicator of the difference between the gold standard values and the ones obtained throughout experiments was not enough for power analysis and thus other statistical tests could be used such as the t-test and the Analysis of Variance (ANOVA) which are not very commonly used in IR community according to the author. The effect sizes studied through these tests could not be applied to our results as there are no variations in the number of topics used or the pool size chosen in each of the experiments and we were not comparing between three or more groups of results, we were only evaluating the difference between two groups exactly: the gold standard scores obtained using human-built qrels and the scores generated using our automatic techniques. Thus, in addition to reporting the p-value, we used first the Shapiro test (Shapiro and Wilk, 1965) to determine whether the MAP scores are normally distributed or not. The p-value obtained for the MAP scores computed using the human-built qrels (original MAP) was greater than 0.05

which means that they were normally distributed while the MAP values computed using the NB and SVM classifiers with the ML two-class and 50-class approaches were not normally distributed with a p-value less than 0.05. Therefore, we applied the non-parametric Wilcoxon test (1945) that is used to compare between the average of two sets: the original MAP and each of the MAP scores produced from the ML techniques. The results are reported in table 5.18 below:

p-value for the Wilcoxon test when comparing the MAP scores for the TREC runs					
For TREC-6					
(Original, NB Two-Class)	**(Original, NB Two-Class alpha =0.1)**	**(Original, NB 50-Class)**	**(Original, NB 50-Class alpha=0.1)**	**(Original, SVM Two-Class)**	**(Original, SVM 50-Class)**
3.97E-08	2.14E-09	2.36E-06	0.0848	0.01399	0.002343
For TREC-7					
2.69E-07	1.42E-07	1.13E-10	0.0005817	1.46E-09	6.98E-11
For TREC-8					
4.11E-12	8.10E-13		2.50E-08	2.20E-16	2.92E-07

Table 5. 18: p-value for the Wilcoxon test when comparing between the TREC runs' MAP scores obtained using the human-built (original) qrels and the pseudo-qrels.

The p-value shown in the table indicates that there is a significant difference between the two sets, which means that the MAP scores computed based on the pseudo-qrels generated with the two ML approaches are different than the ones computed using the human-built qrels except for the p-value obtained when comparing the TREC-6 runs using the SVM 50-class pseudo-qrels, where the p-value > 0.05. The results would have been encouraging if we did not have any significant difference between the compared sets

because this would mean that the pseudo-qrels are as good as the baseline human-built qrels, but that was not possible to achieve. Since the correlations between the best systems rankings were negative, we can conclude that these systems have a high impact on the results we obtained with the Wilcoxon test as well. The selection process the ML technique uses to form the "RelSet" does not guarantee that the documents selected are actually relevant, but these documents have a high probability of being relevant since they were retrieved by several systems. This technique will not select any rare relevant documents for training either since they will not be retrieved by more than S% of the systems. If this assumption is right, we might have been suffering from the problem of the "tyranny of the mass" described in Aslam et al. (2003), where the most popular documents are retrieved as a training set for relevant documents. The set of qrels produced however was able to discriminate between the average and poor systems. We see high correlations between the rankings of the poor systems of more than 0.8 in TREC-6 and TREC-8 while for the average systems, there is still a positive correlation between the rankings, but it differs from one test collection to the other.

5.4.8 Limitation

Our ML classification technique using both approaches (the two-class and the 50-class) failed to discriminate between the best systems like most of the automatic techniques described in the literature which aimed at

automatically producing the pseudo-qrels for a test collection, but it outperforms them all in terms of the overall system correlations. Because a test collection should provide the ability to discriminate between the best systems and not only the poor ones, we proposed a variation of our ML technique which involves using a small number of actual qrels as training sets. The technique works well when we evaluate the generated qrels by comparing the overall systems ranking because the automatic technique is able to discriminate between the average and poor performing systems, and therefore the best systems which we fail to discriminate between, will not affect the other two thirds of systems in the overall rankings. The systems which are considered the best are the ones that are able to retrieve more relevant documents for a topic than others and even find rare ones in case of hard topics. In each TREC test collection, we have several topics that can be considered hard as they have only a few known relevant documents, through our automatic technique, we are selecting several documents which we presume relevant to the topics and then we expand this number, we could be finding these rare relevant documents but not for all the hard topics.

In the next section, we describe the modified version of the ML technique which makes it a semi-automatic approach since it involves using real qrels or actual qrels (AQ), qrels which are judged by human assessors. We refer to this technique as AQML. The aim of the AQML method is to improve the

overall systems ranking, the systems subranking and to discriminate between the best systems, all without having to invest great human effort as in building the actual qrels.

5.5 Conclusion

In this chapter, we described the experiments conducted using the nearest neighbour (NN) technique, the unsupervised and semi-supervised K-means algorithm and the technique we called ML that is based on the supervised machine learning classifiers NB and SVM. The NN technique was based on selecting a set of documents retrieved by more than S% of the systems, since we presumed that they have a high probability of being relevant to the topic, and then expanding this set of pseudo-qrels by measuring the distance between the tf.idf vector representing each retrieved document and each vector representing the documents presumed relevant. The documents which were at a distance less than a threshold ε were considered relevant. The pseudo-qrels generated using the nearest neighbour technique provided better correlations between the overall system rankings produced using the MAP scores from the NN qrels and the ranking based on the MAP scores obtained from using the TREC qrels than the ones obtained using the Rajagopal et al. cutoff percentage described in section 5.2.2 which was the baseline method. The NN method could not however discriminate between the best systems, and the correlations between the best system subrankings were negative. We wanted to use the unsupervised K-means to form 50

clusters for each of the 50 TREC topics, but we could not achieve this aim as we could not control the selection of the initial centroids and therefore it was possible to have several clusters related to the same topic. So, we used a variation on the K-means by initializing the centroids in a way such that each TREC topic had a cluster assigned to it. Unfortunately, this meant that the unsupervised K-means in both its versions could not be used to produce pseudo-qrels. On the other hand, we found that using the supervised machine learning ML technique was the best method proposed to generate pseudo-qrels so far. ML was based on training the supervised machine learning classifiers NB and SVM to classify the pool of documents retrieved by the TREC runs and therefore to expand the pseudo-qrels through two approaches. The first was called the two-class approach. It was used to classify the documents into either a "Relevant" class or a "Non-relevant" class. The second approach was the 50-class classification which classified the documents into one of the 50 classes, each representing a TREC topic. This led to better correlations between the overall system rankings using the different sets of qrels. But once again, none of these automatically generated qrels had the power to discriminate between the best systems as the best ones are evaluated in terms of recall since the aim of developing a new retrieval function is to return more relevant documents than the previous existing functions and therefore answer the user need more accurately. This is why it seemed natural to introduce the use of some real

or actual qrels, which were judged manually by human assessors, to our training sets so we could evaluate the impact of having known relevant documents on the performance of our ML technique in discriminating between the system overall and sub rankings, while maintaining the need to reduce the human effort and cost to build these new qrel sets. Chapter 6 will describe the new version of the ML technique which is called the AQML (Actual Qrels Machine Learning) technique, the process of selecting real qrels, using them as training sets and recording the minimum number of relevant documents required to have reliable results. We then show the experiments done on non-English test collections, using the CLEF 2003 French and Finnish collections.

Chapter 6 – Enhancing the ML Technique with real human-assessed qrels

6.1 Introduction

The research questions addressed in this thesis so far are all related to whether it is possible to produce pseudo-qrels for a test collection without any human intervention and whether these generated judgments could be considered reliable enough to be used for the evaluation of information retrieval systems and comparing between them. The efficiency of the pseudo-qrels can be determined by how well they can be used to rank a set of information retrieval systems with high correlations with the ranks obtained for the same systems using the human-built qrels. As for evaluating their effectiveness, this could be done in terms of the recall metric, which is a function of how many actual relevant documents these qrels include. We will use recall in this chapter to evaluate the TREC information retrieval systems and discriminate between the best ones.

Using the ML technique described in chapter 5, which required no human intervention, we were able to produce pseudo-qrels which were shown to be more efficient than any of the previous methods described in the literature. However, the ML fully automatic technique was unable to measure the recall and discriminate between the best systems. Thus, the enhancement of the ML technique proposed in this chapter is made by using real or actual known

(human-judged) qrels. The aim is to keep the number of known qrels used to a minimum in order to maintain low cost and reduced human effort when building a test collection. This enhanced version of the ML technique is referred to as the "Actual Qrels for Machine Learning" technique or AQML. The experiments conducted using the AQML technique seek to answer the third research question:

Q3. If it is not possible to form the qrels fully automatically, how many human-judged qrels should be supplied to start the process?

In the next section, we explain the pooling technique we used to select the actual qrels to use as a training set in our AQML method, then we report the overall system ranking correlations between the AQML ranking and the TREC ranking in section 6.3. In section 6.4, we discuss the power of the AQML pseudo-qrels to discriminate between the best systems and then we complete our evaluation in section 6.5 using the bpref-10 and infAP metrics which are commonly used in an incomplete judgments environment. We compare the results obtained from using these two metrics with those obtained from using the MAP measure to determine whether the correlations between the system rankings and the TREC rankings are greater with the bpref and infAP than with the MAP, as hypothesized in research question Q4:

Q4. Do bpref and infAP give more accurate system rankings than MAP when we have an incomplete set of judgments?

Before we conclude the chapter in section 6.7, we report the results we obtained when applying the AQML method to the French and Finnish CLEF-2003 test collections in section 6.6.

6.2 AQML Technique

When NIST forms the pools of retrieved documents, they are ordered alphabetically by document number before being distributed to the assessors to determine the relevance of the documents. Since the assessors have to judge all the documents, it does not really matter which document they examine first. However, in the AQML technique, we have to select only a few relevant documents to use as a training set. Instead of randomly selecting a document to judge from the ordered list, we applied the Losada et al. (2016) pooling technique (described in section 2.7.1.4), which has been shown to retrieve relevant documents faster than the traditional and other pooling techniques and then we test how many relevant documents we need to judge and use as a training set for the classifiers in order to produce a set of qels with reduced human effort which allow the discrimination between the best systems.

6.3 Overall system rankings

We applied the MM-NS technique described in section 2.7.1.4 to determine which documents should be judged, then formed a subset of actual or real qrels with 5, 10, 15, … 110 relevant documents as judged by the TREC

human judges in these pools. If a topic does not have that number of relevant documents in its pool, we selected all the relevant documents that could be found, so if topic 1 has only 12 relevant documents, when forming different subsets with more than 15 documents, we will keep all 12. We computed the MAP scores for the TREC runs using each of these subsets of actual qrels and we ranked them according to these scores. We then computed the Spearman's rho and Kendall's tau correlation values between the ranks obtained using the subset of qrels and the ones using the complete judgments set. The results obtained for the ad hoc test collections TREC-7 and TREC-8 are displayed in table 6.1 below.

TREC-7				**TREC-8**		
Subset of qrels with number of relevant documents equal to	Spearman	Kendall's tau		Subset of qrels with number of relevant documents equal to	Spearman	Kendall's tau
5	0.8902	0.7128		5	0.9301	0.7891
10	0.9337	0.7895		10	0.9670	0.8607
20	0.9568	0.8301		20	0.9743	0.8779
30	0.9684	0.8545		30	0.9775	0.8822
40	0.9770	0.8787		40	0.9800	0.8911
50	0.9827	0.8965		50	0.9847	0.9065
60	0.9867	0.9114		60	0.9875	0.9144
70	0.9882	0.9177		70	0.9894	0.9213
80	0.9917	0.9327		80	0.9914	0.9301
90	0.9943	0.9451		90	0.9920	0.9309
100	0.9934	0.9398		100	0.9929	0.9369
110	0.9942	0.9436		110	0.9945	0.9464

Table 6. 1: TREC-7 and TREC-8 Spearman's rho and Kendall's tau correlations between the subsets of qrels and the complete list of qrels.

When more than 20 relevant documents on average for all the 50 topics are available in the qrels, Kendall's tau becomes greater than 0.8 and therefore

we can consider that the rankings obtained are reliable, and hence the subset of qrels used can be considered reliable as well. These correlation values reported in the tables were significant as the p-value for each of the Spearman's rho and Kendall's tau scores was less than 0.05. Losada et al. did not test their technique using the TREC-6 collection, but we did in order to be able to compare the AQML results to the ML technique results.

The AQML technique expands each of the subsets of actual qrels by generating pseudo-qrels using machine learning classifiers: Naïve Bayes (NB) and Support Vector Machines (SVM). The classification for the documents follows the two-class approach of the ML technique and which leads to two classes: Relevant and Non-Relevant. The relevant documents in the subsets of the actual qrels are used to train the classifiers for the Relevant class and the non-relevant documents in the same subset of the actual qrels are used as a training set for the Non-Relevant class. After the classifiers were trained, we classified the pool of the documents formed by NIST because we knew their actual relevance and therefore we could evaluate our results.

We computed the MAP scores for the TREC runs using each set of pseudo-qrels produced, ranked them accordingly, and then computed the correlation measures between each ranking obtained and the gold standard ranking that was produced using the complete human-qrels.

The TREC-6 correlation values using the actual qrel subsets, and the new values obtained using the different expanded pseudo-qrels using the NB and SVM classifiers are reported in table 6.2. It required 10 relevant documents or more to obtain strong correlation values that were greater than 0.8, also with a p-value < 0.05. However, the subsets of actual qrels still had better correlations with the TREC complete set of judgments.

TREC-6						
	Actual Qrel Subsets		Pseudo-Qrels produced using NB		Pseudo-Qrels produced using SVM	
Subset of qrels with number of relevant documents equal to	Spearman ‘s rho	Kendall’s tau	Spearman’s rho	Kendall’s tau	Spearman’s rho	Kendall’s tau
5	0.9450	0.8242	0.9004	0.7397	0.9129	0.7542
10	0.9577	0.8559	0.9230	0.7768	0.9427	0.8037
20	0.9739	0.8872	0.9582	0.8362	0.9636	0.8547
30	0.9836	0.9220	0.9740	0.8750	0.9808	0.8932
40	0.9897	0.9387	0.9832	0.9031	0.9863	0.9083
50	0.9926	0.9478	0.9870	0.9146	0.9880	0.9132
60	0.9939	0.9528	0.9857	0.9087	0.9887	0.9176
70	0.9957	0.9616	0.9878	0.9148	0.9888	0.9163
80	0.9965	0.9627	0.9875	0.9138	0.9895	0.9206
90	0.9969	0.9671	0.9883	0.9157	0.9902	0.9231
100	0.9979	0.9760	0.9888	0.9182	0.9922	0.9339
110	0.9979	0.9743	0.9921	0.9366	0.9948	0.9517

Table 6. 2: TREC-6 Spearman’s rho and Kendall’s tau correlations after producing the subsets of pseudo-qrels using AQML

Similar behaviour was noticed for TREC-7. Even though the correlations obtained with the subsets of the actual qrels are better than the ones obtained with the fully-automatically generated pseudo-qrels, we can see that with 40 relevant documents on average, Kendall's tau became greater than 0.8 when using the SVM classifier and therefore we can consider the produced pseudo-qrels to be reliable.

TREC-7						
	Actual Qrel Subsets		Pseudo-Qrels produced using NB		Pseudo-Qrels produced using SVM	
Subset of qrels with number of relevant documents equal to	Spearman's rho	Kendall's tau	Spearman's rho	Kendall's tau	Spearman's rho	Kendall's tau
5	0.8903	0.7128	0.7760	0.6074	0.7421	0.5640
10	0.9337	0.7896	0.8066	0.6562	0.8336	0.6864
20	0.9684	0.8545	0.8425	0.6799	0.8968	0.7377
30	0.9827	0.8965	0.8973	0.7353	0.9380	0.7945
40	0.9882	0.9177	0.9379	0.7931	0.9736	0.8722
50	0.9943	0.9452	0.9795	0.8905	0.9859	0.9140
60	0.9942	0.9436	0.9857	0.9103	0.9876	0.9179
70	0.9955	0.9527	0.9880	0.9159	0.9898	0.9257
80	0.9967	0.9613	0.9898	0.9253	0.9903	0.9278
90	0.9969	0.9632	0.9914	0.9315	0.9911	0.9308
100	0.9982	0.9731	0.9915	0.9334	0.9915	0.9329
110	0.9987	0.9779	0.9922	0.9376	0.9918	0.9334

Table 6. 3: TREC-7 Spearman's rho and Kendall's tau correlations after producing the subsets of pseudo-qrels using AQML

For the TREC-8 test collection, the generated pseudo-qrels provided high correlations with 20 relevant documents or more. There was a small improvement in both the Spearman and Kendall correlations with 20 and 40 relevant documents when using the SVM. For the remaining values, the correlations between the real and pseudo generated qrels were very close. The difference was at most 0.02 as indicated in table 6.4 below.

TREC-8						
	Actual Qrels Subsets		Pseudo-Qrels produced using NB		Pseudo-Qrels produced using SVM	
Subset of qrels with number of relevant documents equal to	Spearman ’s rho	Kendall’s tau	Spearman’s rho	Kendall’s tau	Spearman’s rho	Kendall’s tau
5	0.8305	0.6691	0.7581	0.6180	0.7429	0.5887
10	0.9302	0.7892	0.8600	0.7103	0.8659	0.7126
20	0.9744	0.8780	0.9707	0.8708	**0.9782**	**0.8917**
30	0.9800	0.8911	0.9806	0.8956	**0.9865**	**0.9123**
40	0.9876	0.9144	0.9837	0.9029	**0.9886**	**0.9198**
50	0.9915	0.9301	0.9860	0.9155	**0.9913**	**0.9322**
60	0.9930	0.9370	0.9881	0.9190	**0.9926**	**0.9399**
70	0.9958	0.9536	0.9910	0.9347	0.9934	0.9440
80	0.9966	0.9585	0.9920	0.9426	0.9935	0.9450
90	0.9971	0.9629	0.9920	0.9438	0.9942	0.9492
100	0.9979	0.9689	0.9934	0.9507	0.9947	0.9512
110	0.9985	0.9749	0.9934	0.9509	0.9949	0.9525

Table 6. 4: TREC-8 Spearman’s rho and Kendall’s tau correlations after producing the subsets of pseudo-qrels using AQML

We applied the Wilcoxon test to compare between the Kendall's tau values obtained for the different subsets of: the actual qrels, the ones generated using the NB classifier and those resulting from the SVM. The statistical test shows that there is a significant difference between the tau values obtained from the subset of actual qrels and the pseudo-generated ones on all TREC test collections since the p-value was < 0.05, except for the TREC-8, where the p-value was of 0.2036 using SVM. We report the p-values in the table below:

p-value for the Wilcoxon test when comparing the Kendall's tau for the different subsets of qrels	
For TREC-6	
(Actual Qrels, NB Qrels)	**(Actual Qrels, SVM Qrels)**
4.88E-04	2.52E+03
For TREC-7	
4.88E-04	4.88E-04
For TREC-8	
9.77E-04	2.04E-01

Table 6. 5: p-value for the Wilcoxon test when comparing the Kendall's tau for the different subsets of qrels

The two-class approach when using the AQML technique which deploys a known set of relevant documents provided much better correlations than the two-class approach of the ML technique (section 5.4.3.1) as expected. We are using here a known set of relevant documents to train the classifiers rather than automatically selecting the set of documents which was retrieved by most of the systems for the topic and presuming it relevant. For all three

TREC test collections, a significant improvement in the correlations was seen. The best Kendall's tau value that could be achieved by an automatic technique for TREC-6 was 0.5887, while for the TREC-7 it was 0.5661 and for TREC-8 the tau value was 0.6095. After applying the AQML technique, Kendall's tau exceeded 0.8, the threshold defined by Voorhees (2001) to indicate a reliable ranking, with most of the subsets tested. This shows how important the relevant documents are and the impact they have on the evaluation of the systems. In the next section, we will show how using known relevant documents to produce the qrels using the AQML technique will also affect the discrimination between the best systems.

6.4 System subrankings

As we did with the ML technique, we divided the TREC systems into three groups according to their MAP scores obtained from using the human-built qrels: the top third, starting at rank 1, with the highest MAP score, as the best systems, the second third as average systems and the last third as the poor systems. We compared the Spearman correlations between the best, average and poor system rankings and the gold standard TREC rankings using 5, 10, 20, 30, …, 110 relevant documents. Tables 6.6, 6.7 and 6.8 below show the results for the TREC-6 best, average and poor systems consecutively.

	Best					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.9817	0.9203	0.9409	0.8043	0.9348	0.7826
10	0.9939	0.9638	0.9670	0.8623	0.9609	0.8478
20	0.9922	0.9565	0.9852	0.9203	0.9800	0.8986
30	0.9957	0.9710	0.9948	0.9638	0.9878	0.9348
40	0.9974	0.9783	0.9930	0.9493	0.9939	0.9565
50	0.9974	0.9783	0.9939	0.9565	0.9937	0.9546
60	0.9974	0.9783	0.9939	0.9565	0.9939	0.9565
70	0.9991	0.9928	0.9930	0.9493	0.9922	0.9493
80	0.9983	0.9855	0.9913	0.9420	0.9922	0.9493
90	0.9983	0.9855	0.9913	0.9420	0.9922	0.9493
100	0.9991	0.9928	0.9930	0.9493	0.9939	0.9565
110	0.9983	0.9855	0.9974	0.9783	0.9957	0.9710

Table 6. 6: Spearman's rho and Kendall's tau correlations between the **best** TREC-6 systems rankings using different numbers of relevant documents

	Average					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.8518	0.6759	0.8271	0.6443	0.8251	0.6522
10	0.8775	0.7233	0.9061	0.7391	0.8579	0.6970
20	0.9328	0.8024	0.9002	0.7549	0.8994	0.7604
30	0.9486	0.8419	0.9417	0.8261	0.9246	0.7921
40	0.9872	0.9368	0.9625	0.8577	0.9565	0.8498
50	0.9911	0.9526	0.9654	0.8577	0.9654	0.8577
60	0.9911	0.9605	0.9674	0.8656	0.9605	0.8498
70	0.9951	0.9684	0.9694	0.8814	0.9664	0.8656
80	0.9941	0.9605	0.9723	0.8972	0.9674	0.8735
90	0.9951	0.9684	0.9743	0.9051	0.9686	0.8792
100	0.9960	0.9763	0.9802	0.9130	0.9713	0.8893
110	0.9958	0.9743	0.9792	0.9289	0.9686	0.8951

Table 6. 7: Spearman's rho and Kendall's tau correlations between the **average** TREC-6 systems rankings using different numbers of relevant documents

	Poor					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.9122	0.7754	0.8765	0.7464	0.8617	0.7246
10	0.9419	0.8312	0.9104	0.7754	0.8989	0.7441
20	0.9730	0.8841	0.9365	0.8043	0.9348	0.7971
30	0.9839	0.9236	0.9704	0.8623	0.9609	0.8623
40	0.9852	0.9275	0.9748	0.8696	0.9774	0.8913
50	0.9861	0.9348	0.9774	0.8841	0.9783	0.8913
60	0.9939	0.9638	0.9783	0.8913	0.9757	0.8696
70	0.9904	0.9493	0.9783	0.8913	0.9722	0.8768
80	0.9913	0.9565	0.9800	0.9058	0.9722	0.8768
90	0.9913	0.9565	0.9800	0.9058	0.9741	0.8748
100	0.9967	0.9764	0.9815	0.9111	0.9728	0.8820
110	0.9967	0.9764	0.9889	0.9401	0.9809	0.9058

Table 6. 8: Spearman's rho and Kendall's tau correlations between the **poor** TREC-6 systems rankings using different numbers of relevant documents

It was clear that the pseudo-qrels produced with the AQML had a better discrimination power than the ones produced using the automatic ML technique, but the actual qrels were still better in discriminating between the best systems. The Spearman correlations for the average and poor systems started with a value close to 0.8 (p-value < 0.05) and then as more relevant documents were added, the correlations became higher. The pseudo-qrels produced using the SVM classifier led to better correlations than the ones generated with the NB classifier. We needed on average 10 relevant documents to generate pseudo-qrels that could reliably discriminate between the best systems, and nearly 30 documents to have a reliable discrimination between the average and poor systems.

However, for TREC-7, the correlations between the best systems' rankings and the gold standard were almost the same as for the pseudo-qrels produced using the AQML technique, and when using the subsets of actual qrels with more than 50 relevant documents. Similar behaviour was noticed for the average systems, but the subsets of actual qrels were better at discriminating between the poor systems as reported in Tables 6.9, 6.10 and 6.11 below, p-value < 0.05 for all the reported results. The Spearman values were below 0.8 for TREC-7 when only 5 to 10 documents were used for training the classifiers, but the more relevant documents we trained the classifiers with, the higher the correlations became.

	Best					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.9050	0.7398	0.7888	0.6292	0.8184	0.6637
10	0.9111	0.7754	0.8460	0.6982	0.8460	0.6827
20	0.9655	0.8574	0.9132	0.7683	0.8344	0.6827
30	0.9835	0.9073	0.9596	0.8457	0.9132	0.7683
40	0.9890	0.9287	0.9725	0.8788	0.9352	0.7932
50	0.9921	0.9394	0.9850	0.9251	0.9804	0.9037
60	0.9927	0.9465	0.9893	0.9358	0.9875	0.9251
70	0.9951	0.9572	0.9899	0.9323	0.9866	0.9216
80	0.9969	0.9715	0.9902	0.9323	0.9908	0.9420
90	0.9972	0.9715	0.9908	0.9394	0.9924	0.9501
100	0.9979	0.9786	0.9942	0.9572	0.9933	0.9537
110	0.9979	0.9786	0.9948	0.9608	0.9951	0.9643

Table 6. 9: Spearman's rho and Kendall's tau correlations between the **best** TREC-7 systems rankings using different numbers of relevant documents

	Average					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.8612	0.6995	0.6984	0.5327	0.7160	0.5782
10	0.9468	0.8254	0.8030	0.7071	0.7942	0.6957
20	0.9752	0.8777	0.8818	0.7336	0.8236	0.6616
30	0.9837	0.9118	0.9225	0.7716	0.8806	0.7223
40	0.9869	0.9270	0.9720	0.8701	0.9291	0.7780
50	0.9952	0.9611	0.9872	0.9270	0.9743	0.8891
60	0.9925	0.9450	0.9875	0.9308	0.9886	0.9270
70	0.9951	0.9611	0.9875	0.9308	0.9911	0.9450
80	0.9971	0.9725	0.9875	0.9308	0.9891	0.9374
90	0.9961	0.9687	0.9899	0.9383	0.9894	0.9308
100	0.9976	0.9763	0.9897	0.9384	0.9877	0.9308
110	0.9983	0.9829	0.9914	0.9460	0.9874	0.9270

Table 6. 10: Spearman's rho and Kendall's tau correlations between the **average** TREC-7 systems rankings using different numbers of relevant documents

	Poor					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.8775	0.7184	0.7730	0.5888	0.8154	0.6435
10	0.9243	0.7922	0.8906	0.7077	0.8646	0.6827
20	0.9618	0.8681	0.9221	0.7600	0.9242	0.7504
30	0.9710	0.8895	0.9446	0.8029	0.9362	0.7922
40	0.9850	0.9180	0.9734	0.8717	0.9545	0.8253
50	0.9933	0.9501	0.9814	0.9002	0.9774	0.8859
60	0.9936	0.9501	0.9861	0.9232	0.9798	0.8992
70	0.9942	0.9537	0.9850	0.9144	0.9811	0.9002
80	0.9927	0.9501	0.9843	0.9135	0.9823	0.9054
90	0.9951	0.9608	0.9829	0.9073	0.9832	0.9073
100	0.9960	0.9679	0.9826	0.9054	0.9826	0.9037
110	0.9972	0.9715	0.9807	0.8966	0.9832	0.9073

Table 6. 11: Spearman's rho and Kendall's tau correlations between the **poor** TREC-7 systems rankings using different numbers of relevant documents

The best results were achieved with TREC-8 where the use of 25 relevant documents produced similar correlations between the system rankings using the pseudo-qrels produced by the AQML and the those produced by the TREC qrels, the subsets of actual qrels including 5, 10, 20, etc. relevant documents and the complete set of TREC qrels. Even slightly better correlations were noticed with the AQML qrels with 20 to 30 relevant documents as shown in tables 6.12, 6.13 and 6.14 below. This could have resulted from the fact that some of the relevant documents we used for training contain more relevant information about the topic than others and they could lead to finding more relevant documents which we could not find earlier using the automatic technique.

	Best					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.8410	0.6856	0.6811	0.5532	0.7057	0.6074
10	0.9331	0.7979	0.8156	0.6818	0.8070	0.6911
20	0.9715	0.8746	0.9682	0.8676	0.9710	0.8600
30	0.9778	0.8885	0.9888	0.9233	0.9870	0.9187
40	0.9822	0.9001	0.9906	0.9350	0.9846	0.8978
50	0.9867	0.9164	0.9921	0.9396	0.9909	0.9326
60	0.9875	0.9187	0.9929	0.9443	0.9914	0.9303
70	0.9912	0.9396	0.9935	0.9466	0.9948	0.9529
80	0.9921	0.9419	0.9937	0.9466	0.9947	0.9559
90	0.9924	0.9443	0.9943	0.9535	0.9940	0.9529
100	0.9938	0.9535	0.9940	0.9529	0.9956	0.9605
110	0.9959	0.9628	0.9953	0.9582	0.9958	0.9628

Table 6. 12: Spearman's rho and Kendall's tau correlations between the **best** TREC-8 systems rankings using different numbers of relevant documents

	Average					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.7130	0.5519	0.6115	0.4579	0.6214	0.4896
10	0.9127	0.7721	0.8414	0.6646	0.8260	0.6597
20	0.9728	0.8755	0.9798	0.9072	0.9702	0.8772
30	0.9810	0.9121	0.9818	0.9114	0.9769	0.8943
40	0.9867	0.9334	0.9877	0.9236	0.9804	0.9078
50	0.9926	0.9505	0.9914	0.9444	0.9821	0.9096
60	0.9934	0.9554	0.9934	0.9585	0.9855	0.9219
70	0.9965	0.9683	0.9929	0.9536	0.9871	0.9328
80	0.9967	0.9707	0.9934	0.9585	0.9876	0.9389
90	0.9970	0.9756	0.9927	0.9609	0.9882	0.9414
100	0.9974	0.9780	0.9932	0.9652	0.9906	0.9512
110	0.9977	0.9829	0.9927	0.9585	0.9901	0.9487

Table 6. 13: Spearman's rho and Kendall's tau correlations between the **average** TREC-8 systems rankings using different numbers of relevant documents

	Poor					
	Actual		SVM		NB	
Nb. Relevant docs	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
5	0.8905	0.7605	0.8466	0.7007	0.8854	0.7583
10	0.9390	0.8271	0.9007	0.7643	0.9026	0.7672
20	0.9706	0.8847	0.9683	0.8803	0.9672	0.8736
30	0.9724	0.8914	0.9733	0.8869	0.9760	0.8914
40	0.9866	0.9313	0.9790	0.9024	0.9825	0.9113
50	0.9884	0.9313	0.9836	0.9168	0.9798	0.9069
60	0.9885	0.9335	0.9843	0.9224	0.9822	0.9129
70	0.9928	0.9512	0.9867	0.9313	0.9833	0.9174
80	0.9932	0.9512	0.9865	0.9285	0.9870	0.9357
90	0.9937	0.9534	0.9869	0.9313	0.9846	0.9262
100	0.9956	0.9623	0.9888	0.9379	0.9858	0.9290
110	0.9967	0.9690	0.9891	0.9401	0.9850	0.9246

Table 6. 14: Spearman's rho and Kendall's tau correlations between the **poor** TREC-8 systems rankings using different numbers of relevant documents

In all cases, we can see a large improvement in the power of discrimination in each category of systems when using the AQML pseudo-qrels rather than the ML qrels. Because our main interest was to discriminate between the best systems and we were able to obtain better Kendall's tau values when using some known relevant documents to train the classifiers, we apply the non-parametric Wilcoxon test to the tau values for the different TREC test collections, since they are not normally distributed. The p-value for the statistical test is listed in the table below:

p-value for the Wilcoxon test when comparing the Kendall's tau for the different subsets of qrels for the best TREC systems	
For TREC-6 Best Systems	
(Actual Qrels, NB Qrels)	**(Actual Qrels, SVM Qrels)**
0.002478	0.002497
For TREC-7 Best Systems	
0.0004883	0.0004883
For TREC-8 Best Systems	
0.7557	0.5301

Table 6. 15: p-value for the Wilcoxon test when comparing the Kendall's tau for the different subsets of qrels for the best TREC systems

While there is a significant difference between the Kendall's tau obtained for the best systems on both TREC-6 and TREC-7, this behaviour seems to change on TREC-8 since the p-value is > 0.5 indicating that there is some similarity between the two sets; when we used between 20 to 60 relevant documents, the correlations obtained from the pseudo-qrels were better than the ones computed using the subsets of actual qrels on TREC-8, so

even though this TREC has more difficult topics than the other two test collections, the improvement is significant.

The experiments done in this section aimed to answer the research question Q3 where we asked how many relevant documents we need to produce a reliable set of qrels that could discriminate between the best systems. However, the results we obtained show that we cannot give a specific value that could work for any test collection, but we can say that the minimum number of relevant documents that could produce reliable results is 20 on average.

6.5 Other evaluation metrics: bpref and infAP

The ML technique does not use any relevance judgments, and thus provides an approach to fully automatically generate the pseudo-qrels. The AQML technique uses a few relevant documents as a training set in order to expand the pseudo-qrels. There has been some previous work investigating measures other than the MAP metric we used which could be more robust in the case of incomplete judgments. These measures do not take into consideration whether documents are actually judged relevant by human assessors or whether they are merely assumed to be relevant or assumed to be non-relevant because they were unjudged. Two of the common measures used in such environments are the binary preference (bpref) which was described in section 3.2.2 and the inferred Average Precision (infAP) described in section 3.2.3.

6.5.1 Evaluation results

The AQML produces pseudo-qrels based on a subset of actual known qrels, therefore, we evaluate the systems with incomplete judgments using both bpref-10 and infAP and we compare the results with the ones obtained from using the subset of actual qrels. After ranking the TREC systems according to each of these measures, we measured the Kendall's tau correlation between the rankings obtained. The tables below show the graphs for the bpref-10 and infAP with the different numbers of relevant documents used.

The Kendall's tau values based on the bpref-10 measure for all three TREC test collections are higher for the AQML SVM pseudo-qrels than the tau values for the ones generated with the NB classifier. Table 6.14 below shows that the correlation values go above 0.8 with more than 20 relevant documents used when evaluating using bpref-10 and more than 30 with infAP scores.

TREC-6 Spearman's rho and Kendall's tau based on bpref-10						
Number of relev. docs. Used	**Subsets of Actual Qrels**		**Pseudo-qrels produced with NB classifier**		**Pseudo-qrels produced with SVM classifier**	
	ρ	τ	ρ	τ	ρ	τ
5	0.8579	0.7007	0.8478	0.6730	0.8410	0.6748
10	0.8267	0.6631	0.8374	0.6565	0.8820	0.7131
20	0.9154	0.7735	0.9088	0.7540	0.9370	0.7969
30	0.9380	0.8113	0.9493	0.8209	0.9576	0.8352
40	0.9615	0.8527	0.9608	0.8362	0.9602	0.8390
50	0.9768	0.8887	0.9665	0.8527	0.9667	0.8567
60	0.9764	0.8894	0.9689	0.8598	0.9630	0.8515
70	0.9808	0.9024	0.9667	0.8567	0.9654	0.8649
80	0.9865	0.9168	0.9691	0.8641	0.9720	0.8786

90	0.9880	0.9204	0.9712	0.8679	0.9727	0.8826
100	0.9887	0.9265	0.9786	0.8926	0.9769	0.8907
110	0.9910	0.9416	0.9831	0.9086	0.9808	0.9020

(a)

TREC-6 Spearman's rho and Kendall's tau based on infAP						
Number of relev. docs. Used	**Subsets of Actual Qrels**		**Pseudo-qrels produced with NB classifier**		**Pseudo-qrels produced with SVM classifier**	
	ρ	τ	ρ	τ	ρ	τ
5	0.9450	0.8242	0.9004	0.7397	0.9129	0.7542
10	0.9577	0.8559	0.9230	0.7768	0.9427	0.8037
20	0.9739	0.8872	0.9582	0.8362	0.9636	0.8547
30	0.9836	0.9220	0.9740	0.8750	0.9808	0.8932
40	0.9897	0.9385	0.9832	0.9031	0.9863	0.9083
50	0.9926	0.9478	0.9870	0.9146	0.9880	0.9132
60	0.9939	0.9528	0.9857	0.9087	0.9887	0.9176
70	0.9957	0.9616	0.9878	0.9148	0.9888	0.9163
80	0.9965	0.9627	0.9875	0.9138	0.9895	0.9206
90	0.9969	0.9671	0.9883	0.9157	0.9902	0.9231
100	0.9979	0.9760	0.9888	0.9182	0.9922	0.9339
110	0.9979	0.9743	0.9921	0.9366	0.9948	0.9517

(b)

Table 6. 16: TREC-6 Spearman's rho and Kendall's tau correlations based on (a) bpref-10 and (b) infAP measures

For TREC-7, using 20 relevant documents for the training set was enough to obtain a tau value > 0.8. However, to get better correlations that the actual qrels, we needed more relevant documents, 60 using bpref-10. For the infAP, we could not achieve better scores than the subsets of actual qrels. With TREC-7, the subsets of actual qrels seem to produce lower correlations than the SVM generated ones and the NB pseudo-qrels with less than 70 relevant documents used when evaluating using bpref-10. As for the results we got using the infAP measure, as Table 6.16 (b) and 6.17 (b) show, the

subsets of actual qrels were always better than the newly produced ones using the AQML SVM and NB classifiers

TREC-7 Spearman's rho and Kendall's tau based on bpref-10						
Number of relev. docs. Used	**Subsets of Actual Qrels**		**Pseudo-qrels produced with NB classifier**		**Pseudo-qrels produced with SVM classifier**	
	ρ	τ	ρ	τ	ρ	τ
5	0.6613	0.5117	0.8618	0.6846	0.8591	0.6829
10	0.6964	0.5289	0.8646	0.6806	0.9247	0.7740
20	0.9020	0.7484	0.9549	0.8269	0.9621	0.8404
30	0.9494	0.8248	0.9690	0.8645	0.9776	0.8831
40	0.9734	0.8701	0.9762	0.8737	0.9886	0.9167
50	0.9863	0.9086	0.9882	0.9155	0.9934	0.9398
60	0.9909	0.9276	0.9913	0.9297	0.9947	0.9460
70	0.9948	0.9478	0.9938	0.9417	0.9957	0.9537
80	0.9964	0.9570	0.9949	0.9478	0.9957	0.9507
90	0.9977	0.9686	0.9949	0.9488	0.9958	0.9501
100	0.9980	0.9705	0.9950	0.9506	0.9965	0.9573
110	0.9986	0.9758	0.9952	0.9510	0.9966	0.9586

(a)

TREC-7 Spearman's rho and Kendall's tau based on infAP						
Number of relev. docs. used	**Subsets of Actual Qrels**		**Pseudo-qrels produced with NB classifier**		**Pseudo-qrels produced with SVM classifier**	
	ρ	τ	ρ	τ	ρ	τ
5	0.8903	0.7128	0.7760	0.6074	0.7421	0.5640
10	0.9337	0.7896	0.8066	0.6562	0.8336	0.6864
20	0.9684	0.8545	0.8425	0.6799	0.8968	0.7377
30	0.9827	0.8965	0.8973	0.7353	0.9380	0.7945
40	0.9882	0.9177	0.9379	0.7931	0.9736	0.8722
50	0.9943	0.9452	0.9795	0.8905	0.9859	0.9140
60	0.9942	0.9436	0.9857	0.9103	0.9876	0.9179
70	0.9955	0.9527	0.9880	0.9159	0.9898	0.9257
80	0.9967	0.9613	0.9898	0.9253	0.9903	0.9278
90	0.9969	0.9632	0.9914	0.9315	0.9911	0.9308
100	0.9982	0.9731	0.9915	0.9334	0.9915	0.9329
110	0.9987	0.9779	0.9922	0.9376	0.9918	0.9334

(b)

Table 6. 17: TREC-7 Spearman's rho and Kendall's tau correlations based on (a) bpref-10 and (b) infAP measures

Table 6.18 shows that for TREC-8, a tau value > 0.8 could be achieved using more than 20 relevant documents and we could already obtain better results than using the subsets of actual qrels. The AQML SVM pseudo-qrels provide better correlations than the other two sets when using 20 to 60 relevant documents.

TREC-8 Spearman's rho and Kendall's tau based on bpref-10						
Number of relev. docs. Used	**Subsets of Actual Qrels**		**Pseudo-qrels produced with NB classifier**		**Pseudo-qrels produced with SVM classifier**	
	ρ	τ	ρ	τ	ρ	τ
5	0.6511	0.4803	0.8274	0.6512	0.9024	0.7343
10	0.6922	0.5297	0.8296	0.6498	0.9046	0.7430
20	0.9317	0.7939	0.9169	0.7733	0.9653	0.8497
30	0.9626	0.8427	0.9507	0.8262	0.9697	0.8609
40	0.9757	0.8779	0.9611	0.8450	0.9820	0.8941
50	0.9834	0.8995	0.9709	0.8646	0.9847	0.9018
60	0.9871	0.9141	0.9756	0.8769	0.9854	0.9074
70	0.9902	0.9256	0.9788	0.8886	0.9849	0.9077
80	0.9929	0.9392	0.9818	0.8963	0.9849	0.9071
90	0.9961	0.9569	0.9850	0.9128	0.9861	0.9149
100	0.9976	0.9663	0.9863	0.9160	0.9874	0.9188
110	0.9979	0.9703	0.9864	0.9150	0.9872	0.9210

(a)

TREC-8 Spearman's rho and Kendall's tau based on infAP						
Number of relev. docs. Used	**Subsets of Actual Qrels**		**Pseudo-qrels produced with NB classifier**		**Pseudo-qrels produced with SVM classifier**	
	ρ	τ	ρ	τ	ρ	τ
5	0.8305	0.6691	0.7581	0.6180	0.7429	0.5887
10	0.9302	0.7892	0.8600	0.7103	0.8659	0.7126
20	0.9744	0.8780	0.9707	0.8708	0.9782	0.8917
30	0.9800	0.8911	0.9806	0.8956	0.9865	0.9123
40	0.9876	0.9144	0.9837	0.9029	0.9886	0.9198
50	0.9915	0.9301	0.9860	0.9155	0.9913	0.9322

60	0.9930	0.9370	0.9881	0.9190	0.9926	0.9399
70	0.9958	0.9536	0.9910	0.9347	0.9934	0.9440
80	0.9966	0.9585	0.9916	0.9413	0.9935	0.9450
90	0.9971	0.9629	0.9920	0.9438	0.9942	0.9492
100	0.9979	0.9689	0.9933	0.9503	0.9947	0.9512
110	0.9985	0.9749	0.9934	0.9509	0.9949	0.9525

(b)

Table 6. 18: TREC-8 Spearman's rho and Kendall's tau correlations based on (a) bpref-10 and (b) infAP measures

These tables confirm the results we obtained with the overall system rankings and when studying the power of discrimination of each set of qrels and pseudo-qrels generated. Even though in some cases, the actual qrels provide higher correlations, the quality of the pseudo-qrels generated is still reliable since they lead to correlations greater than 0.8 with a p-value < 0.05 indicating that the results are significant. Similarly to the previous sections, we performed the non-parametric Wilcoxon test to measure how significant the difference in the Kendall's tau values is between the subsets of actual qrels and the pseudo-qrels formed using the ML approach with both NB and SVM classifiers when using bpref and infAP measures. Table 6.19 below summarizes the p-values obtained:

	p-value for the Wilcoxon test when comparing the Kendall's tau for the different subsets of qrels			
	bpref-10		**infAP**	
	(Actual Qrels, NB Qrels)	**(Actual Qrels, SVM Qrels)**	**(Actual Qrels, NB Qrels)**	**(Actual Qrels, SVM Qrels)**
TREC-6	0.001465	0.09229	0.0004883	0.002516
TREC-7	**0.4697**	0.06396	0.0004883	0.0004883
TREC-8	**0.2334**	**0.8501**	0.0009766	**0.2036**

Table 6. 19: p-value for the Wilcoxon test when comparing the Kendall's tau for the different subsets of qrels based on bpref-10 and infAP measures

In TREC-8, the difference between the Kendall's tau values computed based on bpref-10 and infAP is not significant while for the TREC-6 and TREC-7, there is a significant difference except for TREC-7 between the actual qrels and the ones produced using the NB classifier based on bpref-10. When the Kendall's tau values obtained from using the pseudo-qrels outperformed the correlations using the actual qrels, the difference was not significant which shows that the results are somehow similar and this is what we want when we compare between the baseline and the results from our proposed AQML technique.

Because introducing some real qrels to the training process of the machine learning classifiers improved both the overall and the sub rankings of the TREC systems when compared with the rankings that we obtained from using a fully automated ML technique, we wanted to test how well these techniques performed on non-English test collections, and for these experiments we used the Finnish and French test collections from CLEF 2003.

6.6 Non-English test collection CLEF 2003 Experiments

The work described in this section is designed to answer research question Q5:

Q5. ***How well do the techniques developed in this thesis work for languages other than English such as French and Finnish?***

6.6.1 Introducing the CLEF test collections

The Cross-Language Evaluation Forum (CLEF) is a series of evaluation activities that were supported by the Information Society Technologies program of the European Union. The CLEF consortium operates on European languages and information retrieval tasks are supported. The test collections are either monolingual or in cross-language contexts. CLEF uses the same methodology for evaluation as TREC, and this is why we decided to run the experiments using one of the CLEF test collections. A CLEF test collection consists of a set of documents, topics and relevance assessments. We selected the CLEF 2003 (Braschler, 2003) campaigns for our experiments because it groups several languages Dutch, Finnish, French, German, Italian, Russian, Spanish and Swedish. We tested the ML and AQML approaches on the French test collection since it shares its Latin stems with the English, and the Finnish test collection which has different stems and is not similar to any other European language.

6.6.2 Experimental Design

The 2003 French test collection has a list of 60 topics numbered from 141 to 200 and a set of news articles from Le Monde 1994 and French SDA 1994 (Swiss news agency data - Schweizerische Depeschenagentur) and SDA 1995 which form the set of documents, in addition to the relevance assessments which were produced by human annotators. The Finnish test collection consists of 60 topics as well, also numbered from 141 to 200, and

the documents' source is Aamulehti, a daily morning newspaper, from 1994 - 1995.

6.6.2.1 Retrieval systems

Since the initial runs submitted to CLEF were not available, we used the 24 different weighting models offered by Terrier as surrogates for different retrieval systems and listed in section 3.3.

We stemmed the documents using the SnowballStemmer[16] for the French and Finnish languages when retrieving the documents and when classifying them.

6.6.2.2 Document Selection and classification

We applied both the ML and AQML techniques to produce the set of qrels for the CLEF 2003 Finnish and French test collections. To test the ML technique which generates qrels automatically without any human intervention, we first ran the 24 retrieval models in Terrier, and then applied the NIST pooling technique, selecting the top 100 documents retrieved from all the models for each topic, removing the duplicates and then ordering the documents by their IDs. Next, we counted the number of models that retrieved each document in the pool, and we selected the documents retrieved by more than S% of the models for each topic, as was done for the TREC collections. The problem we faced was that the retrieval models gave very similar retrieval results because these are relatively small test collections and

[16] http://www.nltk.org/_modules/nltk/stem/snowball.html

therefore for most of the topics, there were too many documents retrieved by all the 24 models and which led to the problem of the "tyranny of the mass" or having popularity take precedence over performance.

Thus, a variation on the document selection method was used, which was to select the top document retrieved by each model for each topic. This document had a high probability of being relevant to the topic since it was ranked first in the retrieval result by at least one system. The combination of the top documents from all the retrieval models formed the "RelSet" which was used to train the classifiers for relevant documents. To build the training set for non-relevant documents, we selected the last document retrieved for each topic from all the 24 models. The set of the documents retrieved last from all the retrieval models constitute the "NonRelSet". Figure 6.1 shows the steps taken for document selection.

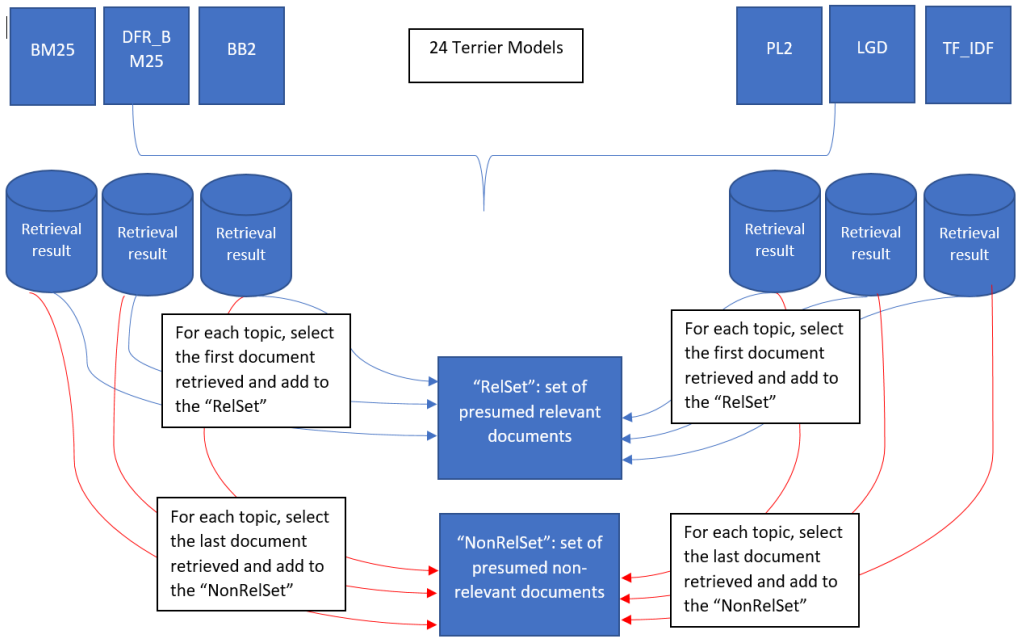


Figure 6. 1: Training document selection process

We applied the ML technique, by training the NB and SVM classifiers using the "RelSet" and "NonRelSet" described above. The pool of documents we classified was the same pool that was formed using the TREC pooling technique and which was given to the human assessors to judge in order to build the qrels for the test collection. First, we used a two-class approach which classifies the documents into two categories "Relevant" and "Non-Relevant". We then applied a second classification approach, which we called the 50-class, and which trains the classifiers using the "RelSet" only and labels the documents using the topic ID. The classification of the documents in the pool then labelled each document with a topic ID. These documents were considered relevant to the topic while all other documents retrieved but not labelled by the classifier were considered non-relevant. These two approaches were first described in sections 5.4.3.1 and 5.4.3.2 respectively.

For the AQML technique, we could not apply the same steps described earlier for TREC since several of these topics had no relevant documents in the actual qrels and the number of relevant documents for the remaining topics did not exceed 40 for the Finnish test collection, except for topic number 197. For the French test collection, the number of relevant documents did not exceed 35 except for topics 164, 181 and 197. Therefore, we excluded the topics which did not have any judged relevant documents in the qrels. The range of relevant documents retrieved and used for training was from 1 to 17 for the French test collection, since the average number of relevant

documents for all the topics is 17. As for the Finnish test collection, the range of relevant documents used was from 1 to 10, since the average number of relevant documents for all topics is 10.

We used the same 24 Terrier retrieval models as before, and for evaluation we approached the problem in the same manner as was done for the TREC collections, which is by ranking the different models using three different measures MAP, bpref-10 and infAP and then measuring both the Spearman's rho and Kendall's tau correlations between the system rankings obtained and the CLEF gold standard rankings. The significance of the results was indicated by the p-value for the Spearman's rho and Kendall's tau. Any time we obtained a p-value < 0.05, the correlations we obtained were considered significant. The results are discussed in the next section.

6.6.3 Evaluation

Using the qrels generated by the ML technique for the French test collection, the overall system rankings' correlations are as shown in table 6.20 below.

	Two-class classification using NB, alpha 0.1		**Two-class classification using SVM**		**50-class classification using NB, alpha 0.1**		**50-class classification using SVM**	
	Kendall's tau	Spearman	Kendall's tau	Spearman	Kendall's tau	Spearman	Kendall's tau	Spearman
French 2003	0.2618	0.3649	0.3666	0.5240	**0.5480**	0.6918	0.5408	**0.6936**

Table 6. 20: Spearman's rho and Kendall's tau correlations for the overall system rankings for the French collection using the ML technique

The overall retrieval model rankings were positively correlated with the rankings obtained from using the human-built qrels for the French test collection 2003. As for the Finnish test collection, we obtained positive correlations for Kendall's tau and Spearman's rho of 0.2218 and 0.3197 respectively. However, the corresponding p-values were 0.13 and 0.127 respectively which were not statistically significant. To understand these correlation values, we looked at the actual qrels, the ones judged by human assessors, and we noticed that for several topics in the Finnish test collection there are no documents judged relevant (topics: 141, 144, 145, 146, 160, 167, 169, 175,182, 186, 188, 189, 191, 194, 195), and that the same is true for the French test collection (topics: 146, 160, 161, 166, 169, 172, 191, 194). Other topics have only one or two relevant documents which makes them all hard topics. The number of relevant documents judged for each topic in the Finnish and French test collections are reported in the table 6.21 below.

Topic Number	Number of relevant documents		Topic Number	Number of relevant documents		Topic Number	Number of relevant documents	
	French	Finnish		French	Finnish		French	Finnish
141	1	0	**161**	0	4	**181**	193	82
142	13	8	**162**	25	5	**182**	12	0
143	28	36	**163**	11	3	**183**	5	2
144	3	0	**164**	89	26	**184**	31	1
145	14	0	**165**	2	2	**185**	1	1
146	0	0	**166**	0	1	**186**	13	0
147	8	6	**167**	2	0	**187**	7	2
148	2	6	**168**	13	12	**188**	3	0
149	5	1	**169**	0	0	**189**	2	0

150	19	4	**170**	1	1	**190**	17	10
151	10	2	**171**	3	2	**191**	0	0
152	27	3	**172**	0	5	**192**	19	10
153	12	1	**173**	3	2	**193**	41	20
154	2	16	**174**	7	3	**194**	0	0
155	2	6	**175**	1	0	**195**	19	0
156	1	3	**176**	19	2	**196**	1	2
157	3	12	**177**	7	14	**197**	131	63
158	2	15	**178**	3	3	**198**	3	2
159	19	6	**179**	8	18	**199**	33	17
160	0	0	**180**	32	22	**200**	18	21

Table 6. 21: Number of relevant documents per topic for CLEF 2003 Finnish and French test collections

Therefore, our automatic selection for the "RelSet" had considered documents which are non-relevant as relevant, and this had led to such low correlations. When using the AQML technique the correlation values improved to give p-values < 0.05, the highest p-value obtained was 0.0104, while all other p-values were less than 10^{-8}.

The correlations between the rankings based on the MAP scores and the CLEF standard rankings obtained for the French test collection are shown in table 6.22 below. Clearly using the classifiers SVM and NB improved the correlations between the system rankings.

Number of relevant docs used	**Qrels Subsets**		**SVM**		**NB**		**NB with alpha 0.1**	
	P	τ	ρ	τ	ρ	τ	ρ	τ
1	0.7997	0.6425	0.8354	0.6860	0.7684	0.6207	0.7910	0.6788
2	0.9215	0.7877	0.8076	0.6715	0.9398	0.8385	0.8293	0.6933
3	0.9302	0.8094	0.8911	0.7731	0.8954	0.7731	0.9372	0.8457
4	0.9528	0.8457	0.9426	0.8509	0.8937	0.8022	0.9432	0.8530
5	0.9519	0.8530	0.9652	0.9018	0.9537	0.8748	0.9667	0.8893
6	0.9624	0.8675	0.9763	0.9256	0.9772	0.9256	0.9448	0.8582
7	0.9746	0.8966	0.9772	0.9290	0.9841	0.9401	0.9650	0.8893

8	0.9652	0.8800	0.9920	0.9619	0.9841	0.9329	0.9789	0.9256
9	0.9572	0.8675	0.9920	0.9619	0.9896	0.9564	0.9780	0.9183
10	0.9774	0.9018	0.9920	0.9619	0.9922	0.9636	0.9848	0.9309
11	0.9772	0.9111	0.9911	0.9546	0.9937	0.9619	0.9911	0.9546
12	0.9806	0.9183	0.9867	0.9401	0.9937	0.9619	0.9911	0.9546
13	0.9720	0.8893	0.9867	0.9401	0.9937	0.9619	0.9841	0.9329
14	0.9606	0.8748	0.9833	0.9256	0.9928	0.9546	0.9893	0.9546
15	0.9606	0.8748	0.9859	0.9329	0.9850	0.9256	0.9911	0.9546
16	0.9572	0.8603	0.9682	0.9018	0.9859	0.9256	0.9780	0.9183
17	0.9580	0.8675	0.9752	0.9177	0.9893	0.9474	0.9761	0.9164

Table 6. 22: Spearman's rho (ρ) and Kendall's tau (τ) correlations for the overall system rankings for the French collection using the AQML technique with different numbers of relevant documents

We obtained Kendall's tau correlations higher than 0.8 starting with 2 relevant documents. When adjusting the smoothing parameter alpha in the Naïve Bayes classifier to 0.1 instead of the default value of 1.0, the classifier worked better than the SVM and the default NB in a few cases.

As for the Finnish test collection, the correlations also computed based on the MAP scores of the retrieval models are reported in table 6.23 below:

Number of relevant docs used	**Qrels Subsets**		**SVM**		**NB**		**NB with alpha 0.1**	
	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall	Spearman	Kendall
1	0.8417	0.6727	0.7903	0.6364	0.7903	0.6364	0.5128	0.3673
2	0.8547	0.6873	0.8706	0.7359	0.8730	0.7236	0.7234	0.5636
3	0.8884	0.7432	0.9371	0.8233	0.9374	0.8182	0.8712	0.7236
4	0.8834	0.7309	0.9356	0.8400	0.9079	0.7409	0.9600	0.8764
5	0.9487	0.8327	0.9593	0.8869	0.9226	0.7964	0.9574	0.8473
6	0.9435	0.8255	0.9658	0.8889	0.9652	0.8836	0.9641	0.8816
7	0.9278	0.7964	0.9739	0.9055	0.9739	0.9055	0.9739	0.9055
8	0.9293	0.8087	0.9730	0.8982	0.9730	0.8982	0.9739	0.9055
9	0.9382	0.8358	0.9739	0.9055	0.9774	0.9127	0.9748	0.9055
10	0.9519	0.8452	0.9739	0.9055	0.9719	0.9035	0.9748	0.9055

Table 6. 23: Spearman's rho and Kendall's tau correlations for the overall system rankings for the Finnish collection using the AQML technique with different numbers of relevant documents

The correlations also improved after using the classifiers, and the SVM seemed to provide better results than NB in most cases. At least 3 relevant documents were required to get a Kendall's tau value higher than 0.8. The experiments conducted on the CLEF non-English test collections confirm the findings we obtained with TREC-8 and in some cases for TREC-7 and TREC-6.

Similar to the results obtained for the TREC test collections, the best systems' Spearman correlations showed negative values when using the two-class approach with both classifiers for the French and Finnish collections. We were able to discriminate between the best systems using the ML automatic technique using the second approach, the 50-class, rather than the two-class approach.

The actual values of the Kendall's tau (τ) and Spearman (ρ) correlations are included in table 6.24 below figure 6.2.

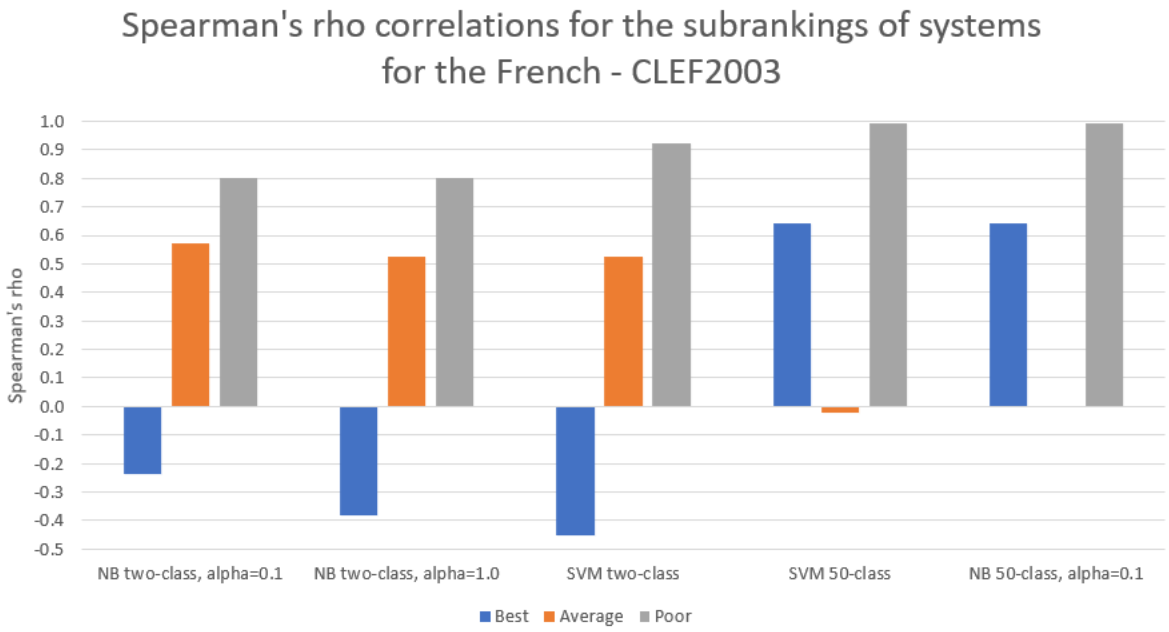


Figure 6. 2: Spearman correlations between the three subsets of systems: best, average and poor for the French test collection.

		Best	Average	Poor
NB two-class, alpha=0.1	**ρ**	-0.2381	0.5714	0.8024
	τ	-0.2143	0.4286	0.691
NB two-class, alpha=1.0	**ρ**	-0.4524	0.5238	0.8024
	τ	-0.3571	0.3571	0.691
SVM two-class	**ρ**	-0.3809	0.5238	0.9222
	τ	-0.2857	0.3571	0.8365
SVM 50-class	**ρ**	0.6428	-0.0238	0.994
	τ	0.4286	0.0714	0.982
NB 50-class, alpha=0.1	**ρ**	0.6428	0	0.994
	τ	0.4286	0.1429	0.982

Table 6. 24: Spearman correlations between the three subsets of systems: best, average and poor for the French test collection.

The p-value is less than 0.05 for the poor systems' correlations in the French collection, while for the best and average systems the p-value is greater than 0.05. The same observations can be seen for the Finnish test collection regarding the correlations for the best and average systems. The

correlations for the poor systems are also high, as was the case for the French collection, with a perfect agreement with the CLEF rankings when using the NB and the 50-class approach as shown in Figure 6.3 below. The detailed Kendall and Spearman correlations values are listed in table 6.25 below the figure.

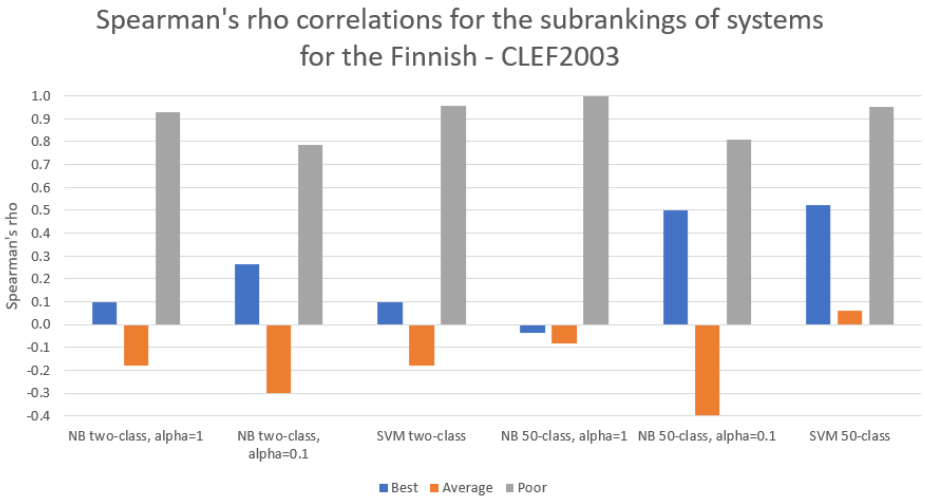


Figure 6. 3: Spearman correlations between the three subsets of systems: best, average and poor for the Finnish test collection.

		Best	Average	Poor
NB two-class, alpha=1	**ρ**	0.0952	-0.1807	0.9286
	τ	0.0714	0.037	0.8571
NB two-class, alpha=0.1	**ρ**	0.2619	-0.3012	0.7857
	τ	0.2143	-0.1111	0.6429
SVM two-class	**ρ**	0.0958	-0.1807	0.9581
	τ	0.1091	0.037	0.9092
NB 50-class, alpha=1	**ρ**	-0.0359	-0.0849	1
	τ	0.0364	0	1
NB 50-class, alpha=0.1	**ρ**	0.5	-0.3976	0.8095
	τ	0.3571	-0.3333	0.6429
SVM 50-class	**ρ**	0.5238	0.0602	0.9524
	τ	0.3571	0.1111	0.8571

Table 6. 25: Spearman correlations between the three subsets of systems: best, average and poor for the Finnish test collection.

The reason behind having positive correlations for the non-English test collections could be related to the number of retrieval models used which is less than the number of TREC systems. We only used 24 Terrier models, and some of them share similar behaviour since they are based on the Divergence From Randomness, while the TREC runs could be totally different and we had 76 runs for TREC-6 up to 129 for TREC-8. The size of the collection could also be a factor because the human pool formed for the French 2003 collection has 20358 documents and the Finnish one has 15605, while for TREC-6 it has 72270, for TREC-7 80345, and for TREC-8 the number of relevance assessments is 86830. When the number of documents in a test collection increases, maintaining the same pool depth could result in a shallow pool and many documents would be left out. Figure 6.2 shows the Spearman values for the best, average and poor systems for the French test collections.

6.6.4 Evaluation using bpref-10 and infAP

Because we were dealing with a set of documents with an incomplete set of judgments, we used other metrics that work better in such environments, namely infAP and bpref-10. We show the results in the tables below. Table 6.26 compares the different Kendall's tau and Spearman's rho values obtained when the Terrier models were ranked using the bpref-10 measure.

Correlations for the French test collection based on bpref-10								
Number of relev. docs. used	**Actual Qrels**		**NB two-class, alpha=1**		**NB two-class, alpha=0.1**		**SVM two-class**	
	T	ρ	τ	ρ	τ	ρ	τ	ρ
1	0.6182	0.7964	0.8146	0.9356	0.7495	0.9029	0.7905	0.9254
2	0.3752	0.5325	0.5428	0.6931	0.8509	0.9517	0.6764	0.8404
3	0.6909	0.8599	0.6255	0.7956	0.7127	0.8747	0.6350	0.7879
4	0.7541	0.8978	0.7468	0.8915	0.7855	0.9047	0.7491	0.8943
5	0.7855	0.9039	0.7636	0.8982	0.7055	0.8730	0.7418	0.8921
6	0.8073	0.9156	0.7418	0.8869	0.7491	0.8886	0.7200	0.8726
7	0.8146	0.9182	0.7782	0.9017	0.7927	0.9008	0.7905	0.9039
8	0.7927	0.9095	0.8073	0.9187	0.8124	0.9080	0.8000	0.9169
9	0.8000	0.9113	0.7709	0.8969	0.7636	0.8939	0.8000	0.9169
10	0.8727	0.9530	0.8073	0.9256	0.7832	0.9039	0.8218	0.9304
11	0.8509	0.9382	0.8073	0.9256	0.8000	0.9082	0.8124	0.9267
12	0.8488	0.9323	0.8218	0.9274	0.8073	0.9161	0.8270	0.9293
13	0.8509	0.9317	0.8146	0.9200	0.8146	0.9200	0.8364	0.9334
14	0.8655	0.9395	0.8291	0.9326	0.8509	0.9400	0.8509	0.9435
15	0.8509	0.9343	0.8342	0.9330	0.8364	0.9313	0.8509	0.9374
16	0.8291	0.9287	0.8582	0.9487	0.8509	0.9400	0.8582	0.9491
17	0.8436	0.9313	0.8727	0.9552	0.8582	0.9435	0.8727	0.9552

Table 6. 26: Kendall's tau (τ) and Spearman's rho (ρ) correlations based on bpref-10 scores for the French test collection

In table 6.27, the results with the infAP measure are more robust with fewer fluctuations and they do confirm the correlations obtained using the MAP scores.

Correlations for the French test collection based on infAP								
Number of relev. docs. used	**Actual Qrels**		**NB two-class, alpha=1**		**NB two-class, alpha=0.1**		**SVM two-class**	
	T	ρ	τ	ρ	τ	ρ	τ	ρ
1	0.7997	0.6425	0.6207	0.7684	0.6788	0.7910	0.6860	0.8354
2	0.9215	0.7877	0.8385	0.9398	0.6933	0.8293	0.6715	0.8076
3	0.9302	0.8094	0.7731	0.8954	0.8457	0.9372	0.7731	0.8911
4	0.9528	0.8457	0.8022	0.8937	0.8530	0.9432	0.8509	0.9426
5	0.9519	0.8530	0.8748	0.9537	0.8893	0.9667	0.9018	0.9652
6	0.9624	0.8675	0.9256	0.9772	0.8582	0.9448	0.9256	0.9763

7	0.9746	0.8966	0.9401	0.9841	0.8893	0.9650	0.9290	0.9772
8	0.9652	0.8800	0.9329	0.9841	0.9256	0.9789	0.9619	0.9920
9	0.9572	0.8675	0.9564	0.9896	0.9183	0.9780	0.9619	0.9920
10	0.9774	0.9018	0.9636	0.9922	0.9309	0.9848	0.9619	0.9920
11	0.9772	0.9111	0.9619	0.9937	0.9546	0.9911	0.9546	0.9911
12	0.9806	0.9183	0.9619	0.9937	0.9546	0.9911	0.9401	0.9867
13	0.9720	0.8893	0.9619	0.9937	0.9329	0.9841	0.9401	0.9867
14	0.9606	0.8748	0.9546	0.9928	0.9546	0.9893	0.9256	0.9833
15	0.9606	0.8748	0.9256	0.9850	0.9546	0.9911	0.9329	0.9859
16	0.9572	0.8603	0.9256	0.9859	0.9183	0.9780	0.9018	0.9682
17	0.9580	0.8675	0.9474	0.9893	0.9164	0.9761	0.9177	0.9752

Table 6. 27: Kendall's tau (τ) and Spearman's rho (ρ) correlations based on infAP scores for the French test collection

Similar behaviour is seen for the Finnish test collection, where the qrels generated using the classifiers provide better correlations with only a few relevant documents selected for training than the subsets of actual qrels. Although the correlations resulting from using the bpref-10 measure do not exceed the reliability threshold of a Kendall's tau of 0.8, the qrels seem to provide better correlations with the infAP measure and they confirm the values obtained when using the MAP scores. Because the infAP measure was found to be more robust than the bpref-10 measure in the literature, we can say that the pseudo-qrels generated when using only 3 relevant documents and more are reliable as we can see in table 6.28.

Correlations for the Finnish test collection based on bpref-10								
Number of relev. docs. used	**Actual Qrels**		**NB two-class, alpha=1**		**NB two-class, alpha=0.1**		**SVM two-class**	
	T	ρ	τ	ρ	τ	ρ	τ	ρ
1	0.2160	0.2708	0.3656	0.4746	0.2633	0.4039	0.3803	0.4997
2	0.3876	0.5430	0.4562	0.5805	0.5876	0.7480	0.4854	0.6079

3	0.4388	0.5667	0.6095	0.7154	0.5867	0.7183	0.6898	0.7798
4	0.6069	0.7530	0.5876	0.7154	0.7117	0.8699	0.5850	0.6788
5	0.6606	0.8185	0.4899	0.6185	0.5511	0.7332	0.5219	0.6327
6	0.7190	0.8629	0.5730	0.7346	0.6679	0.7950	0.6874	0.8348
7	0.7336	0.8564	0.7263	0.8755	0.7044	0.8612	0.7532	0.8990
8	0.7336	0.8616	0.7117	0.8621	0.7336	0.8925	0.7166	0.8614
9	0.7971	0.9045	0.7774	0.8834	0.7774	0.9012	0.7774	0.8943
10	0.7898	0.9084	0.7532	0.8753	0.7628	0.8964	0.7628	0.8812

Table 6. 28: Kendall's tau (τ) and Spearman's rho (ρ) correlations based on bpref-10 scores for the Finnish test collection

Number of relev. docs. used	**Actual Qrels**		**NB two-class, alpha=1**		**NB two-class, alpha=0.1**		**SVM two-class**	
	T	ρ	τ	ρ	τ	ρ	τ	ρ
1	0.6727	0.8417	0.6364	0.7903	0.3673	0.5128	0.6364	0.7903
2	0.6873	0.8547	0.7236	0.8730	0.5636	0.7234	0.7359	0.8706
3	0.7432	0.8884	0.8182	0.9374	0.7236	0.8712	0.8233	0.9371
4	0.7309	0.8834	0.7409	0.9079	0.8764	0.9600	0.8400	0.9356
5	0.8327	0.9487	0.7964	0.9226	0.8473	0.9574	0.8869	0.9593
6	0.8255	0.9435	0.8836	0.9652	0.8816	0.9641	0.8889	0.9658
7	0.7964	0.9278	0.9055	0.9739	0.9055	0.9739	0.9055	0.9739
8	0.8087	0.9293	0.8982	0.9730	0.9055	0.9739	0.8982	0.9730
9	0.8358	0.9382	0.9127	0.9774	0.9055	0.9748	0.9055	0.9739
10	0.8452	0.9519	0.9035	0.9719	0.9055	0.9748	0.9055	0.9739

Table 6. 29: Kendall's tau (τ) and Spearman's rho (ρ) correlations based on infAP scores for the Finnish test collection

Tables 6.30 and 6.31 below show the Spearman correlations between the different retrieval models' subrankings after dividing them into three categories: best, average and poor. The AQML pseudo qrels provided high correlations when using different numbers of relevant documents. In most cases and for both the French and Finnish test collections, the pseudo-qrels produced using the AQML technique showed a greater power of discrimination with a Spearman value that reached 1 for the poor systems

and in some cases for the average systems. However, for the Finnish test collection, unusual fluctuations in the correlations were noticed. This could be related to the fact that the Finnish language is a complex language and there were not sufficient relevant documents for all the topics to better train the classifiers.

Correlations of the Best systems, French Test Collection								
Number of relev. docs. Used	**Actual Qrels**		**SVM two-class**		**NB two-class, alpha=1**		**NB two-class, alpha=0.1**	
	P	τ	ρ	τ	ρ	τ	ρ	T
1	0.8810	0.7143	0.8810	0.7857	0.9048	0.7857	0.7381	0.6429
2	0.8571	0.7857	0.9286	0.8571	0.7381	0.6429	0.9286	0.8571
3	0.8810	0.7857	0.7857	0.6429	0.8095	0.7143	0.9048	0.7857
4	0.8095	0.7143	0.9102	0.8365	0.8333	0.7143	0.8333	0.7143
5	0.9286	0.8571	0.9762	0.9286	0.9286	0.8571	0.9048	0.7857
6	0.9048	0.7857	0.9048	0.7857	0.9286	0.8571	0.7143	0.5714
7	0.9286	0.8571	0.9222	0.8365	0.9048	0.7857	0.8333	0.7143
8	0.9048	0.7857	0.9286	0.8571	0.9286	0.8571	0.9048	0.7857
9	0.9286	0.8571	0.9286	0.8571	0.9286	0.8571	0.9048	0.7857
10	0.9762	0.9286	0.9286	0.8571	0.9286	0.8571	0.9048	0.7857
11	0.9762	0.9286	0.9048	0.7857	0.9286	0.8571	0.9286	0.8571
12	0.9762	0.9286	0.9048	0.7857	0.9286	0.8571	0.9286	0.8571
13	0.9762	0.9286	0.9048	0.7857	0.9286	0.8571	0.9048	0.7857
14	0.9762	0.9286	0.9048	0.7857	0.9286	0.8571	0.9286	0.8571
15	0.9762	0.9286	0.9048	0.7857	0.9286	0.8571	0.9286	0.8571
16	0.9762	0.9286	0.8333	0.7143	0.9048	0.7857	0.9048	0.7857
17	0.9762	0.9286	0.9222	0.8365	0.9286	0.8571	0.8743	0.7638

(a)

Correlations of the Average systems, French Test Collection								
Number of relev. docs. Used	**Actual Qrels**		**SVM two-class**		**NB two-class, alpha=1**		**NB two-class, alpha=0.1**	
	P	τ	ρ	τ	ρ	τ	ρ	T
1	0.6786	0.5238	0.7857	0.7143	0.6429	0.6190	0.8214	0.7143
2	0.8214	0.7143	0.8571	0.7143	0.8929	0.8095	0.8571	0.7143
3	0.8571	0.7143	0.8929	0.8095	0.8929	0.8095	0.8929	0.8095

4	0.8929	0.8095	0.9643	0.9048	0.8929	0.8095	0.9643	0.9048
5	0.8571	0.7143	0.9643	0.9048	0.9643	0.9048	1.0000	1.0000
6	0.8929	0.8095	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
10	0.9643	0.9048	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
11	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
12	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
13	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
14	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
15	1.0000	1.0000	1.0000	1.0000	0.9643	0.9048	1.0000	1.0000
16	1.0000	1.0000	1.0000	1.0000	0.9643	0.9048	1.0000	1.0000
17	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

(b)

Correlations of the Poor systems, French Test Collection								
Number of relev. docs. Used	**Actual Qrels**		**SVM two-class**		**NB two-class, alpha=1**		**NB two-class, alpha=0.1**	
	P	τ	ρ	τ	ρ	τ	ρ	T
1	0.9910	0.9759	0.9910	0.9759	0.9550	0.8783	0.7748	0.6831
2	0.9910	0.9759	0.7748	0.6831	0.9910	0.9759	0.7748	0.6831
3	0.9910	0.9759	0.9550	0.8783	0.9550	0.8783	0.9550	0.8783
4	0.9910	0.9759	0.9550	0.8783	0.9550	0.8783	0.9550	0.8783
5	0.9910	0.9759	0.9550	0.8783	0.9550	0.8783	0.9550	0.8783
6	0.9910	0.9759	0.9550	0.8783	0.9550	0.8783	0.9550	0.8783
7	0.9910	0.9759	0.9550	0.8783	0.9910	0.9759	0.9550	0.8783
8	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759	0.9550	0.8783
9	0.9910	0.9759	0.9910	0.9759	1.0000	1.0000	0.9550	0.8783
10	0.9910	0.9759	0.9910	0.9759	1.0000	1.0000	0.9910	0.9759
11	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759
12	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759
13	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759
14	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759
15	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759	0.9910	0.9759
16	0.9910	0.9759	0.9818	0.9500	0.9910	0.9759	0.9910	0.9759
17	0.9910	0.9759	0.9818	0.9500	0.9910	0.9759	0.9910	0.9759

(c)

Table 6. 30: French retrieval model subrankings Spearman correlations: (a) best systems, (b) average systems and (c) poor systems

Correlations of the Best systems, Finnish Test Collection								
Number of relev. docs. Used	Actual Qrels		SVM two-class		NB two-class, alpha=1		NB two-class, alpha=0.1	
	P	τ	ρ	τ	ρ	τ	ρ	T
1	0.2857	0.2143	0.3571	0.3571	0.3571	0.3571	-0.1190	-0.1429
2	0.4286	0.3571	0.3333	0.2857	0.3095	0.2143	0.0714	0.0714
3	0.5000	0.3571	0.7143	0.5714	0.7619	0.6429	0.1429	0.1429
4	0.5476	0.4286	0.7381	0.6429	0.6667	0.5000	0.8810	0.7857
5	0.8333	0.7143	0.8024	0.6910	0.5952	0.4286	0.8095	0.6429
6	0.7857	0.6429	0.8810	0.7857	0.8810	0.7857	0.8571	0.7143
7	0.6905	0.5714	0.9286	0.8571	0.9286	0.8571	0.9286	0.8571
8	0.6905	0.5714	0.9286	0.8571	0.9286	0.8571	0.9286	0.8571
9	0.7785	0.6910	0.9286	0.8571	0.9286	0.8571	0.9286	0.8571
10	0.8333	0.7143	0.9286	0.8571	0.8982	0.8365	0.9286	0.8571

(a)

Correlations of the Average systems, Finnish Test Collection								
Number of relev. docs. Used	Actual Qrels		SVM two-class		NB two-class, alpha=1		NB two-class, alpha=0.1	
	P	τ	ρ	τ	ρ	τ	ρ	T
1	0.7857	0.6190	0.7857	0.6190	0.7857	0.6190	0.3214	0.3333
2	0.9643	0.9048	0.8571	0.7143	0.8571	0.7143	0.7500	0.6190
3	0.9643	0.9048	0.9286	0.8095	0.9286	0.8095	0.9286	0.8095
4	0.9286	0.8095	1.0000	1.0000	0.9550	0.8783	1.0000	1.0000
5	0.9286	0.8095	1.0000	1.0000	0.9643	0.9048	0.9643	0.9048
6	0.9286	0.8095	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
7	0.9643	0.9048	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	1.0000	1.0000	0.9643	0.9048	0.9643	0.9048	1.0000	1.0000
9	1.0000	1.0000	0.9643	0.9048	0.9643	0.9048	0.9643	0.9048
10	0.9643	0.9048	0.9643	0.9048	0.9643	0.9048	0.9643	0.9048

(b)

Correlations of the Poor systems, Finnish Test Collection								
Number of relev. docs. Used	Actual Qrels		SVM two-class		NB two-class, alpha=1		NB two-class, alpha=0.1	
	P	τ	ρ	τ	ρ	τ	ρ	T
1	1.0000	1.0000	0.9643	0.9048	0.9643	0.9048	0.9286	0.8095
2	0.8929	0.8095	1.0000	1.0000	1.0000	1.0000	0.9286	0.8095
3	0.8829	0.7807	0.9643	0.9048	0.9643	0.9048	0.9286	0.8095

4	0.8929	0.8095	1.0000	1.0000	0.8571	0.7143	1.0000	1.0000
5	0.9643	0.9048	1.0000	1.0000	0.9286	0.8095	0.9643	0.9048
6	0.9643	0.9048	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
7	0.8929	0.8095	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	0.8929	0.8095	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9	0.9370	0.8783	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

(c)

Table 6. 31: Finnish retrieval models subrankings Spearman correlations: (a) best systems, (b) average systems and (c) poor systems

With only a few relevant documents selected for training, the AQML technique we proposed in section 6.2 using the NB and SVM classifiers seems to provide better correlations than using the subset of actual qrels alone. With 30 relevant documents for TREC-6, 60 relevant documents for TREC-7 and 40 relevant documents for TREC-8, the AQML SVM pseudo-qrels achieved higher tau correlations than the ones using the NB classifier and the subsets of actual qrels.

6.7 Conclusion

Ranking retrieval systems with automatically generated qrels has for a long time been an active research field in information retrieval, since test collections are the standard framework for evaluation, and building the relevance assessments remains the most expensive and time-consuming task. Several approaches have succeeded in producing qrels which allow ranking the retrieval systems with a positive correlation with the rankings obtained from using the human-built qrels. However, all these techniques

suffer from their inability to discriminate between the best systems. We described in chapter 5 a new technique called the ML technique which classifies the retrieved documents for a topic using the supervised machine learning NB and SVM classifiers by following either a two-class classification approach or the 50-class approach. We showed that this method was able to outperform previous approaches by providing higher correlations in the overall system rankings between the ML rankings and the TREC rankings. Since it failed to discriminate between the best systems, we investigated in this chapter the impact of using actual qrels to train the classifiers. We called this new method the AQML technique. The aim of the AQML approach was to increase the overall system rankings' correlations and to give more power of discrimination between the best, average and poor systems while using a minimal number of actual qrels to be able to reduce the cost of building the relevance assessments. Since the qrels in the NIST pool are ordered alphabetically, we needed a more efficient technique to retrieve relevant documents in a fast way. That is why we used the Max Mean, Non-Stationary pooling technique described by Losada et al. (2016) and which was based on the multi-armed bandits' probability concept. Having used the documents retrieved by Losada's technique, we formed subsets of actual qrels of different sizes. We ranked the TREC systems using each of these subsets and we measured the correlations between each of the rankings obtained with the TREC rankings. The correlations exceeded the 0.8

threshold for the Kendall's tau and 0.9 for the Spearman's rho in most cases. Next, we wanted to explore if it was possible to apply our AQML technique to expand a minimal set of qrels by selecting a few relevant documents for each topic and using them as a training set for the ML technique and then apply the two-class classification approach to find more relevant documents, and thus build the pseudo-qrels and improve the overall and sub rankings obtained with the subsets of actual qrels. We found for TREC-6 and TREC-7 that the correlations obtained when using the expanded qrels did not actually improve, however they maintained the same high correlations we got from using the actual qrels solely. A small improvement was noticed on TREC-8. In all the cases, and for the different number of relevant documents tested, the difference in the correlations between the rankings from the subsets of actual qrels and the TREC rankings and from the expanded qrels using the AQML technique and the TREC rankings is not very high. Using the AQML expanded qrels, we were able to discriminate between the best systems. Not only did we evaluate the systems rankings using the MAP scores, but we also computed the bpref and infAP metrics which are recommended in the case of incomplete sets of judgments. The results we found using the bpref and infAP measures confirmed our findings with the MAP scores. The number of relevant documents to select as a training set differed from one test collection to another, but it was possible to set a minimum boundary for the number.

Whenever we used more 25 relevant documents on average, we were able to achieve reliable pseudo-qrels which led to a Kendall's tau value > 0.8. The automatically expanded qrels did not provide better correlations than the subsets of actual qrels for all test collections, and this could be related to the number of hard topics found in each test collection.

After showing that AQML worked on TREC English test collections, we reproduced the experiments for both the ML and AQML methods for non-English languages, namely the French and Finnish test collections found in the CLEF European 2003 monolingual sets. We chose the CLEF datasets because they are TREC-like test collections, in that they consist of a set of documents, topics and human relevance. The ML automatic technique for producing pseudo-qrels provided high and positive correlations between the overall systems rankings generated based on the ML qrels and the ones based on the CLEF qrels. We did not have any access to the original CLEF system runs, so we used the Terrier retrieval models as surrogates. Similar to how the ML performed on TREC, it lacked the power of discrimination between the best and average systems, but it could discriminate between the poor ones. When applying the AQML technique, we faced a problem with some of the topics in the CLEF2003 since they did not have any relevant documents available in the set of actual qrels, therefore we excluded them from the experiments. The correlations between the overall rankings obtained from using the AQML qrels and the CLEF rankings were very high,

and we were able to discriminate between the best systems. This can confirm that hard topics have a huge impact on discriminating between the best systems because with the CLEF test collections, they produced qrels using the AQML better than the actual qrels. The rankings of the system were computed initially based on the MAP scores then re-evaluated using the bpref-10 and the infAP. The results obtained confirmed those observed using the MAP measure.

Chapter 7 – Conclusions

Since information retrieval has become a major part of our daily activities with the use of web search engines to satisfy a particular information need, it is important to evaluate the performance of a new retrieval system by measuring its recall and precision and to compare it with other ones by ranking them according to a performance measure. The traditional framework to conduct such an evaluation is through the use of test collections which consist of a set of documents, a set of topics and a set of relevance assessments. The cost of building a test collection is always a concern, since the task of producing the relevance judgments is time, effort and money consuming as it relies on human judges to examine the documents retrieved for a range of topics and identify their relevance. All the attempts made to reduce this cost approached the problem as a ranking problem as evaluating the systems based on the recall is not possible in the absence of real judgments. The baseline for comparing the rankings obtained using the automatically produced pseudo-qrels is the rankings produced by the human assessments. There is a continuous effort to devise new techniques that could improve further the correlations between the system rankings and that could produce pseudo-qrels that discriminate well between the best performing systems. The work done in this thesis aimed to devise methods that could outperform previous fully automatic techniques

used to rank retrieval systems in the absence of judgments and could discriminate between the best ones.

7.1 Review of Research Questions

With the work presented in the thesis, we showed that it was possible to form a set of qrels for a test collection fully automatically and in each of the chapters we attempted to answer the research questions we asked in chapter 1.

In chapter 4, we answered research question Q1:

Q1. Can we use keyphrases describing a topic as queries to retrieve more qrels?

which focused on using the keyphrases (KP) describing a topic as queries to retrieve more relevant documents and then considering the top k retrieved documents from all these queries as pseudo-qrels. Because we were approaching the problem as a ranking problem, the correlations that we obtained between the TREC rankings and the KP rankings were significant.

We were able to answer the third research question Q2, by making use of supervised machine learning to produce the qrels automatically in chapter 5.

Q2. Is it possible to use machine learning techniques to expand an initial set of presumed relevant documents and produce more qrels?

The correlations we obtained from using this technique that started with the NN technique and led to the ML technique with two approaches, the two-class and 50-class classifications outperformed the results from all the previous fully automatic methods in the literature. Because the recall measure is very important for the evaluation of an information retrieval system, we used these automatically produced qrels to discriminate between the best systems. However the correlations we obtained were negative, indicating that our rankings are almost the opposite of the gold standard ranking produced by using the human-built qrels. The contribution of chapter 6 was achieved by using a few known or actual qrels as training sets for the machine learning classifiers, thus answering research question Q3.

Q3. If it is not possible to form the qrels fully automatically, how many human-judged qrels should be supplied to start the process?

To answer this question, we tried different numbers of known relevant documents for each topic and used them to train the classifiers, so we can better classify the remainder of the documents. Both the overall system rankings and the best performing system rankings improved significantly when using actual qrels. Since we were conducting evaluation with incomplete judgments, we evaluated our results using the bpref and infAP measures which are usually recommended when there is no complete set of

assessments, unlike the MAP measure. This last evaluation was to answer research question Q4.

Q4. Do bpref and infAP give more accurate system rankings than MAP when we have an incomplete set of judgments?

The findings obtained when using the infAP and bpref measures confirmed those we obtained when we evaluated with the MAP. In most of the cases, the bpref showed many fluctuations in the results, while the infAP conformed well to the MAP values.

The last research question we addressed was Q5:

Q5. How well do the techniques developed in this thesis work for languages other than English such as French and Finnish?

This was answered in chapter 6 by applying both proposed techniques, the fully automatic (ML technique) and the one using actual qrels (AQML technique) to non-English test collections namely the CLEF2003 French and Finnish, both techniques behave similarly to how they did on TREC English test collections.

7.2 Overall conclusions

The conclusions that we can draw from each experimental chapter are as listed below:

Chapter 4:

- The KP method was not robust enough since there were several parameters that needed tuning and their values changed from one test collection to another.
- We could not find a way to standardize these values. This technique was similar to query expansion and the results we obtained were not satisfying to us.
- Neither using a manually annotated dataset for training, nor using a semantic similarity approach to compare between the keyphrases and the topics helped improving the correlation scores as we wished, instead when using a training set formed automatically, the correlations obtained were better than the ones resulting from a manually annotated dataset.
- KEA, according to the authors, works well with controlled vocabulary. We think since we are using a free vocabulary dataset, in no specific domain, this could have affected our results.

These findings motivated us to investigate other techniques which could be more robust and that could generate better sets of qrels.

Chapter 5:

- The nearest neighbour technique which selected a set of documents presumed relevant because it was retrieved by more than S% of the TREC runs was able to produce pseudo-qrels by expanding this initial set using a distance measure. The TREC runs rankings produced based

on NN qrels were better than the rankings obtained by Rajagopal et al.'s (2014) qrels from which this technique was inspired. The reason behind this improvement is due to the fact that the documents selected for training with a high cutoff percentage are more likely to be relevant than those used with a very low cutoff or number of occurrences. Even though we could measure the correlations between the rankings of the systems, it was not possible to measure the recall for each TREC run since the qrels are built automatically and thus we were not really sure of the relevance of each document.

- It was not possible to use the unsupervised machine learning K-Means algorithm to generate pseudo-qrels. In all our techniques, we rely on some known information about the topics or the documents to perform the classification and thus to produce the qrels, letting the K-Means choose randomly the centroids, and thus the clusters could not work well since we wanted exactly 50 clusters for the 50 TREC topics. The semi-supervised version did not provide acceptable MAP and correlation values, so we just have to conclude that unsupervised techniques cannot help to answer our research questions.
- The ML technique based on using supervised machine learning Naïve Bayes classifier and Support Vector Machines with the two-class and 50-class approaches was able to outperform the correlation results of all previous fully automatic techniques. This technique had a training

set formed automatically, so there was no human intervention in building the sets of qrels. These sets were able to rank the TREC runs with positive Kendall's tau and Spearman's rho correlations, and the values were better than any of the scores obtained by the automatic techniques in the literature. The statistical test showed however that the difference in the MAP scores was significant. The ad hoc test collections we used are relatively small when compared to the large Web Test collections. We expect our technique to work in a similar manner on large test collections.

- Measuring the recall for each TREC run is a limitation for the ML technique as well as it fails in discriminating among the best systems which usually have the best recall and are able to find relevant documents more than other systems even if these documents were not retrieved by the majority of these systems.

Chapter 6:

- Since the best performing systems are the ones with high recall, the discrimination between them could not happen based on automatic qrels because we don't know the true relevance of the documents, so we used a small number of actual or real qrels to train the ML classifiers to produce the AQML (Automatic Qrels for Machine Learning).

- AQML led to a significant improvement in the overall system rankings' correlation and it succeeded in discriminating between the best systems.
- Using the traditional NIST pooling technique, it was not possible to select which documents to use in the AQML, and thus we used a new pooling technique based on the multi-armed bandits problem devised by Losada et al. (2016) which retrieved relevant documents faster than the other pooling technique.
- The number of actual known relevant documents to use differed from one test collection to another. However we noticed that for all the test collections whenever we used more than 20 relevant documents for training, we started obtaining reliable Kendall's tau values (> 0.8). For TREC8, the qrels we generated were able to provide better correlations than the subsets of actual qrels.
- Both the ML and AQML technique worked with CLEF French and Finnish test collections and resulted in significant correlations between the overall and sub system rankings.

In both ML and AQML, we faced a challenge related to how hard the topic was. These topics had only a few relevant documents and in some of the CLEF topics, no relevant document was found at all for that topic. Thus, applying the automatic technique to these topics where the relevant documents are automatically selected would have affected the overall

rankings and the power of the technique to discriminate between the best systems.

The work done in this thesis has successfully resulted in forming pseudo-qrels which allow ranking different retrieval systems with high correlations with the gold standard rankings. If we wish to discriminate between the best systems, using actual qrels is needed, however with reduced human effort. The limitation of the work is still related to recall. We cannot claim that this technique can work with applications where high recall is needed or if we need to evaluate the performance of a single information retrieval system.

7.3 Future work

The techniques we proposed were tested using relatively small test collections. Thus, a future direction of the work could be to test how well these techniques will work on large-scale web test collections which consist of millions of documents such as the .GOV2 or ClueWeb test collections. The problem related to recall could become more difficult to solve as finding relevant documents would become harder from a pool of large documents. Using pools of depth 100 can become less efficient as well because these pools will be shallow compared to the amount of data retrieved and therefore it will be interesting to study different depth pools and how they would affect the performance of the machine learning based methods we devised. We could also study the variation of the pool depth and their effect

on the TREC test collections we used, the impact of the pool size on the number of relevant documents selected for training especially if we choose shallow pools (depth 10, 20 or 30) and then how the reliability of the qrels related to hard topics would become. It was not possible to test for larger pool depths (greater than 100) as we did not know the real relevance of the documents below rank 100 and that would require human assessors to judge each document and therefore increase the cost of the experiments. It would also be interesting to measure how well the Losada pooling technique would perform in retrieving relevant documents from such large collections, so that they can be used for training.

Other future work we suggest is related to graded relevance, since in web retrieval results we usually classify web pages as highly relevant, relevant or non-relevant (one could think of more grades to add according to the need). Three training sets at least would be required to train the NB classifier and SVM to classify documents and when using actual qrels to apply the AQML technique, differentiating between the highly-relevant, relevant and marginally relevant documents will probably be harder to achieve than simple binary relevance judgements. The highly relevant documents can be found much easily than the marginally relevant ones as they would contain clear relevant information about the topics. So, even average systems in most of the cases can find these highly relevant documents. But when it comes to the best systems, these can find the hard and marginally relevant

ones which can be hard for the automatic technique to find on their own if they do not contain explicit relevant documents. If it is just in the semantics of the documents, we could use word embedding used in deep learning to help detect such relevant documents. Such questions related to large web test collections and graded relevance remain open and could constitute an open field for future work.

References

Al-Maskari, A., Sanderson, M. and Clough, P. (2008). 'Relevance judgments between TREC and Non-TREC assessors'. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 683-684. ACM.

Alonso, O., 2013. 'Implementing crowdsourcing-based relevance experimentation: an industrial perspective'. Information retrieval, 16(2), pp.101-120.

Alonso, O., Rose, D. E. and Stewart, B. (2008) 'Crowdsourcing for relevance evaluation', ACM SIGIR Forum, 42(2), p. 9. doi: 10.1145/1480506.1480508.

Amati, G. and Van Rijsbergen, C. J. (2002) 'Probabilistic models of information retrieval based on measuring the divergence from randomness', ACM Transactions on Information Systems, 20(4), pp. 357–389. doi: 10.1145/582415.582416.

Aslam, J. A. and Savell, R. (2003) 'On the Effectiveness of Evaluating Retrieval Systems in the Absence of Relevance Judgments', in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: ACM (SIGIR '03), pp. 361–362. doi: 10.1145/860435.860501.

Baeza-Yates, R. and Ribeiro-Neto, B. (2011) 'Modern Information Retrieval, 2nd edition', Modern Information Retrieval, 2nd edition. Available at: http://portal.acm.org/citation.cfm?id=1796408&coll=DL&dl=GUIDE&CFID=329172738&CFTOKEN=24573745.

Bailey, P. et al. (2008) 'Relevance assessment: Are judges exchangeable and does it matter', Sigir '08, pp. 667–674. doi: 10.1145/1390334.1390447.

Bailey, P., Craswell, N. and Hawking, D. (2003) 'Engineering a Multi-purpose Test Collection for Web Retrieval Experiments', Inf. Process. Manage. Tarrytown, NY, USA: Pergamon Press, Inc., 39(6), pp. 853–871. doi: 10.1016/S0306-4573(02)00084-5.

Barry, C.L. (1994). 'User-defined relevance criteria: An exploratory study.' Journal of the American Society for Information Science, 45(3), pp.149-159.

Bernstein, Y. and Zobel, J. (2005). 'Redundant documents and search effectiveness'. In Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 736-743. ACM.

Braschler, M. (2003). 'CLEF 2003–Overview of results'. In Workshop of the Cross-Language Evaluation Forum for European Languages, pp. 44-63. Springer, Berlin, Heidelberg.

Buckley, C. and Voorhees, E. M. (2004) 'Retrieval evaluation with incomplete information', Proceedings of the 27th annual international conference on

Research and development in information retrieval - SIGIR '04, p. 25. doi: 10.1145/1008992.1009000.

Burgin, R. (1992) 'Variations in relevance judgments and the evaluation of retrieval performance', Information Processing & Management, 28(5), pp. 619–627. doi: https://doi.org/10.1016/0306-4573(92)90031-T.

Carterette, B. (2012) 'Incremental Test Collections Categories and Subject Descriptors', ACM Transactions on Information Systems, 30(1), pp. 680–687.

Carterette, B. and Soboroff, I. (2010) The effect of assessor error on IR system evaluation. doi: 10.1145/1835449.1835540.

Carterette, B., Allan, J. and Sitaraman, R. (2006) 'Minimal test collections for retrieval evaluation', Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '06, p. 268. doi: 10.1145/1148170.1148219.

Carterette, B., Bennett P., Chikering D. and Dumais S. (2008) 'Here or there', in European Conference on Information Retrieval, pp. 16–27.

Chandar, P., Webber, W. and Carterette, B. (2013) 'Document features predicting assessor disagreement', in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 745–748.

Chen, H. and Karger, D. R. (2006) 'Less is More: Probabilistic Models for Retrieving Fewer Relevant Documents', in Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: ACM (SIGIR '06), pp. 429–436. doi: 10.1145/1148170.1148245.

Chen, S. F. and Goodman, J. (1999) 'An empirical study of smoothing techniques for language modeling', Computer Speech & Language. Elsevier, 13(4), pp. 359–394.

Clarke, C. L. A., Cormack, G. V and Palmer, C. R. (1998) 'An Overview of Multitext', SIGIR Forum. New York, NY, USA: ACM, 32(2), pp. 14–15. doi: 10.1145/305110.305117.

Cleverdon, C. (1997) 'Readings in Information Retrieval', in Spärck Jones, K. and Willett, P. (eds). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 47–59. Available at: http://dl.acm.org/citation.cfm?id=275537.275544.

Clinchant, S. and Gaussier, E. (2009) Bridging Language Modeling and Divergence from Randomness Models: A Log-Logistic Model for IR. doi: 10.1007/978-3-642-04417-5_6.

Cormack, G. V and Mojdeh, M. (2009) 'Machine Learning for Information Retrieval: TREC 2009 Web, Relevance Feedback and Legal Tracks.', in TREC.

Cormack, G. V, Palmer, C. R. and Clarke, C. L. A. (1998) 'Efficient construction of large test collections', SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 282–289. doi: 10.1145/290941.291009.

Cortes, C. and Vapnik, V. (1995) 'Support-Vector Networks', Mach. Learn. Norwell, MA, USA: Kluwer Academic Publishers, 20(3), pp. 273–297. doi: 10.1023/A:1022627411411.

Cristianini, N. and Shawe-Taylor, J. (2000) An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. New York, NY, USA: Cambridge University Press.

Croft, B., Metzler, D. and Strohman, T. (2009) Search Engines: Information Retrieval in Practice. 1st edn. USA: Addison-Wesley Publishing Company.

Cronen-Townsend, S., Zhou, Y. and Croft, W. B. (2002) 'Predicting query performance', in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 299–306.

Cuadra, C. A. (1967) Experimental Studies of Relevance Judgments. Final Report [by Carlos A. Cuadra and Others]. System Development Corporation.

Cuadra, C. A. (1968) A Study of Relevance Judgments [microform] / Carlos A. Cuadra. Distributed by ERIC Clearinghouse [Washington, D.C.]. Available at: https://eric.ed.gov/?id=ED027921.

Cuadra, C. A. and Katter, R. V (1967) 'Opening the black box of "relevance"', journal of Documentation. MCB UP Ltd, 23(4), pp. 291–303.

Cuadra, C. A. and Katter, R. V (1967) 'The relevance of relevance assessment', in the American Documentation Institute, pp. 95–99.

Efron, M. (2009) 'Using multiple uery aspects to build test collections without human relevance udgments', Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5478 LNCS(m), pp. 301–312. doi: 10.1007/978-3-642-00958-7_28.

Fleiss, J. L. (1971) 'Measuring nominal scale agreement among many raters.', Psychological bulletin. American Psychological Association, 76(5), p. 378.

Gantz, J. and Reinsel, D. (2012) 'The Digital Universe in 2020: Big Data, Bigger Digi tal Shadows, and Biggest Growth in the Far East', International Data Corporation (IDC)., 2007(December 2012), pp. 1–16. doi: 10.1098/rspl.1860.0124.

Granmo, O.-C. (2008) 'A Bayesian Learning Automaton for Solving Two-Armed Bernoulli Bandit Problems', in International Journal of Intelligent Computing and Cybernetics. doi: 10.1108/17563781011049179.

Harman, D. (2010) 'Is the Cranfield Paradigm Outdated?', Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in

Information Retrieval, (January 2010), p. 1. doi: 10.1145/1835449.1835450.

Hawking, D., Craswell N., Bailey P. and Griffihs K. (2001) 'Measuring Search Engine Quality', Inf. Retr. Norwell, MA, USA: Kluwer Academic Publishers, 4(1), pp. 33–59. doi: 10.1023/A:1011468107287.

Hiemstra, D. (2009) 'Information retrieval models', Information Retrieval: searching in the 21st Century. Wiley London, pp. 2–19.

Hu, X., Bandhakavi, S. and Zhai, C. (2003) 'Error analysis of difficult TREC topics', in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 407–408.

Jayasinghe, G. K. et al. (2014) 'Extending test collection pools without manual runs', Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14, pp. 915–918. doi: 10.1145/2600428.2609473.

Joachims, T. (1998) 'Text Categorization with Support Vector Machines: Learning with Many Relevant Features', in Proceedings of the 10th European Conference on Machine Learning. Berlin, Heidelberg: Springer-Verlag (ECML'98), pp. 137–142. doi: 10.1007/BFb0026683.

Kazai, G. (2011) ‘In Search of Quality in Crowdsourcing for Search Engine Evaluation’, Advances in Information Retrieval, 6611, pp. 165–176. doi: 10.1007/978-3-642-20161-5.

Kekäläinen, J. (2005) ‘Binary and Graded Relevance in IR evaluations-Comparison of the Effects on Ranking of IR Systems’, Inf. Process. Manage. Elmsford, NY, USA: Pergamon Press, Inc., 41(5), pp. 1019–1033. doi: 10.1016/j.ipm.2005.01.004.

Kendall, M. G. (1945) ‘The treatment of ties in ranking problems’, Biometrika. JSTOR, pp. 239–251.

Kocabas, I. and Dincer, B. T. (2013) ‘A new statistical strategy for pooling : ELI A new statistical strategy for pooling : ELI ˙ lker Kocaba s’, (February 2019). doi: 10.1016/j.ipl.2013.07.007.

Koopman, B. and Zuccon, G. (2014) ‘Relevation!: An open source system for information retrieval relevance assessment’, in SIGIR 2014-Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1243–1244.

Krippendorff, K. (2004) Content Analysis: An Introduction to Its Methodology (second edition). Sage Publications.

Le, J. et al. (2010) ‘Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution’, in In SIGIR 2010 workshop, pp. 21–26.

Le, Q. and Mikolov, T. (2014) 'Distributed Representations of Sentences and Documents', in Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. JMLR.org (ICML'14), pp. II–1188--II–1196. Available at: http://dl.acm.org/citation.cfm?id=3044805.3045025.

Lease, M. and Yilmaz, E. (2012) 'Crowdsourcing for information retrieval', ACM SIGIR Forum, 45(2), p. 66. doi: 10.1145/2093346.2093356.

Lesk, M. (1995) 'The Seven Ages of Information Retrieval', Director, (5), pp. 12–14. doi: 10.1090/bull/1514.

Lesk, M. E. and Salton, G. (1968) 'Relevance Assessments and Retrieval System Evaluation', Information Storage and Retrieval, 4, pp. 343–359. doi: 10.1016/0020-0271(68)90029-6.

Lewandowski, D. and Sünkler, S. (2013) 'Designing search engine retrieval effectiveness tests with RAT', Information Services and Use, 33, pp. 53–59. doi: 10.3233/ISU-130691.

Losada, D. E., Parapar, J. and Barreiro, Á. (2016) 'Feeling Lucky?: Multi-armed Bandits for Ordering Judgements in Pooling-based Evaluation', Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp. 1027–1034. doi: 10.1145/2851613.2851692.

Lovins, J. B. (1968) 'Development of a stemming algorithm', Mech. Translat. & Comp. Linguistics, 11(1–2), pp. 22–31.

MacQueen, J. and others (1967) ‘Some methods for classification and analysis of multivariate observations’, in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp. 281–297.

Makary, M. et al. (2017) ‘Using Supervised Machine Learning to Automatically Build Relevance Judgments for a Test Collection’, in 28th International Workshop on Database and Expert Systems Applications, {DEXA} 2017 Workshops, Lyon, France, August 28-31, 2017. {IEEE} Computer Society, pp. 108–112. doi: 10.1109/DEXA.2017.38.

Makary, M., Oakes, M. P. and Yamout, F. (2016a) ‘Using key phrases as new queries in building relevance judgments automatically’, in Krestel, R., Mottin, D., and Müller, E. (eds) Proceedings of the Conference ‘Lernen, Wissen, Daten, Analysen’, Potsdam, Germany, September 12-14, 2016. CEUR-WS.org ({CEUR} Workshop Proceedings), pp. 175–176. Available at: http://ceur-ws.org/Vol-1670/paper-43.pdf.

Makary, M., Oakes, M. P. and Yamout, F. (2016b) ‘Towards automatic generation of relevance judgments for a test collection’, in Eleventh International Conference on Digital Information Management, {ICDIM} 2016, Porto, Portugal, September 19-21, 2016. IEEE, pp. 121–126. doi: 10.1109/ICDIM.2016.7829763.

Manning, C. D., Raghavan, P. and Schütze, H. (2008) Introduction to information retrieval. Cambridge University Press.

Mikolov, T. et al. (2013) 'Efficient Estimation of Word Representations in Vector Space', in 1st International Conference on Learning Representations, {ICLR} 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings. Available at: http://arxiv.org/abs/1301.3781.

Mizzaro, S. (1997) 'Relevance: The whole history', Journal of the American Society for Information Science, 48(9), pp. 810–832. doi: 10.1002/(SICI)1097-4571(199709)48:9<810::AID-ASI6>3.0.CO;2-U.

Mollá, D., Martinez, D. and Amini, I. (2013) 'Towards information retrieval evaluation with reduced and only positive judgements', Proceedings of the 18th Australasian Document Computing Symposium on - ADCS '13, pp. 109–112. doi: 10.1145/2537734.2537748.

Nuray, R. and Can, F. (2006) 'Automatic ranking of information retrieval systems using data fusion', Information Processing and Management, 42(3), pp. 595–614. doi: 10.1016/j.ipm.2005.03.023.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C. and Lioma, C. (2006). Terrier: A high performance and scalable information retrieval platform. In Proceedings of the OSIR Workshop, pp. 18-25.

Pavlu, V. et al. (2012) 'IR system evaluation using nugget-based test collections', Proceedings of the fifth ACM international conference on Web search and data mining - WSDM '12, p. 393. doi: 10.1145/2124295.2124343.

Porter, M. F. (1997) 'Readings in Information Retrieval', in Sparck Jones, K. and Willett, P. (eds). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 313–316. Available at: http://dl.acm.org/citation.cfm?id=275537.275705.

Rajagopal, P., Ravana, S. D. and Ismail, M. A. (2014) 'Relevance Judgments Exclusive of Human Assessors in Large Scale Information Retrieval Evaluation Experimentation', Malaysian Journal of Computer Science, 27(2), pp. 80–94. Available at: https://ejournal.um.edu.my/index.php/MJCS/article/view/6795.

Rajput, S. et al. (2012) 'Constructing test collections by inferring document relevance via extracted relevant information', Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12, p. 145. doi: 10.1145/2396761.2396783.

Robbins, H. (1952) 'Some Aspects of the Sequential Design of Experiments', Bulletin of the American Mathematical Society, 58, pp. 527-. doi: 10.1090/S0002-9904-1952-09620-8.

Sakai, T. (2016) 'Statistical Significance, Power, and Sample Sizes', pp. 5–14. doi: 10.1145/2911451.2911492.

Sakai, T. and Lin, C.-Y. (2010) 'Ranking Retrieval Systems without Relevance Assessments --- Revisited', EVIA 2010Proceedings of the 3rd

International Workshop on Evaluating Information Access (EVIA), pp. 25–33. doi: 10.1145/1008992.1009000.

Salton, G. (1968) Automatic Information Organization and Retrieval. McGraw Hill Text.

Salton, G. and McGill, M. J. (1986) Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, Inc.

Salton, G. and McGill, M. J. (1986) Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, Inc.

Salton, G., Wong, A. and Yang, C. S. (1975) ‘A Vector Space Model for Automatic Indexing’, Commun. ACM. New York, NY, USA: ACM, 18(11), pp. 613–620. doi: 10.1145/361219.361220.

Sanderson, M. (1998) ‘Accurate User Directed Summarization from Existing Tools’, in Proceedings of the Seventh International Conference on Information and Knowledge Management. New York, NY, USA: ACM (CIKM ’98), pp. 45–51. doi: 10.1145/288627.288640.

Sanderson, M. and Joho, H. (2004) ‘Forming test collections with no system pooling’, Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR ’04, p. 33. doi: 10.1145/1008992.1009001.

Sanderson, M., Scholer, F. and Turpin, a (2010) 'Relatively relevant: Assessor shift in document judgements', ADCS 2010 - Proceedings of the Fifteenth Australasian Document Computing Symposium, (December), pp. 60–67. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-84872873938&partnerID=40&md5=84ccc4f195e296b4e2ac5b1b9f80f746.

Saracevic, T. (1976) 'Relevance: A review of the literature and a framework for thinking on the notion in information science', in Eds.), Advances in Librarianship 6. Academic Press, pp. 79–138.

Saracevic, T. (1996). Relevance reconsidered. In Proceedings of the second conference on conceptions of library and information science (CoLIS 2) pp. 201-218. New York: ACM

Schamber, L. (1991) 'Users' Criteria for Evaluation in a Multimedia Environment', Proceedings of the ASIS Annual Meeting, 28.

Scholer, F., Turpin, A. and Sanderson, M. (2011) 'Quantifying Test Collection Quality Based on the Consistency of Relevance Judgements', Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval-SIGIR'11, pp. 1063–1072. doi: 10.1145/2009916.2010057.

Shapiro, S. S. and Wilk, M. B. (1965) 'An analysis of variance test for normality (complete samples)', Biometrika. JSTOR, 52(3/4), pp. 591–611.

Shi, Z., Li, P. and Wang, B. (2010) 'Using Clustering to Improve Retrieval Evaluation without Relevance Judgments', in Huang, C.-R. and Jurafsky, D. (eds) {COLING} 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China. Chinese Information Processing Society of China, pp. 1131–1139. Available at: http://aclweb.org/anthology/C/C10/C10-2130.pdf.

Smucker, M. D. and Jethani, C. P. (2012) 'Time to judge relevance as an indicator of assessor error', in Hersh, W. R. et al. (eds) The 35th International {ACM} {SIGIR} conference on research and development in Information Retrieval, {SIGIR} '12, Portland, OR, USA, August 12-16, 2012. ACM, pp. 1153–1154. doi: 10.1145/2348283.2348515.

Soboroff, I., Nicholas, C. and Cahan, P. (2001) 'Ranking retrieval systems without relevance judgments', Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01, pp. 66–73. doi: 10.1145/383952.383961.

Soboroff, I. and Robertson, S. (2003) 'Building a Filtering Test Collection for TREC 2002', in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. New York, NY, USA: ACM (SIGIR '03), pp. 243–250. doi: 10.1145/860435.860481.

Sormunen, E. (2002) 'Liberal relevance criteria of TREC - Counting on negligible documents?', Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02, p. 324. doi: 10.1145/564376.564433.

Spärck Jones, K. and Van Rijbergen, C. J. (1976) 'Information retrieval test collections', Journal of Documentation, 32, pp. 59–75. doi: 10.1108/eb026616.

Spearman, C. (1987) 'The proof and measurement of association between two things', The American journal of psychology. JSTOR, 100(3/4), pp. 441–471.

Spoerri, A. (2007) 'Using the structure of overlap between search results to rank retrieval systems without relevance judgments', Information Processing and Management, 43(4), pp. 1059–1070. doi: 10.1016/j.ipm.2006.09.009.

Turk, A. M. (2012) 'Amazon mechanical turk', Retrieved August, 17, p. 2012.

Vakkari, P. and Sormunen, E. (2004) 'The influence of relevance levels on the effectiveness of interactive information retrieval', JASIST, 55, pp. 963–969. doi: 10.1002/asi.20046.

Voorhees, E. M. (1998) 'Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness', in Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in

Information Retrieval. New York, NY, USA: ACM (SIGIR ’98), pp. 315–323. doi: 10.1145/290941.291017.

Voorhees, E. M. (2001) ‘Evaluation by Highly Relevant Documents’, in Croft, W. B. et al. (eds) {SIGIR} 2001: Proceedings of the 24th Annual International {ACM} {SIGIR} Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, {USA}. ACM, pp. 74–82. doi: 10.1145/383952.383963.

Voorhees, E. M. (2003) ‘Overview of the {TREC} 2003 Robust Retrieval Track’, in Voorhees, E. M. and Buckland, L. P. (eds) Proceedings of The Twelfth Text REtrieval Conference, {TREC} 2003, Gaithersburg, Maryland, USA, November 18-21, 2003. National Institute of Standards and Technology {(NIST)}, pp. 69–77. Available at: http://trec.nist.gov/pubs/trec12/papers/ROBUST.OVERVIEW.pdf.

Voorhees, E. M. and Harman, D. (1997) ‘Overview of the Sixth Text REtrieval Conference {(TREC-6)}’, in Voorhees, E. M. and Harman, D. K. (eds) Proceedings of The Sixth Text REtrieval Conference, {TREC} 1997, Gaithersburg, Maryland, USA, November 19-21, 1997. National Institute of Standards and Technology {(NIST)}, pp. 1–24. Available at: http://trec.nist.gov/pubs/trec6/papers/overview.ps.

Voorhees, E. M. and Harman, D. (1998) 'Overview of the Seventh Text REtrieval Conference TREC-7', in Proceedings of the Seventh Text REtrieval Conference (TREC-7), pp. 1–24.

Voorhees, E. M. and Harman, D. (1999) 'Overview of the Eighth Text REtrieval Conference {(TREC-8)}', in Voorhees, E. M. and Harman, D. K. (eds) Proceedings of The Eighth Text REtrieval Conference, {TREC} 1999, Gaithersburg, Maryland, USA, November 17-19, 1999. National Institute of Standards and Technology {(NIST)}. Available at: http://trec.nist.gov/pubs/trec8/papers/overview_8.ps.

Wan, X. and Xiao, J. (2008) 'Single Document Keyphrase Extraction Using Neighborhood Knowledge', in Fox, D. and Gomes, C. P. (eds) Proceedings of the Twenty-Third {AAAI} Conference on Artificial Intelligence, {AAAI} 2008, Chicago, Illinois, USA, July 13-17, 2008. {AAAI} Press, pp. 855–860. Available at: http://www.aaai.org/Library/AAAI/2008/aaai08-136.php.

Webber, W. et al. (2010) 'Assessor Error in Stratified Evaluation', in Proceedings of the 19th ACM International Conference on Information and Knowledge Management. New York, NY, USA: ACM (CIKM '10), pp. 539–548. doi: 10.1145/1871437.1871508.

Webber, W., Chandar, P. and Carterette, B. (2012) 'Alternative assessor disagreement and retrieval depth', in Chen, X. et al. (eds) 21st {ACM} International Conference on Information and Knowledge Management,

CIKM'12, Maui, HI, USA, October 29 - November 02, 2012. ACM, pp. 125–134. doi: 10.1145/2396761.2396781.

Wilcoxon, F. (1945) 'Individual Comparisons by Ranking Methods', Biometrics Bulletin. [International Biometric Society, Wiley], 1(6), pp. 80–83. doi: 10.2307/3001968.

Witten, I. et al. (1999) KEA: Practical Automatic Keyphrase Extraction, ACM DL. doi: 10.1145/313238.313437.

Wu, S. and Crestani, F. (2003) 'Methods for ranking information retrieval systems without relevance judgments', Proceedings of the 2003 ACM symposium on Applied computing - SAC '03, p. 811. doi: 10.1145/952532.952693.

Yilmaz, E. and Aslam, J. (2008) Estimating average precision when judgments are incomplete, Knowl. Inf. Syst. doi: 10.1007/s10115-007-0101-7.

Zhai, C. (2001) 'Notes on the Lemur TFIDF model'. http://www.cs.cmu.edu/~lemur/1.0/tfidf.ps

Zobel, J. (1998) 'How reliable are the results of large-scale information retrieval experiments?', Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '98, pp. 307–314. doi: 10.1145/290941.291014.