Université de Neuchâtel
Faculté des Sciences
Institut d'Informatique

Multilingual and Domain-Specific IR: A Case Study in Cultural Heritage

par

Mitra Akasereh

Thèse

Présentée à la Faculté des Sciences
Pour l'obtention du grade de Docteur ès Sciences

Jury de Thèse:

Prof. Jacques Savoy, directeur de thèse
Université de Neuchâtel, Suisse

Prof. Patrice Bellot, rapporteur
Université Aix-Marseille, France

Prof. Pascal Felber, rapporteur
Université de Neuchâtel, Suisse

Soutenue le 21 mai 2015

UNIVERSITÉ DE NEUCHÂTEL

Faculté des sciences
Secrétariat-décanat de Faculté
Rue Emile-Argand 11
2000 Neuchâtel - Suisse
Tél: + 41 (0)32 718 2100
E-mail: secretariat.sciences@unine.ch

IMPRIMATUR POUR THESE DE DOCTORAT

La Faculté des sciences de l'Université de Neuchâtel autorise l'impression de la présente thèse soutenue par

Madame Mitra AKASEREH

Titre:

"Multilingual and Domain Specific Information Retrieval: A Case Study in Cultural Heritage"

sur le rapport des membres du jury composé comme suit:

- Prof. Jacques Savoy, directeur de thèse, Université de Neuchâtel
- Prof. Pascal Felber, Université de Neuchâtel
- Prof. Patrice Bellot, Université Aix-Marseille, France

Neuchâtel, le 4 juin 2015

Le Doyen, Prof. B. Colbois

B. Colbois

“What is above all needed is to let the meaning choose the word, and not the other way around.”

George Orwell, Politics and the English Language

Abstract:

Nowadays we can find data collections in many different languages and in different fields. So we are facing with a rising need for search systems handling multilinguality as well as professional search systems which allow their users to search in a specific field of knowledge.

In this thesis we propose a search system for data on cultural heritage. Our data comes from different resources located in different countries and written in various languages. We study the specific structure, characteristics and terminology of data in this field in order to build an effective retrieval system. We evaluate different information retrieval models and indexing strategies on monolingual data to find the ones which are effective and compatible with the nature of our data the most. To deal with different languages we study each language separately and propose tools such as stemmers for each language and fusion operators to merge the results from different languages. To be able to cross the languages easily we study different translation methods. Moreover in order to enhance the search results we investigate different query expansion technics.

Based on our results we propose using models from DFR family for the English language and Okapi model for the French and Polish language along with a light stemmer. For crossing the language barrier we propose using a combination of translation methods. The Z-score operator is the best evaluated one when merging different results from different languages in our multilingual tests. Finally we propose applying query expansion using an external source to improve the search performance

Keywords: Domain-Specific IR, Cultural Heritage (CH), Query Expansion, Pseudo-Relevance Feedback, Data Fusion, Bilingual IR, Multilingual IR.

Acknowledgements

First of all, I wish to thank Prof. Jacques Savoy for providing me with his trust and his constant support and guidelines during the accomplishment of this thesis.

I would also like to express my thanks to Professors Pascal Felber and Patrice Bellot for being part of the jury.

My special thanks to:

My dearest Linda and Jean Saxod for their unconditional support and love.

My beloved sister Nikta who make me feel so blessed with her generous love and her constant support.

My very dear friend Antoine Baillieux for encouraging me to start my PhD and for his priceless friendship during my master and PhD studies.

My dear François Derouwaux for his moral support during writing this thesis.

Last but not least, I want to express my many thanks to my dear parents, my lovely sisters and my dearest nephew Kiavash for their endless love and support.

مامان و بابای عزیزم ، دوستتون دارم و بابت محبت‌ها و حمایت‌های همیشگی تون ازتون ممنونم.

Table of Contents

1 Introduction 1

1.1 Problem Statement 2

1.2 Motivation and Objectives 4

1.3 Organization of the Thesis 5

2 State of the Art 7

2.1 Information Retrieval 7

2.1.1 Challenges 8

2.2 Cross Language Information Retrieval (CLIR) 12

2.2.1 Challenges 14

2.3 Domain-Specific Information Retrieval 15

2.3.1 Challenges 15

3 Methodology (IR) 17

3.1 Indexing 17

3.1.1 Tokenization 18

3.1.2 Stopwords Elimination 18

3.1.3 Stemming and Lemmatization 19

3.2 Term Weight 21

3.3 Retrieval Models 22

3.3.1 Boolean Model 23

3.3.2 Vector-Space Model 24

3.3.3 Probabilistic Model 25

3.4 Evaluation 28

3.4.1 Efficiency 29

3.4.2 Effectiveness 29

3.4.3 Evaluation Measures 29
3.4.4 Relevance Assessment 32
3.5 Query Expansion and Relevance Feedback 33
3.5.1 Pseudo-Relevance Feedback 35
4 Methodology (CLIR) 37
4.1 Query Translation and Document Translation 37
4.2 Indirect Translation 38
4.3 No Translation 38
4.4 Translation Methods 39
4.4.1 Machine Readable Dictionaries 39
4.4.2 Statistical Approaches (Parallel and Comparable Corpora) 40
4.4.3 Machine Translation 40
4.4.4 Combination Approaches 41
4.5 Fusion 41
4.6 Query Expansion 42
5 Experiments 43
5.1 Introduction 43
5.2 Cultural Heritage 44
5.3 Challenges 45
5.4 Test-Collections 46
5.4.1 Monolingual Corpus 48
5.4.2 Multilingual Corpus 51
5.4.3 Relevance Assessment 53
5.5 Monolingual Retrieval 54
5.5.1 English and French 55
5.5.2 Polish Language 62
5.5.3 Conclusion 66

5.6 Bilingual Retrieval 67
5.6.1 Experiment Architecture 67
5.6.2 Results and Discussions 68
5.6.3 Conclusion 69
5.7 Multilingual Retrieval 69
5.7.1 Setup and Indexing 70
5.7.2 IR Models and Data Fusion 73
5.7.3 Results and Discussion 74
5.7.4 Conclusion 77
5.8 Query Expansion and Relevance Feedback 78
5.8.1 Related Work 78
5.8.2 Experiment Architecture 79
5.8.3 Query Expansion using Wikipedia 81
5.8.4 Pseudo-Relevance Feedback 84
5.8.5 Query-by-Query Analysis 85
5.8.6 Conclusion 86
6 Conclusion and Future Work 89
References 93

List of Figures

Figure 2.1 Basic structure of an IR system 8

Figure 2.2 Structure of a CLIR system using query translation 13

Figure 2.3 Structure of a CLIR system using document translation 14

Figure 3.1 Cosine of Θ is assumed as sim(d_j, q), $\Theta_1 < \Theta_2$ so d_1 will be ranked higher than d_2.....24

Figure 5.1 Sample of an English record (image of a Roman coin) 47

Figure 5.2 Sample of English, French and German topics.....49

Figure 5.3 Example of an enriched topic 82

List of Tables

Table 1.1 Ten most used languages on the Web as of 2010 3
Table 5.1 Documents in the monolingual corpus by language and media type.....48
Table 5.2 Statistics on the number of distinct indexing terms per topic 49
Table 5.3 Documents in the multilingual corpus by language and media type.....51
Table 5.4 Distinct indexing terms per topic 52
Table 5.5 Statistics on English, French & German corpora.....56
Table 5.6 Formulas used in different models for assigning indexing weight 57
Table 5.7 MAP of different IR models, English corpus.....58
Table 5.8 MAP of different IR models, French corpus.....58
Table 5.9 MAP of different combinations of IR models, English corpus.....60
Table 5.10 MAP of *idf*-based blind-query expansion, English and French queries....62
Table 5.11 Statistics on Polish corpus.....63
Table 5.12 Result MAP based on *n*-gram or trunc-*n* approaches 64
Table 5.13 Result MAP based on word-based indexing 65
Table 5.14 Result MAP based on Rocchio pseudo-relevance feedback 66
Table 5.15 MAP of different IR models, German topics on English corpus 69
Table 5.16 MAP of different IR models, French topics on English corpus 69
Table 5.17 Statistics for each language in the corpora.....71
Table 5.18 Size of stopword list for each language 72
Table 5.19 MAP for each language, with original or automatically translated queries, with or without a light stemmer 75
Table 5.20 Evaluation of different stemming and merging strategies 76
Table 5.21 Evaluation of different server selection approaches 76
Table 5.22 Results for different query formulations 83
Table 5.23 Results for PRF approach.....84

To my Father

تقدیم به بابا

Introduction 1

In our highly connected society the data growth rate follows an exponential curve. In April 2011 the U.S. Library of Congress has collected 235 Terabytes of data. More than 5 billion people are calling, texting, tweeting and browsing on mobile phones worldwide. In 2008, Google was processing 20,000 terabytes of data which makes it 20 petabytes a day. In the field of genomics world capacity is now 13 quadrillion DNA bases a year. But what is the use of data without transforming it into value? We need to analyze these data, discover patterns, and reveal new insights. On the other hand the actual Internet users expect that they will find all information they need on Internet. They use Internet to find answers to their questions, information on a specific issue or data to help them to make a decision. In 2013, the average number of searches per day on Google was almost 6 billion (in 1998, Google's official first year, only 9,800 searches performed per day)[1]. Therefore the information should not only be accessible but also easily consultable. These needs and the exponential growth in data volume result enormous challenges in analyzing, mining and retrieving data; challenges targeting the information retrieval systems. Information retrieval (IR) is concerned with representation, storage, organization and access to large collections of data (Baeza-Yates & Ribeiro-Neto, 1999). The main task of an information retrieval system is to find relevant information to a particular information need in data collections.

[1] http://wikibon.org/blog/big-data-statistics/
http://statisticbrain.com/

1.1 Problem Statement

Even though English language has dominated the information retrieval community since the late 1960s, a growing demand for tools capable of handling other languages is prompted in recent years. Data collections nowadays contain information in many different languages. Considering the Web as an example, its first decade (1990-2000) was marked by the omnipresence of the language of Shakespeare. At the end of 1996, 85% of 47 million Internet users spoke English. In December 2000, this proportion was 47% for about 407 million Internet users. While in December 2013 with a user population of 2,802 million the proportion of Internet users browsing in English was estimated at 28.6% against 23.2% for Chinese, 7.9% for Spanish, 4.8% for Arabic and 4.3% for Portuguese. Japanese language is in sixth position with 3.9% followed by 3.1% for Russian and 2.9% for German, 2.8% for French and finally 2.7% for Malay. Thus, during the period from 2000 to 2013 the number of online users per language has raised an average of 5,296 % for Arabic, 2,721% for Russian 1,910% for Chinese, 1,507% for Portuguese, 1,216% for Malay, 1,123% for Spanish and only 468% for English. Table 1.1 [2] shows the ten most used languages on the Web as of 2010. In this table only one language is assigned per person and people who speak more than one language are not taken into consideration. The Internet penetration rate in the table is the ratio between the number of users speaking a language and the total number of people who speak that language. On this basis, it is estimated that the use of other languages than English on the Web will reach levels comparable to or greater than those of English. Similarly, opportunities for growth in Spanish or Arabic are more important than the language of Molière. In this list of major languages, languages such as Urdu or Hindi should not be forgotten as the number of Internet users is growing rapidly in the Indian subcontinent (Peters et al., 2012; Nie & Savoy, 2008). Consequently the easy and efficient electronic access regardless the underlying language becomes an important issue. Users need to access data and search for information in any language. Furthermore they should be able to understand and reuse the sought data (Peters et al., 2012). That is why there is an essential need to

[2] http://internetworldstats.com

move from within-language information retrieval systems toward multilingual IR systems.

Table 1.1 Ten most used languages on the Web as of 2013

Top 10 Languages on Internet	Internet users by language	% of Internet penetration by language	% of growth in Internet 2000-2010	% of Internet users	Total population per language as of 2014
English	800,625,314	58.4 %	468.8 %	28.6 %	1,370,977,116
Chinese	649,375,491	46.6 %	1,910.3 %	23.2 %	1,392,320,407
Spanish	222,406,379	50.6 %	1,123.3 %	7.9 %	439,320,916
Arabic	135,610,819	36.9 %	5,296.6 %	4.8 %	367,465,766
Portuguese	121,779,703	46.7 %	1,507.4 %	4.3 %	260,874,775
Japanese	109,626,672	86.2 %	132.9 %	3.9 %	127,103,388
Russian	87,476,747	61.4 %	2,721.8 %	3.1 %	142,470,272
German	81,139,942	85.7 %	194.9 %	2.9 %	94,652,582
French	78,891,813	20.9 %	557.5 %	2.8 %	377,424,669
Malay	75,459,025	26.6 %	1,216.9 %	2.7 %	284,105,671
Top 10	2,362,391,905	48.5 %	696.1 %	84.3 %	4,856,715,562
Rest of Languages	440,087,029	19.0 %	585.2 %	15.7 %	2,325,143,057
World total	2,802,478,934	39.0 %	676.3 %	100.0 %	7,181,858,619

Another challenge in the field of IR systems originates from the fact that the available data in the digital universe comes in various flavors. We can find data in various fields: science, health, industry, commerce, entertainment, life, etc. A general user who seeks information on any of these fields can easily find some data relating

to her/his field of interest. But the challenge rises when a professional user search for precise information in a particular field of knowledge. In such a case we are facing with precise queries, specific document formats, particular topicalities and terminology. Consequently an IR system used for professional search should consider different strategies and assumptions according to each specific domain of search. Certainly in the growing rate of using search engines in a daily basis, professional users are not an exception. The reason which necessitates that we investigate in particular domain-specific information retrieval systems.

1.2 Motivation and Objectives

The need for an IR system that handles multilingual data along with the need for professional search systems in order to handle one specific domain of knowledge (as mentioned in Section 1.1) motivated us to investigate in the current research. Having this motivation, the availability of a multilingual test-collection containing data only from cultural heritage (CH) domain led us to start designing and conducting our experiments. The data in CH domain is normally characterized by:

- Short and ambiguous queries
- Containing names (geographical, person, work)
- Multilingiality
- Containing lots of images and pictures with a short textual description

Our first objective is to design a system that deals with this specific data (cultural heritage objects) and its particular characteristics. We want to investigate the impact of the structure and characteristics of the collection (e.g., short descriptions, sparse information, etc.) on the efficiency of our search. We also aim to find the best way to deal with the specific terminology of this field of knowledge in order to propose solutions for enriching the submitted queries. Accordingly we will investigate different query expansion techniques in our study. Our other objective is to identify the best techniques in retrieving data from multilingual data collections. To achieve this goal we first investigate different translation techniques. Moreover we investigate the possibility of merging the results from different systems in order to produce the

final results with a higher precision. In the multilingual context we also propose different stemming algorithms for different languages according to their corresponding morphology and grammar.

1.3 Organization of the Thesis

The outline of this thesis is as follows:

- In Chapter 2 we talk about classical information retrieval and the challenges we deal with in this field. Then we describe cross language and domain-specific information retrieval and their corresponding challenges.

- Chapter 3 is dedicated to the main methods that are used in different steps in the whole process of information retrieval.

- In a cross language information retrieval (CLIR) system for the retrieval process the same methodology is applied as in a classical IR system. But in order to deal with different languages we would need some othe methods. In Chapter 4 we explain these extra methods that are needed in a CLIR context.

- Chapter 5 is dedicated to our experiments. In this chapter we describe different experiments that we designed in order to fulfill our goals of the current research and we discuss the obtained results.

- Finally in chapter 6 we explain the conclusions we can deduce from our obtained results and propose the future works that can be conducted in order to enhance our results.

State of the Art 2

As in this work we are exploring multilingual and domain-specific IR systems it is important to have a clear definition for each of them. This chapter is dedicated to the definitions of three IR systems: classical information retrieval, cross language information retrieval and domain-specific information retrieval. We will give a brief description of each and discuss their related challenges.

2.1 Information Retrieval

Information retrieval (IR) is concerned with representation, storage, organization and access to large collections of data (Baeza-Yates & Ribeiro-Neto, 1999). The main job of an IR system is looking for specified information, sought by user, in a data collection and extracting those items that are relevant to the users' information need. The data within the collection can be of any type such as text, images, video, spoken documents, as well as any mix of them (Grossman & Frieder, 2004; Boughanem, 2008; Sanderson, 1996).

Figure 2.1 illustrates the main structure of an IR system. The system, with a given method, converts the documents within the data collection into an appropriate representation form. Now the system can easily search the data using this new format. This new representation is called an index. When a user needs some information s/he builds her/his request and gives it to the IR system. The IR system then indexes the user's query using the same methods as it used for the documents, and transforms it to the proper representation form. Now the query is understandable by the system and the retrieval process can start. The system then will compare the restructured query with the documents in the document index using certain retrieval strategies or matching methods. The retrieval strategy assigns to each document a relevance score,

sometimes called RSV (Retrieval Status Value). This score shows the similarity level between the given query and that document. Afterwards the system puts all the documents as well as their assigned scores in a list ordered by the document with the highest score on the top. This ranked list of documents, estimated as relevant, will be send back to the user as the result (Boughanem, 2008).

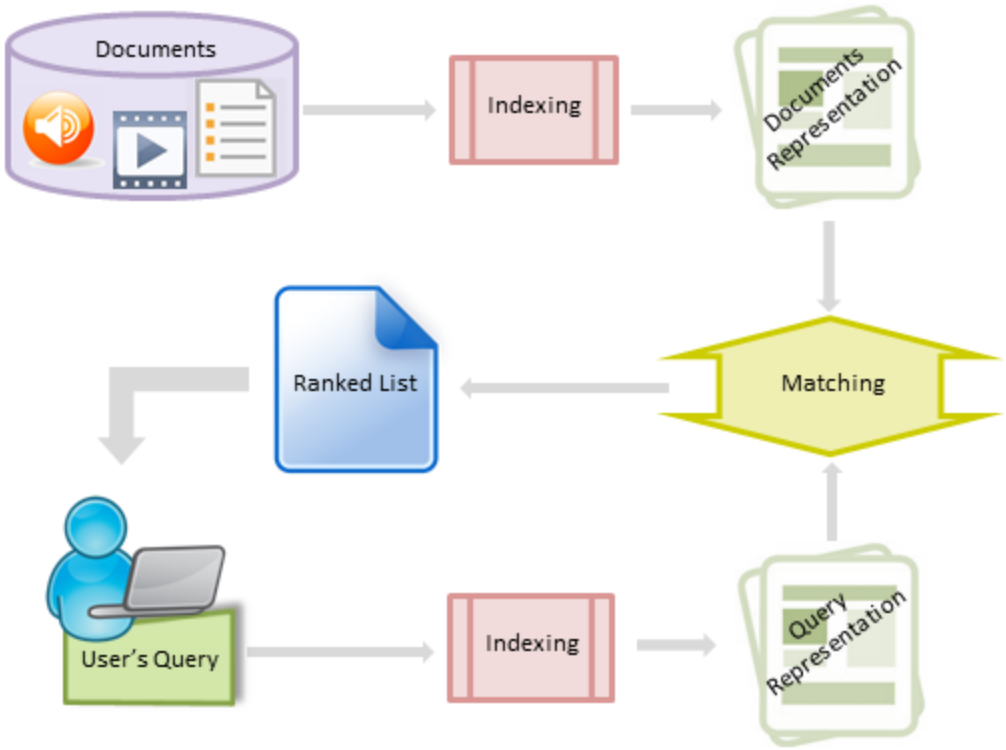

Figure 2.1 Basic structure of an IR system

2.1.1 Challenges

As mentioned before the first part in an IR process is to convert the data collection into an index. However processing the textual data in order to prepare them for indexing is not trivial. When dealing with natural languages each language brings its own difficulties according to its corresponding morphology and characteristics. Different natural languages have different linguistic constructions that influence the retrieval procedure. For example in some languages like Chinese or Japanese, word segmentation is a challenging task. In the German language the existence of

compounds causes remarkable difficulties. In this language we can express the same concept using a compound or a noun phrase so the choice of the way in which we treat the compounds becomes important.

The existence of many derivational suffixes as well as the use of numerous inflectional suffix in Hungarian or Finnish language, their use for names in the case of Czech or Russian, are also not easy issues to deal with (Dolamic & Savoy, 2009; Dolamic & Savoy, 2009; Savoy, 2008; Dolamic & Savoy, 2009). So, linguistic variability for different families of languages raises particular problems in the development of an IR system.

When switching from one language to another, morphological changes as well as differences in syntax or semantics are the aspects that should also be taken into account (Nie & Savoy, 2008). Finally we are limited to minimal changes that are related to morphology. It is obvious that if a system takes into account syntax, semantics or pragmatics, these aspects will also be modified, at least in part, when shifting from one language to another.

The vast majority of European languages have similar characteristics in lexicon, morphology or syntax because of their common Indo-European origin. Of course some languages do not belong to this family as Finnish, Hungarian, Basque or Arabic found in Malta. The Latin alphabet has 26 letters fail to meet the needs of the French language and accents (e.g., “à”, “â”, “é”, “è”, “î”, “ç”) or ligatures (i.e., “æ”, “œ”). This is also the case in Spanish, German, Italian or Swedish. Russian and Bulgarian languages require the introduction of the Cyrillic alphabet with 32 letters while Arabic requires the presence of an alphabet of 28 letters (whose shape can vary if the letter is isolated, at the beginning, the middle or end of a word).

Many languages have spelling variations (several possible spellings for the same word) that may raise difficulties in matching between the query and documents. As in English: “color” and “colour” or “data base” and “database” or in French: “cowboy” and “cow-boy” or “eczéma “and “exéma”. In some cases, the difference lies in accented letters: “Québecois” and “Québécois”. The presence of specific words, in particular those of foreign origin or infrequent also tend to produce various possible forms: “Gorbachev”, “Gorbacheff” or “Gorbatchev”. For the German language, the

spelling reform of 1996 has generated a lot of trouble, leaving the words with two or more possible spellings, as in: “Jogurts” and “joghurt”.

The existence of homographs in almost all languages is another obstacle in the process of finding relevant documents. Homographs are words with the same spelling but different meanings (e.g., “To book a hotel”, “to read a book”, “I saw a man with a saw”).

Different characteristics of different languages produce particular problems while providing the stemming procedure for a language and furthermore make it necessary to have different stemming strategies for different languages. For instance, for European languages, even if they belong to the same Indo-European family, different suffixes are used to note, for example, the number. French makes the plural by adding “-s”, which is the case for English or Spanish, but the Italian morphology plays on alternating vowels (e.g., “aeroporto” and “airport”). In German different forms are used to indicate the plural: an accent (e.g., “Apfel” and “Äpfel”), the suffixes: “-er” (e.g., “Bild” and “Bilder”), “-en” (e.g., “Staat” and “Staaden”) or “-e” (e.g., “Boot” and “Boote”), without one form being much more common than another.

Also, dealing with derivational suffixes is not always as simple as one can imagine. Cases where the accent of a letter changes, the last letter doubles, changes or eliminates are complicated cases to handle specially in indexing phase (e.g., “stem” and “stemming” or “lazy” and “laziness”).

Compounds are present in all European languages. However, their use is more common in some languages than others and their format can vary from one language to another. In information retrieval two problems arise while dealing with compounds. First, term weighting should take the presence of compounds into account by assigning a greater weight to them. Second, as the same concept can be expressed in different forms, partial matching between queries and documents is more complicated (e.g., “Bundesbankpräsident” and “der Präsident des Bundesbank” in German). To overcome this linguistic variability, several authors have proposed to automatically decompose the compounds in queries and documents. However, this automatic decomposition is not error free as the German word “Frühstück” (breakfast) which could be divided into “Früh” (early) and “Stück” (piece, part). So

this process of decomposition is important as, sometimes, inclusion of components of the compound will add noise in the query (or document), thus makes detection of relevant documents more problematic.

Presence of hyphens in English, either to split up vowels (e.g., co-education) or to join nouns as names (e.g., Hewlett-Packard) It is easy to feel that the first example should be considered as one token while the other case is not as clear as the first one. So dealing with hyphens automatically can be complex. In French using apostrophe before a word beginning with a vowel (e.g., l'ensemble, "the set") using hyphens with pronouns in imperatives and questions (e.g., donne-moi, "give me") makes again some complexity in automatic tokenization process (Manning et al., 2008).

While working with Far East languages some different kinds of problems may occur as some of them like Chinese, Japanese or Korean have unique characteristics. First, the words are not explicitly marked in Japanese or Chinese. In both languages, a sentence is a sequence of symbols without spaces. Thus when indexing a document, a preliminary step, usually automatic, is to segment the text to be able to work with.

The presence of unknown words (not stored in a dictionary as proper names) raises another difficulty. Chinese language is very tolerant to the creation of new words, often composed of two or more ideograms. This might be due to using modern communication media (e.g., Internet) and the rapid creation of new technological concepts (e.g., mobile phone). If a new word is not recognized by an approach exploiting a dictionary, it will be segmented with separate ideograms, which causes a loss of accuracy when searching.

Also in this family of languages the number of ideograms is very high. There are more than 13000 for traditional Chinese or 7700 for simplified Chinese. Japanese combines Chinese characters with two other syllabaries, and the Latin alphabet. In Korean alphabetical system each syllabic block has usually between two and four letters (giving a total of 11,172 distinct possible syllabic blocks). In this language, words are explicitly defined, but as in German, it has very many compound words, usually generated by concatenation of adding various simple words and suffixes (Nie & Savoy, 2008).

What mentioned above were some examples of the challenges we have to deal with when processing the textual data. Now once the documents are pre-processed and indexed the second step in the IR process is when the user formulates a query and provides it to the system. However the user's query may be a poor representation of her/his information need and as such, even an effective IR system may not return relevant results. In other cases the user queries are very short (2-3 terms on average) which leads to a poor retrieval effectiveness because of vocabulary mismatch. As a classical method we handle this problem by query expansion techniques (by adding additional terms to the user's initial query). Or in the case of an interactive system, in light of the retrieved documents, the user may choose to re-formulate her/his query to be more specific.

2.2 Cross Language Information Retrieval (CLIR)

Cross-language information retrieval (CLIR) is an extension of classical IR. In CLIR users can query across two (bilingual) or multiple (multilingual) languages. In classical IR the collection and the queries are all in one same language (within-language retrieval). In CLIR the collection and the user query are in different languages. This can happen in different scenarios. Users query a monolingual collection in more than one language different from the collection language. Users query a multilingual collection in one or several languages. Or the collection has mixed language content documents and the users build their queries in one or several languages. A system might cover some or all the above mentioned cases. CLIR supports at least the bilingual case and if a system supports all these scenarios it is called a Multilingual Information Retrieval (MLIR) system. It is clear that such systems cannot match the query and the documents without translating them into a common language. As shown in Figure 2.2 and 2.3, a CLIR system follows the same structure as a classical IR system adding a translation phase to it. A system can translate the queries (Figure 2.2), the documents (Figure 2.3) or both. The choice of what to translate and how to translate makes the difference between different systems (Peters et al., 2012).

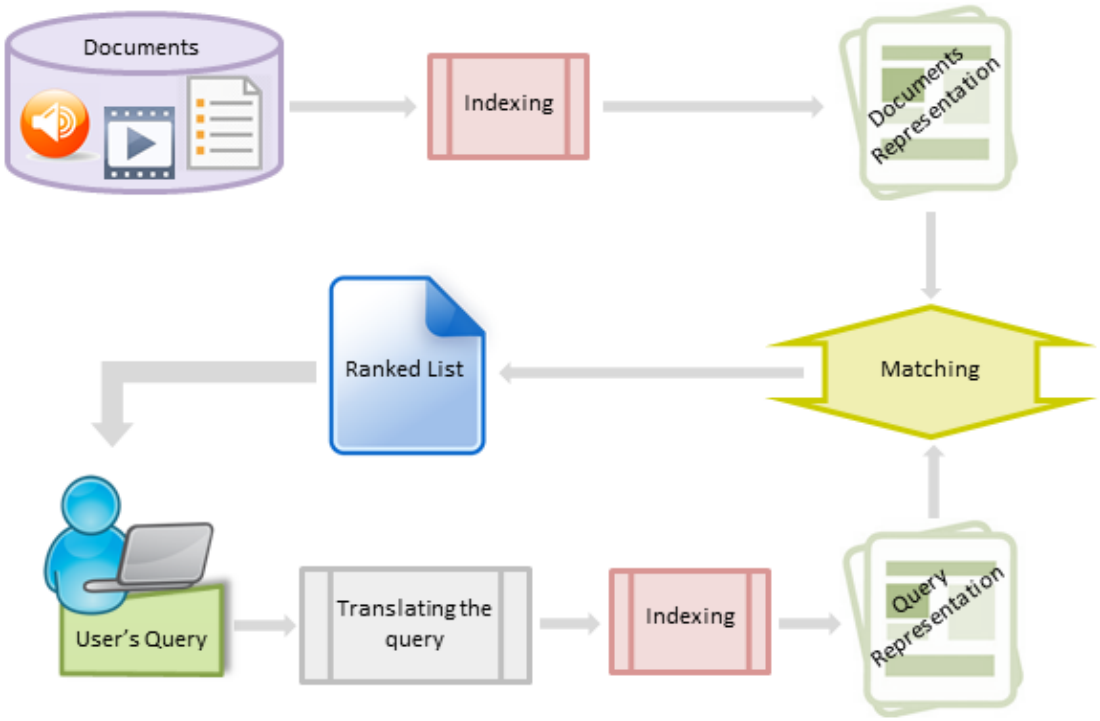

Figure 2.2 Structure of a CLIR system using query translation

If we have the documents in one language and the queries in another language (bilingual retrieval) then either we translate the queries into the language of the documents or we translate the documents into the language of queries. Afterwards we can conduct our retrieval as in a classical within-language system. But if we have our collection in several languages the process will become more complex. In such a case we can adopt two different strategies. As the first method we can translate both the documents and the queries into a common single language and accomplish the retrieval process. As the second method we can index each document collection within its language (producing one index per language), conduct the retrieval using the corresponding queries and producing a result list for each language. As a final step we need to merge these different result lists on order to produce our final single ranked result list. There are different merging strategies than we can use in order to merge the results produced for each language. We will talk about some of these merging strategies later in Section 5.7.2.

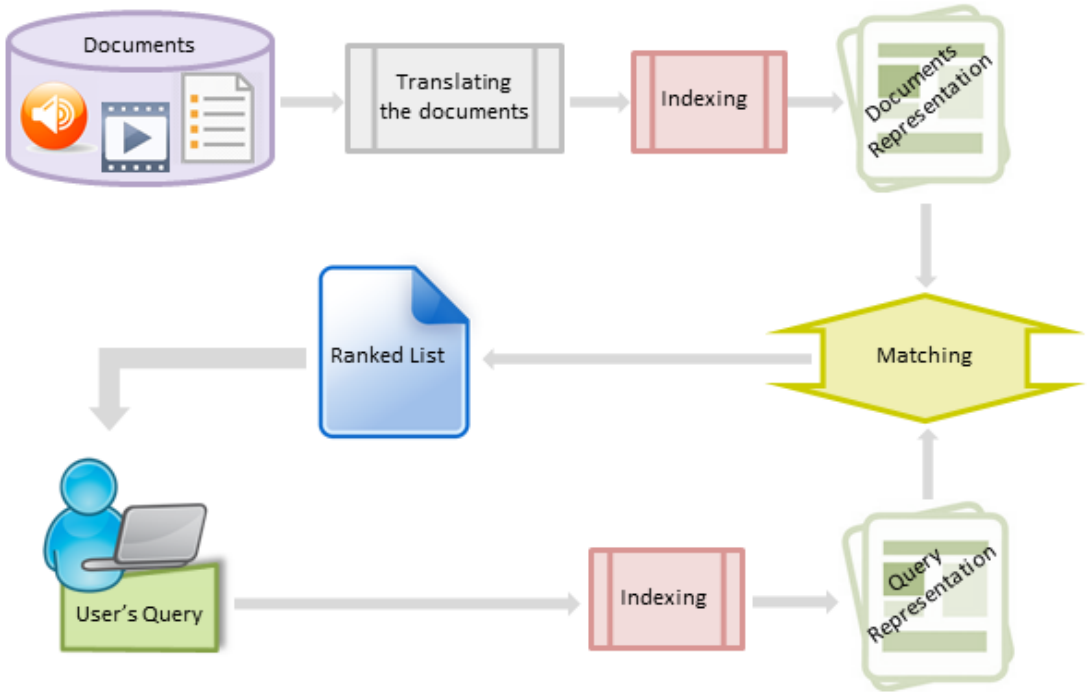

Figure 2.3 Structure of a CLIR system using document translation

2.2.1 Challenges

Obviously in a cross-language IR the additional challenges emerge with the translation. One of these problems is word sense disambiguation. First of all a single word can convey different meanings. Moreover one word can be translated into different words in a target language. Consequently finding the proper translation is not trivial. This problem becomes bigger when we deal with the lack of context which, as mentioned before, is the problem with short queries. So the challenge of a CLIR system is to avoid the ambiguity in translation and to produce the closest meaning as in the original language (Grossman & Frieder, 2004).

As another problem we can mention the proper name matching problem when having several languages. Proper names might be spelled differently in different languages (e.g., London and Londres). Besides the different spellings for one proper word exists even within a single language. These facts make it difficult to transliterate the proper names into the names in the target language. Clearly the challenge

becomes more important when the two languages use different character sets (e.g., English with Arabic or Japanese) (Grossman & Frieder, 2004).

When the number of languages that the system should handle increases the above-mentioned problems become more complex. More languages we have more difficult becomes finding direct translation resources. And when using a pivot language for translation the quality of the translation will at some point decrease and also the ambiguity would increase (Peters et al., 2012). We will talk more about translation in Chapter 4.

2.3 Domain-Specific Information Retrieval

Domain-specific IR systems are used to retrieve information in a given field of knowledge, e.g., patent, genomics, chemical, etc. These systems are thus limited to specific document formats, particular topicalities and terminology. Consequently they often deal with users who are specialist or have a strong interest in that particular domain of search. Depending on the domain and its target users there might be some differences between the architecture of these systems and their expectations. For instance in a professional search like patent or medical retrieval the user needs to retrieve as much as results as possible which makes these systems recall-oriented professional search systems. On the other hand if we deal with general users who just have a passion for a specific field of knowledge, the system does not need to retrieve all possible results. In such a case retrieving some relevant documents at top ranks will normally fulfill the users' expectations.

2.3.1 Challenges

In a domain-specific IR (either monolingual or cross lingual) we are dealing with the same difficulties as in a classical IR and CLIR. But certainly here again each domain brings its own difficulties along with its specific characteristics. For example in a patent retrieval the problem comes with the fact that in this domain the queries are provided as a whole document (in a patent format) (Fautsch, 2009). As another

example we can refer to the domain of Genomics where the presence of different spellings for the same term causes challenges (Yu & Agichtein, 2003).

Methodology (IR) 3

The whole IR process, as explained before, can be divided into different steps. The process starts with building the index. When the system receives a user's query it should find matches for the query within the index. The similarity score between each retrieved document and the query is then calculated using a weighting model. The system now can present the ranked list of the retrieved documents to the user. As all IR systems follow this pattern, it is important to study these different steps in detail.

In the first three sections of this chapter we will describe the indexing process, term weighting and different IR models. Afterwards in Section 3.4 we discus different evaluation technics for evaluating the performance of the systems. And finally in Section 3.5 we will talk about query expansion technics which are used for improving the retrieval performance.

3.1 Indexing

During indexing the document collection as well as the queries is analyzed in order to produce a list of keywords out of each document. This list contains the most significant words carrying most important concepts of the related document. The list of keywords which summarizes the document contents is called document descriptor or document surrogate. Indexing makes the documents representable to the system, creates a searchable data structure, easy to exploit for the system, and hence reduces the cost of the search (Boughanem, 2008; Kowalski, 1997).

There are three ways of indexing:

1. Manual: an expert analyzes documents. In this way quite good results might be ensured.

2. Semi-automatic: indexation is done automatically but a human expert does the last selection.

3. Automatic: the whole process is completely computerized.

The process of automatic indexing is usually a combination of different automatic treatments, mainly:

- Tokenization
- Stop words elimination
- Lemmatization and stemming

3.1.1 Tokenization

This process determines the words in a text and converts the document to a collection of lexemes or terms. It handles this goal by treating the spaces, digits, hyphens, punctuations, the case of the letters, etc. In this way matching between the queries and documents can take place regardless to superficial differences between words (for example USA" can match with "U.S.A." or "naive" with "naïve"). Tokenization is normally accomplished by performing different procedures in different steps. The first step is the task of dividing a text into tokens (a sequence of characters that make together semantic units useful for processing) and often throwing away certain characters, such as punctuation marks or numbers. Afterwards, we usually replace the uppercase letters with their corresponding lowercase letters. Then once the document is broken up into tokens, the next step could be the normalization process that deals with accents or diacritics (Manning et al., 2008; Baeza-Yates & Ribeiro-Neto, 1999; Boughanem, 2008)

3.1.2 Stopwords Elimination

Certain types of words such as closed-class part-of-speech (POS) categories (e.g., prepositions, determiners, pronouns, conjunctions) or some names (e.g., "year" or "day") that are used everywhere in a language do not carry significant semantic meaning by themselves (their role is to modify other words or define grammatical relationships). They do not have a real ability to distinguish relevant documents to a

subject of those which are non-relevant. So these common forms in a language, known as stopwords, can be removed from the documents without violating the semantic of the text. Consequently, there is usually no need to index them and mention them in the document descriptor. Stopwords are usually the words with a high frequency of occurrence in a document (e.g. the). By eliminating the stopwords the produced set of features for a document becomes smaller in size. Consequently, the built index for the document will reduce its size (by ~30–50%) and the execution time for searching the queries will also be reduced (Dolamic & Savoy, 2010; Baeza-Yates & Ribeiro-Neto, 1999; Nie & Savoy, 2008; Boughanem, 2008; Büttcher et al., 2010).

3.1.3 Stemming and Lemmatization

Words that do not appear in the stoplist are considered as candidates to appear in the index. However, if one uses these surface forms directly to generate the index, the system will create separate entries for words that vary in form but corresponding to a same or a similar meaning (morphological variants of a word). In a given text there always exist different forms of one word differing according to the role of the word in the sentence (e.g., “leave”, “leaves”, and “leaving”). This is called inflection in linguistics. In this case suffixes are added at the end of a word to indicate its number (singular or plural), gender (masculine, feminine or neutral), time, mode or person (for verbs). Another phenomenon is derivation where new word is created from existing words usually by modifying the POS category (e.g., “national”, “nationally”, and “nationalize”). All these morphological variants, having the same root, carry the same or similar concept. Therefore, it is obvious that there is no need to index all these words, rather it is enough to group these words of similar meaning and treated them under only one lexeme that carry the concept of them all (Manning et al., 2008; Nie & Savoy, 2008; Boughanem, 2008). This is the aim of stemming and lemmatization.

The stemming algorithm (to extract the root form of the terms) differs from language to language. Obviously such an algorithm should be designed according to each language’s morphology and grammar. Taking the English language as example we can think of many different methods. One option could be removing the

inflectional and derivational suffixes (e.g., "-s", "-ed", "-ing", "-ion"). Such a removal should be monitored by quantitive or qualitive restrictions, for instance the "-ing" from "king" should not be removed as "-ize" in a term like "seize". In addition removing suffixes for certain words result a wrong spell (e.g., "absorption" should be changed into "absorb" and not "absorp" or "running" into "run" and not "runn"). In such cases additional rules, known as conflation rules, are needed. For irregular cases such as irregular verbs a table (dictionary) listing the transformation of each individual variant could be used. We can also consider prefix removal (e.g., "kilo", "milli", "micro") when stemming. Also a recognition phase for proper nouns can be applied to the text in order to prevent the proper nouns from being stemmed (Fautsch & Savoy, 2009; Boughanem, 2008; Sanderson, 1996).

Another strategy that can replace stemming is *n*-grams. *N*-grams is the act of splitting a word into overlapping sequences of *n* characters. For example the result of splitting the word "system" into its 3-grams will be: "sys", "yst", "ste" and "tem". So here instead of replacing the terms by their root forms they will be replaced by their *n*-grams. The value of *n* is chosen according to the underlying language characteristics. Nevertheless the method remains the same regardless the underlying language. The impact of this technique differs from language to language but in general the query execution time and the index size will increase (Büttcher et al., 2010). As another language-independent strategy we can mention trunc-*n* method this method is the process of truncating a word by keeping its first *n* characters and cutting of the remaining letters. For example applying this method to the term "system" with *n*=5 our index term will be: "syste".

3.1.3.1 Stemming vs. Lemmatization

Stemming as mentioned earlier is not error-free. A usual problem is the problem of over-stemming where "general" becomes "gener" or "organization" becomes "organ" or under-stemming where "create" and "creation" do not categorized under the same root. This is where the difference between stemming and lemmatization occurs. In lemmatization the goal is to return the dictionary entry of a word, known as "lemma". So here a more profound morphological analysis and maybe POS recognition is needed in order to obtain the precise lemma. Therefore with performing, properly, the

use of vocabulary or morphological analysis the aim of lemmatization is to remove the inflectional endings only (Manning et al., 2008; Fautsch & Savoy, 2009).

3.2 Term Weight

In the document descriptor a numerical weight is assigned to each term. This value represents the importance of each word within the related document. Term weighting is one of the fundamental functions in IR and is the backbone of most IR models and approaches in order to determine the relevance score of a document to a certain query. Using statistical methods is the most common way to calculate this weight. These methods are based on two factors:

1. *tf* (Term Frequency): number of term's occurrence within the document (local weight).
2. *idf* (Inverse Document Frequency): term's frequency of occurrence within the collection (global weight).

Inverse document frequency of term *t* in a collection consists of *N* documents is defined as:

$$idf_t = \log \frac{N}{df_t}$$

df_t (document frequency) is the number of documents in the collection that contain the term *t*. So if a term is appears in many documents within the corpus, its *idf* value will be low and if a term is a rare one in the document collection then its *idf* will be high.

The calculated weight using the product of these two factors is called *tf idf* weight. This weight will be high if for a given document, the term appears many times in that document (*tf*) and appears rarely in the other document (*idf*). The *tf idf* weighting scheme is a good estimation to show the importance of a word in a document particularly in a corpus consisting of documents with more or less the same sizes. As we can see the long documents have an advantage over the shorter ones (Manning et al., 2008; Boughanem, 2008; Sanderson, 1996). For a corpus consisting

of documents with variable lengths it is recommended to consider the document length in the calculation of *tf idf* (Robertson & Walker, 1994; Singhal et al., 1996). The Okapi weighting function is one of the most common weighting functions of this method:

$$w_{ij} = \frac{tf_{ij}\,(k_1+1) \times \log\frac{N - df_1 + 0.5}{df_1 + 0.5}}{k_1 \times \left((1-b) + b\frac{l_{d_j}}{avg_{dl}}\right) + tf_{ij}}$$

where:

- tf_{ij} is the frequency of the term t_i in the document d_j.
- l_{d_j} and avg_{dl} are the size of the document d_j and the average of the documents sizes in the collection.
- k_1 and b are constants.
- N is the number of documents in the collection.
- df_i is the number of documents in which the term *i* occurs.

3.3 Retrieval Models

The core activity of an IR system is to define which document in a collection is relevant to the user's query. When user sends a request to the system, the system will return an ordered list of relevant documents with documents at the top of the list considered to be more relevant to the user's search. The system makes this decision based on a ranking algorithm (Baeza-Yates & Ribeiro-Neto, 1999). These ranking algorithms are based on different retrieval strategies. A retrieval strategy allocates a measure of similarity (called also as RSV for Retrieval Status Value) between the query and each document (Grossman & Frieder, 2004). There are sets of different assumptions, regarding the document relevancy that can be adopted by a retrieval strategy. These fundamental assumptions, which are the basis of the ranking algorithms, establish different IR models. In other words the selected IR model for a system determines the criteria with which the system decides what is relevant and what is not (Baeza-Yates & Ribeiro-Neto, 1999). So the main role of an IR model is

to build a certain theoretical framework to measure the similarity between the queries and the documents (Boughanem, 2008).

The three principal IR models are:

- Boolean Model
- Vector-Space Model
- Probabilistic Model

3.3.1 Boolean Model

The Boolean model is the oldest IR model. This model is based on the set theory and Boolean algebra. In this model the documents are presented as sets of terms and the queries are formulated as a Boolean expression on terms, linked by logic operators AND, OR and NOT. So it lets the retrieval mechanism to use set operators (union, intersection and difference). This model considers that the index terms are present in a document or not. So the term weight in this model is a binary weight: {1, 0}. Retrieval strategies in this model are using binary decision criteria. They are based on exact match means that the similarity between a query and a document is also a binary value: it is 1 if the document completely fulfills the criteria given in the query and 0 otherwise. Consequently in this model a document is either relevant or not relevant and there is no partial answer to the query (Boughanem, 2008; Pasi, 2010). The decision to retrieve or not a document is clear and can be easily explained to the user. The main advantage of this model is its simplicity and the clean formalization. The main disadvantages are:

- Retrieval based on exact match may results the retrieval of either too few or too many documents (Baeza-Yates & Ribeiro-Neto, 1999).
- Results cannot be ranked so it prevents a good retrieval performance (Baeza-Yates & Ribeiro-Neto, 1999). It is known that a non-binary term weight remarkably improves the performance (Boughanem, 2008).
- It is often difficult for users to formulate their request as Boolean expressions (Baeza-Yates & Ribeiro-Neto, 1999).

3.3.2 Vector-Space Model

This model is based on linear algebra. In this model all index terms in both queries and documents are weighted with a positive non-binary value, as shown in Figure 3.1. The term weights are usually calculated using *tf* and *idf* measures. Both queries and documents are presented as vectors (Grossman & Frieder, 2004; Boughanem, 2008). These vectors, representing the terms in the documents and the queries are defined as follows:

$$\overrightarrow{d_j} = (w_{1j}, w_{2j}, \dots, w_{tj})$$

$$\vec{q} = (w_{1q}, w_{2q}, \dots, w_{tq})$$

where:

- t is the total number of index terms.
- w_{ij} indicates the weight associated to term i in document j.
- w_{iq} indicates the weight associated to term i in query q.

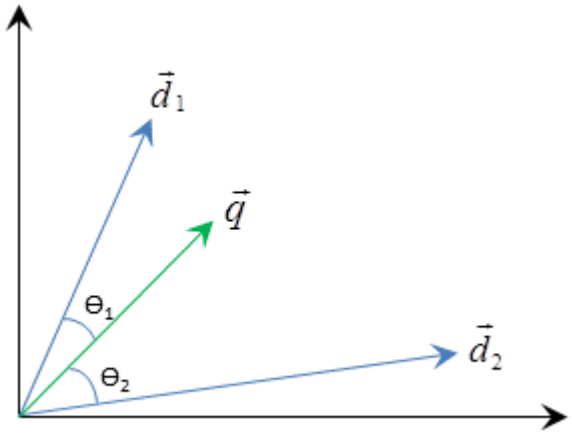

Figure 3.1 Cosine of Θ is assumed as sim(d_j, q), $\Theta_1 < \Theta_2$ so d_1 will be ranked higher than d_2

Matching mechanisms evaluate the closeness of these vectors to calculate the similarity (relevance score) between a document and a query. This closeness can be determined by calculating the cosine of the angle between these two vectors:

$$sim(d_j, q) = \frac{\vec{d_j}.\vec{q}}{|\vec{d_j}| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{iq}^2}}$$

where $|\vec{d_j}|$ and $|\vec{q}|$ are the norms of the document and query vectors.

So here instead of defining a document as relevant or not relevant, the retrieved documents are ranked according to their degree of similarity to the submitted query (partial matching). Thus a document can be partially related to a query (Baeza-Yates & Ribeiro-Neto, 1999; Pasi, 2010).

3.3.3 Probabilistic Model

The probabilistic model defines the IR problem in a probabilistic framework. It was first proposed by (Maron & Kuhns, 1960). The idea is based on probability ranking principle (PRP) where the result documents are scored according to the probability of relevance between a document and a given query (Robertson, 1997). A document is either relevant to the query or it is not. So here the similarity between the document and the query is calculated as the probability that the document will be relevant to the query. This model estimates the probability that the document is in the set of relevant documents or in the set of non-relevant documents. The document d_j will be selected if $P(R|d_j)$ (the probability that it will be relevant) is higher than $P(NR|d_j)$(the probability that it will be non-relevant) (Boughanem, 2008). The retrieved documents can be sorted according to the following formula:

$$sim(q, d_j) = \frac{P(R|d_j)}{P(NR|d_j)}$$

by applying Bayes' law:

$$P(R|d_j) = \frac{P(d_j|R)P(R)}{P(d_j)}$$

$$P(NR|d_j) = \frac{P(d_j|NR)P(NR)}{P(d_j)}$$

so the documents can be ordered according to:

$$sim(q, d_j) = \frac{P(d_j|R)}{P(d_j|NR)} \times \frac{P(R)}{P(NR)} \propto \frac{P(d_j|R)}{P(d_j|NR)}$$

As $P(R)/P(NR)$, being a constant, does not influence the rank order we can remove it from the formula. So: In different probabilistic models, different approaches are used to estimate these probabilities. In the following sections we present three of the most important models.

3.3.3.1 Language Model

A language model or a statistical language model is a probabilistic approach to generate a piece of a text in a particular language (Grossman & Frieder, 2004). It thus models the arrangement of words in a language and measures the probability of observing a sequence of words in a language. In a language model each term or sequence of terms accepted by the model has a probability of being generated by the model.

In the classical probabilistic models the probability that the document meets the query criteria is estimated. The basic assumption in these models is that a document is considered as relevant if only it is similar to the query. The language models based on a different assumption: while a user, interacting with an IR system, provides a query, s/he already has in mind one or more documents which s/he wishes to retrieve. In other words the user inferred the submitted request according to the documents that s/he has in mind (Boughanem, 2008). A document is considered as relevant if only the query is similar to that inferred (generated) from the document. So the main idea here is to order the documents according to their likelihood of generating a query (Grossman & Frieder, 2004). So the probability that the query has been inferred from the language model of the document should be calculated.

Formally, let M_d be the language model of document d then the relevance of d to a query q is estimated as $P(q|M_d)$ which means the probability that query q is generated by M_d:

$$sim(q, d) = P(q|M_d) = P(t_1, t_2, \dots, t_n|M_d) = \prod_{i=1}^{n} P(t_i|d)$$

$P(t_i|d)$ can be estimated based on the maximum likelihood estimation (MLE) as:

$$P(t_i|d) = \frac{tf(t_i|d)}{\sum_r tf(t_r|d)}$$

where $tf(t_i|d)$ is the frequency of term t_i in document d.

However, with this type of estimation when query term does not exist in the document systematically the similarity will be zero. To overcome this problem smoothing techniques must be used. Smoothing is to assign nonzero probabilities to terms that do not appear in the document (Boughanem, 2008).

3.3.3.2 Divergence from Randomness

In Divergence From Randomness (DFR) approach, proposed by Amati & Van Rijsbergen (Amati & Van Rijsbergen, 2002), we consider that the term distribution in documents is a random process. Therefor it ranks the documents according to the probability that a term distribution in a document would take place randomly (Büttcher et al., 2010). A term is considered as informative if its distribution in a document and its distribution in all documents are different. This method generates the similarity as:

$$sim(d_j, q) = \sum_{i=1}^{q} w_{ij}.qtf_i$$

where qtf_i represents the term t_i frequency's in the query.

This approach considers the term weight as the product of two informative content functions (Inf_1 related to all documents and Inf_2 to the elite set of the term):

$$w_{ij} = Inf_1.Inf_2$$

where $Inf_1 = -\log_2 prob_1$ and $Inf_2 = 1 - prob_2$

$prob_1$ is the probability of obtaining by pure chance (according to the chosen model of randomness) term frequency (*tf*) occurrences of a term *t* in a document *d*. A small $prob_1$ shows that the term *t* is not distributed according to the frequency given by the underlying model of randomness. So *t* is considered as a term that provides informative content of the document.

Prob₂ is defined regarding only the set of all documents in which a term occurs (known as elite set of the term). It is the probability of occurrence of a term within a document respecting its elite set. It is related to the risk 1 - *prob₂* , of considering a term as a good descriptor of the document when the document is compared with the elite set of the term. When *prob₂* of a word frequency within a document is relatively low with respect to its elite set, the level informative content provided by this word is relatively high (Amati & Van Rijsbergen, 2002).

3.3.3.3 Okapi

BM (Best Match) is based on the 2-Poisson model (Harter, 1975). It takes into account both term frequency and document length in order to estimate the probability that a document is relevant to a given query. The similarity function BM25 model (known as Okapi) (Robertson et al., 2000) uses is as follows:

$$sim(d_i, q) = \sum_{t_j \in q} qtf_j \cdot \log\left[\frac{N - df_j}{df_j}\right] \cdot \frac{(k_1+1)\cdot tf_{ij}}{K + tf_{ij}}$$

where:

- $K = k_1 \cdot \left((1-b) + b \cdot \frac{l_i}{avdl}\right)$
- qtf_i is the frequency of term t_j in query q.
- l_i is the length of document d_j.
- *avdl* is the average document length.
- b and k_1 are constants typically set to b=0.75, k_1=1.2 but can be modified empirically according to the underlying collection.

3.4 Evaluation

Evaluation measures the effectiveness and the performance of an IR system. This can be done in different ways. Whether by considering the number of relevant document that are retrieved for a given query or by taking into account the order in which these documents are ranked or yet time and cost concerns. Which aspect is the most

important to evaluate and with which measures, mostly depends on the main tasks of the system. Evaluation helps to recognize if the users are satisfied or not and leads to distinguish which aspects should be changed or added in order to improve the system. That is why many works has done in the field of the evaluation of IR systems. Two major aspects in evaluation are: efficiency and effectiveness.

3.4.1 Efficiency

Efficiency deals with time, space and cost (Büttcher et al., 2010). The shorter time lag (the average interval between the search request and the results), the smaller space used, the better the system is evaluated to be. This evaluation is referred to as performance evaluation. This measurement is concern mostly the systems with a precise functionality, which is usually not the case for an experimental IR system (Baeza-Yates & Ribeiro-Neto, 1999; Rijsbergen, 1979).

3.4.2 Effectiveness

Effectiveness shows the amount of relevant documents that the system retrieves according to a certain query (with respect to the total number of retrieved items). In other words how well the system is functioning (Büttcher et al., 2010). The more a retrieval system is effective the more its users are satisfied. This kind of evaluation is referred to as retrieval performance evaluation (Baeza-Yates & Ribeiro-Neto, 1999). To evaluate the effectiveness of a retrieval system two main measurable factors are used: precision (the proportion of retrieved documents that are relevant) and recall (the proportion of relevant documents that have been retrieved). These two factors are the most important aspects used in evaluation process (Rijsbergen, 1979).

3.4.3 Evaluation Measures

Various evaluation measures have been proposed to evaluate IR system. What to evaluate and thus the choice of one of these measures depends on the nature of the system and its main task. Evaluating a set of experiments done in a laboratory is

different from the one from real life situations. In the same way the expected performance from a question-answering system is different from a professional search system or still a system for web surfers (Manning et al., 2008; Baeza-Yates & Ribeiro-Neto, 1999). Below we describe some of the main evaluation measures.

3.4.3.1 Precision and Recall

Considering that a system generates and returns a set *A* of documents as the response to a given query. *R* is the set of relevant documents exist in the whole collection for the submitted query and R_a is the set of relevant documents in the answer set *A*: precision (P) calculates as the proportion of the retrieved documents that is relevant to the whole number of documents retrieved by the system:

$$P = \frac{|R_a|}{|A|}$$

Recall (R) calculates as the proportion of the relevant documents that has been retrieved by the system to the whole number of pertinent documents in the collection:

$$R = \frac{|R_a|}{|R|}$$

As explained before, in different situation each of these measures is more demanded than the other. Precision derives from the idea that the user (e.g. a web surfer) only wishes to find a reasonable number of relevant documents. Recall measure derives from the assumption that the user (e.g. a professional user looking for a literature review, medical, patent or legal issues) wants to have all relevant documents. Obviously there is a trade-off between the two measures. In order to make this trade-off optimal a measure combining these two metrics known as F-measure has been proposed:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R}$$

The value of the parameter β can be any real number which defines the importance that is given to recall over the precision. When $\beta > 1$, it puts the emphasis on recall while $\beta < 1$ gives the importance to precision (Manning et al., 2008; Sanderson, 1996; Büttcher et al., 2010; Rijsbergen, 1979).

3.4.3.2 Precision at *k* Documents (P@*k*)

As mentioned earlier recall measure assumes the idea that the user wants to find all relevant documents. This used to be reasonable for any system in early years of IR systems where the collections were not as big as now. As the document collection grows, it is become more important to consider the rank of the relevant and retrieved itens.as a result we can define the precision at *k.* this value is the fraction of the relevant documents among the *k* documents retrieved by the system, with a small value of *k* (usually *k* = 5, 10 or 20).

$$\text{P@k} = \frac{|A[1..k] \cap R|}{k}$$

where $A[1..k]$ is the set of the top *k* documents retrieved by the system.

Here it is assumed that only up to *k* documents is retrieved and returned to the user (Boughanem, 2008; Sanderson, 1996; Büttcher et al., 2010; Rijsbergen, 1979).

3.4.3.3 Average Precision (AP)

In calculating precision at *k* document the problem that arises is selecting the value of *k*. Another problem is the fact that with this measure retrieving relevant documents on the top of the result list or near to the value of *k* is viewed as identical. But having relevant items in the highest ranks is certainly better than just before the *k* limit. To solve these problems, another measure is defined: Average Precision (AP). Average Precision is the average of the precision value for each relevant document in the result list. Obviously the closer the retrieved and relevant documents are to the top ranks, the higher the AP is.

$$AP = \frac{1}{|R|} \times \sum_{i=1}^{|A|} relevant(i) \times P@i$$

where *relevant (i)* is 1 if the *ith* document in A (ranked list of retrieved documents) is relevant and is 0 if not (Büttcher et al., 2010).

3.4.3.4 Mean Average Precision (MAP)

Mean average precision (MAP) measure is another popular measure. It is the mean of the average precision value for a set of queries. It is defined as:

$$\frac{1}{|Q|} \times \sum_{q=1}^{|Q|} AP(q)$$

where Q is the set of the queries.

3.4.4 Relevance Assessment

In order to be able to properly evaluate an IR system we need to have information on the relevance of the retrieved documents for each query. Having some knowledge of documents retrieved for a certain query by different systems and how they were ranked make it possible to not only evaluate the systems but also to design a system which produces better results. A test-collection is a combination of a corpus of documents, a list of queries and the related relevance judgments. So having a test-collection available, any IR system can use the corpus and the queries and compare its results to the relevance judgments. Cranfield experiments started this approach by providing full relevance judgments, assessed by human, for each query (Cleverdon, 1991). In this way a human assesses the retrieved documents by a system given a query and marks each document as relevant or not to that query. Clearly producing full relevance judgments for large corpora is too costly to be possible. Therefore for large modern collections incomplete assessments are used instead. This means that only a subset of documents will be judged for each query. One of the most usual methods to choose this subset of documents is pooling. In this method a query is submitted to a number of IR systems each using a different IR strategy. Afterwards the top *k* retrieved documents returned from different systems are combined in order to make the subset of documents to be manually assessed (Sparck-Jones & van Rijsbergen, 1975). If the size of this subset is reasonable for the available resources to judge it then it will be fully judged. Otherwise the subset will be judged in the rank order as far as possible.

Mentioning the facts above, we can see that creating such collections is costly. The collection should be large and as diverse as possible. Finding the set of queries is not trivial and as mentioned assessing the relevancy of each document is a hard task and needs too much manual effort. But once a collection is created it can be easily re-used.

3.4.4.1 Evaluation Campaigns

Many of these above mentioned test-collections are created as part of evaluation campaigns such as TREC (Text Retrieval Conference) or CLEF (Cross Language Evaluation Forum), FIRE (Forum for Information Retrieval Evaluation), INEX (Initiative for the Evaluation of XML Retrieval), NTCIR (NII Testbeds and Community for Information access Research). The philosophy of these campaigns is to provide test-collection in order to: provide the possibility for researchers to discuss their common problems by using common data and producing comparable results. Provide them the facility to evaluate their systems using the same collection and conduct a direct inter-system comparison. It also guarantees the validity of the comparison results, within a given test-collection, by providing the measures of comparison in completely equal conditions (Boughanem, 2008). Also as a test-collection is reusable, this provides researchers the possibility of evaluate and re-evaluate their systems and verify their progress and improvements easily. In this way the key factors of different systems can be quickly recognized. Consequently notable performance progresses are usually seen after the evaluation campaigns (Büttcher et al., 2010).

3.5 Query Expansion and Relevance Feedback

For most of the users it is difficult to formulate their information need in a form of a query. First of all they might not be clear with what they really need. Besides, users normally do not have enough information concerning the collection and the retrieval process. Moreover the existence of synonyms, homographs or spelling variations in natural languages affects the recall of an IR system. In a collection the same concept may be presented using different terms. Moreover same spellings might refer to

completely different concepts. Therefore it is usually useful to reformulate the initial query in order to improve the results. During the reformulation we might expand the original query with new terms and reweight the terms in the query. Query expansion refers to the cases where the initial query is expanded by adding additional terms and phrases. The expansion can be done in many different ways.

Another scenario in which query expansion can be helpful is in a MLIR system when using query translation. A translated term may not convey the exact concept of the original term or sometimes a term might remain untranslated. So this failure to translate or to have a good translation leads to less accurate retrievals. So a solution to overcome this problem is to apply query expansion. In this way by adding related terms to the query the probability of missing concepts will be decreased. The expansion however can be done before the translation step (pre-translation) or after (post-translation) (Peters et al., 2012).

One of the possible query reformulation techniques is *relevance feedback*. It is the processes of revising the initial query submitted to a system, using the relevance judgment information, and then presenting this new query in a second search with the aim of improving the retrieval effectiveness. In this process at a first step the user checks the retrieved documents and indicates to the system which documents in the result list are relevant and/or non-relevant. Based on this information the system then reformulates the initial query and creates a new query. For example new terms extracted from the documents assessed as relevant can be added to the initial query. This expanded query conveys more precisely the information the user needs. The new query goes back to the system's input to produce a new result list. Relevance feedback can go through multiple passes of this kind. The system reformulates the initial query by adding new terms and/or deleting some terms and/or adjusting the term weights. The technique by which the reformulation is implemented depends on the IR model used to construct the system (Sanderson, 1996; Pasi, 2010). The process of relevance feedback is based on the idea that formulating a good query is difficult for the user. But the users can easily judge if the documents are relevant to their request or not. Thus seeing the retrieved document gives the users a clearer image of how they should formulate their needs in order to get the desired result (Manning et al., 2008).

3.5.1 Pseudo-Relevance Feedback

The relevance feedback mechanism is however not widely applied. The reason is that users are not usually proactive and also sometimes after several iterations they may lose the track and do not understand why some documents have been retrieved. Another reason is that sometimes the reformulated queries are too complex or too long which is not desirable for an IR system and increases the response time and cost (Manning et al., 2008). So the alternative methods to this mechanism would be pseudo-relevance feedback (PRF) also known as blind query expansion. In relevance feedback, user indicates some additional input (relevant or non-relevant) on documents in order to modify the initial query or to reweight the terms. Pseudo-relevance feedback automates this manual part of relevance feedback. In pseudo-relevance feedback the user does not provide any feedback. PRF considers the top-ranked documents initially retrieved by the system as relevant ones and uses them as implicit feedback to generate additional terms. In some implementations, we can also consider non-relevant items (e.g., the documents displayed at the bottom of the returned list). PRF is often considered as an effective query expansion approach. Moreover it is particularly attractive because it does not need any external input. However it is important to consider the fact that some of the top-ranked documents might be non-relevant which makes them noisy feedbacks (Shokouhi et al., 2009; Xu et al., 2009; Cao et al., 2008). Besides not all the terms extracted from feedback documents are useful ones. Some studies show that in order to select useful terms for expansion using a term classification method could be useful and overcome the blind query expansion drawbacks (Cao et al., 2008).

Methodology (CLIR) 4

As mentioned before we can divide the whole monolingual IR process into two major parts: indexing and matching. But as mentioned in Section 2.2 in a MLIR system the matching cannot take place directly as the documents and the queries are not in the same language. Consequently a translation phase should be added to the whole process when we are dealing with a MLIR system. The main question is what to translate and which translation method should be used. And these choices are what make the difference between different systems. As for what concerns indexing and evaluation, the process remains the same as in classical IR systems, in this chapter we will only talk about different translation strategies. Finally in the last section we discuss the process of query expansion in a MLIR system.

4.1 Query Translation and Document Translation

Obviously in order to match the queries and the documents either we should translate the queries into the language of the documents (Query Translation (QT)) or the documents should be translated into the language of the queries (Document Translation (DT)). We can however adopt a combination of both QT and DT which means translating both the queries and the documents into an intermediate language. The advantage of using the document translation is that it is less time consuming at the retrieval time as it can be done offline. At the same time we would need more storage requirements and considering the improvements in translation systems we might need to re-translate the whole collection after a while. Furthermore we should know the query language in advance and it cannot be changed. Query translation on the other hand overcomes the disadvantages of DT. But an online query translation it

might prolong the response time which is the sum of the translation time and the retrieval time (Bikel & Zitouni, 2012). Although having short text queries looks easier to translate than a long document but in fact having short queries or queries in the form of some terms might result ambiguous translations (Peters et al., 2012; Grossman & Frieder, 2004).

4.2 Indirect Translation

One way to combine query and document translation is to use a pivot language and translate both queries and documents into this language. The pivot language can be a natural or artificial language. We can also use a pivot language when there is no proper direct translation between two languages. So with the pivot language we can translate these two languages into another. Usually the English language plays this role. In this case the first step is to translate the first language into the pivot language and then from the pivot language into the target language. Obviously we should opt this method if the available resources for translating into the pivot language for both languages are more suitable than the resources for the direct translation. In some cases where the quality of the resources for translating into the pivot language is much better than the one for direct translation, this method may increase the retrieval effectiveness (Dolamic & Savoy, 2009). It is also possible to use two pivot languages in order to produce two different translations of the first language into the target one. The advantage is that by comparing the two translations we might be able to clarify some of the translation ambiguities according to the context (Mikolov et al., 2013).

4.3 No Translation

In the case of some languages the speakers of one language can understand the other language without knowing it. These languages are often from the same language family and the have similarities in grammar, vocabulary and pronunciation (e.g., Swedish and Norwegian or Czech and Slovak). This is called “mutual intelligibility” in linguistics. When we are dealing with such languages we can possibly avoid the

translation phase. In such cases (as the two language have similar vocabularies) strategies such as *n*-grams are useful in order to conduct the matching without any translation of either the corpus or the queries. It is also possible to accomplish matching between whole words using a spelling correction without translation. In this case we assume that one language is a "mis-spelled" form of the other one.

4.4 Translation Methods

There are several translation methods than we can use in CLIR. There is also the possibility to apply a combination of different methods. Three major translation methods are using a machine readable dictionary, using statistical resources to translate and machine translation. We discuss each method in detail in the following sections. Obviously if we find a proper translation (either for the queries or the documents) then we can deal with our CLIR problem the same way as a within-language retrieval problem. But of course this is far from what happens in practice. Several translation problems complicate the process of translation and thus the implementation of CLIR systems.

4.4.1 Machine Readable Dictionaries

By using machine readable dictionaries we translate each word separately regardless its particular meaning in the sentence. In this way each term is simply replaced by all its possible translations. In a good translation we do not need all but the proper term so obviously with this method we produce noise. As the queries are usually short and in the form of a phrase rather than a complete sentence, this method is more appropriate for translating the queries. In using dictionaries studies show that dictionary coverage is important for increasing the accuracy. They show that a dictionary that contains between 3000 and 20000 terms, it linearly increases the accuracy but the accuracy does not increase after that (Demner-Fushman & Oard, 2003). Nevertheless we cannot avoid the major problems with this method even when using it for query translation. These problems are as follows. Translating each term individually from the rest of eh text results ambiguity. Another problem is that

the expressions will not be correctly translated. And the third problem is the fact that in a dictionary we find the lemma for each term and not all the possible variations. If we apply stemming to solve the latter we will add more noise and ambiguity to the result translation (Darwish & Orad, 2002).

4.4.2 Statistical Approaches (Parallel and Comparable Corpora)

The idea behind the statistical approaches is to learn translation rules from parallel or comparable corpora. Unlike dictionary-based translation statistical approaches try to translate the whole sentences. Consequently using statistical approaches we can solve to some extend the problem of ambiguity and translating the expressions. The parallel corpora are parallel texts which are perfect translations of each other. So we can use them to learn which words/phrases in two different languages are the translations of each other and use this information as training data. To do so we use statistics on the available texts. Different methods are proposed to do the word alignment.

Obviously finding parallel corpora for certain pair of languages is more difficult than others (e.g., Finnish/Hindi compare to Finnish/Swedish). If there are no parallel corpora for a certain language or in a particular domain then we can use comparable corpora which contain texts in different languages. These documents are not the literal translation of each other but they are similar in content. They should be from the same period and on the same topic (ex. newspaper articles on the same topic). There are different ways to use comparable corpora. As one option we can simply search the query in one of the collections and retrieve the relevant documents. Afterwards we can map the retrieved documents to their comparable documents in another language. As another option we can use these documents to extract bilingual terms lists and then use them for query translation.

4.4.3 Machine Translation

The use of machine translation (MT) in CLIR seems evident. However, this method also has also its own limitations. As mentioned before usually users query are short

and they are not complete sentences but a sequences of terms. Consequently if we use MT for query translation we will probably not have the desired result.

4.4.4 Combination Approaches

Another suggested solution to the translation problem is to use different translation methods and combine them. One possible way to combine different methods is to apply them separately on the data and produce the retrieval results and finally merge these results into a single result list. Different studies show that the combination of different methods is a promising alternative to using each method individually. In this way the lexical coverage will be improved. Besides we can cover more languages or domains as we are not limited to what a certain method offers us.

4.5 Fusion

As mentioned in Section 2.2 when we have the document collection in many different languages we can adopt the following strategy in order to search the collection: a centralized approach or a distributed one. In a centralized approach we translate the query into all the existing languages in the collection and then combine those in order make a one query out of all the translated ones. In a distributed approach we index each collection in its own language, translate the queries into each of the existing languages in the document collection and finally search each collection using the corresponding queries. In this way we will produce one ranked result list for each of the existing languages. In order to produce the final results we need to merge these result lists. We can do this “merging” or “fusion” in different ways. We have the option of ignoring the scores and just choose one document in turn according to their position from each ranked list. In this method we consider that the results from different languages are all produced under equal conditions. In this case if the relevant documents are wrongly ranked in one of the result list it will harm the whole performance. Another option will be to take into account the scores and map them into comparable units (normalize the scores). We can use different methods for mapping the scores.

4.6 Query Expansion

With the same concept as in monolingual IR systems, we can apply query expansion technics to CLIR systems. But in a CLIR system we can apply query expansion in two ways: pre-translation expansion or post-translation expansion. In pre-translation expansion, we first expand the query and then translate it. The advantage is that in this way we will provide more contexts for the translation process (Bikel & Zitouni, 2012). Some studies show that in this way we can improve the precision (Ballesteros & Croft, 1997). On the other hand post-translation expansion is the same as in classical IR. Ballesteros & Croft (1997) show that this method might reduce translation errors. For example when applying pseudo-relevance feedback applying analysis on the results helps in order to identify wrong translations.

Experiments 5

5.1 Introduction

In our experiments we try to design an IR system to retrieve information from data collections containing data from only one specific field of knowledge (domain-specific IR). We also aim to design a system that functions when there are different languages in one corpus and when users use different languages to express their needs. With these goals in mind we conduct ad-hoc retrieval experiments on different monolingual corpora and one multilingual collection dedicated to cultural heritage objects.

The CH collection is characterized by short text descriptions. We attempt to evaluate the influence of document and query structure on the search quality. We try to investigate the impact of different IR models and indexing strategies on the retrieval effectiveness for different natural languages. We also aim to investigate and propose effective stemming algorithms for different languages. Moreover we consider integrating translation into the search process and adapting our system for bilingual and multilingual IR. As our other objective we examine the effect of different query expansion techniques in order to improve the search quality. And finally we suggest producing the final output of a search by merging the results obtained from different approaches and we investigate the improvement of the search quality when applying different merging strategies. Accordingly we divide our experiments into four major parts: monolingual retrieval, bilingual retrieval, multilingual retrieval and query expansion. At the first 3 parts we aim to explore:

- The efficiency of different IR models, stemming methods and indexing strategies in searching cultural heritage objects.

- The effectiveness of different query translation methods in a bilingual and multilingual retrieval.
- The impact of pseudo-relevance feedback on enhancing the retrieval effectiveness.
- And finally the effect of application of data fusion operator on the retrieval performance.

In the last part we evaluate:

- The relative effect of various query expansion and semantic enrichment techniques, using external resources, on the retrieval effectiveness in a domain-specific search.

For evaluating the retrieval performance we choose the MAP (mean average precision) measure in all our experiments. This is computed with the TREC_EVAL program where MAP value is computed based on, maximum, 1000 retrieved items per query. It is important to mention that when computing the MAP, the topics with no relevant items are not taken into account.

In each part of our experiments we apply different IR models and indexing strategies. Moreover we propose different stemming algorithms in different experiments depending on the language for which we run the experiment. Accordingly at the beginning of each section we provide some details on the architecture of the corresponding experiment.

5.2 Cultural Heritage

Cultural heritage can be outlined as any tangible feature (e.g., hand-crafted substance, built or natural environments) or any intangible feature (e.g., music, dances, traditions, languages) which is reserved from the past. The developing use of digital information challenges the cultural heritage organizations to provide cultural heritage collections in electronic format. The data may come from different sources (libraries, archives, museums, audiovisual archives, books, journals, etc.) in various languages and formats. In order to bring the utmost utility to their users, these digital libraries

should not only be accessible but also easily consultable. To do so they should be properly managed and assessed once they are created. As yet no proper evaluation approaches are available and there is work to be done in this area. Accordingly our aim in this study is to investigate the possibilities to improve the retrieval effectiveness when searching such information systems.

5.3 Challenges

Searching for pertinent cultural heritage objects in response to a short user's query is a challenging task for various reasons. First, in the collection the provided descriptions of the cultural heritage objects are rather short. For example in the English corpus there are 35 indexing terms per record in average (Table 5.5). Moreover the descriptions are rather broad and are produced by different content providers having different indexing policies (for example for an object which is an image of Calliope (muse of epic poetry in Greek mythology), the only provided descriptions are: *Goddess*, *Greek mythology* and *Color aquatint*). In addition, the described objects may originate from different media such as text, image, photo, video, music or sound. Therefore a direct comparison between these descriptors is not really possible. Of course, facing with short item descriptions and short query formulations is not frequent but we can find them in other IR domains (Metzler et al., 2007; Sahami & Heilman, 2006).

The cultural heritage domain is also characterized by a frequent use of names such as personal names (e.g., Picasso), works (e.g., Mona Lisa) as well as geographical entities (e.g., Paris) and temporal references (e.g., Baroque). Moreover it is known that users searching for cultural heritage objects frequently tend to use names in their queries. We must however recall the challenging fact that some proper names may change between languages (e.g., London, Londres) while some others are relatively stable (e.g., Paris). For some cases the spelling variation could be limited between different languages (e.g., Oskar, Oscar).

Another factor that makes our task complex is the multilingual nature of the cultural heritage objects descriptors and topics. For each object, the given description

is available in at least one language, and for many of them, a passage is available in a second language (for example descriptions in French and Dutch for a CH object described by a Belgian source). However, no single language (e.g., English) covers all available records. The user's information needs are also given in various languages but only one must be selected to perform the search. This additional constraint can also be found in the commercial world as, for example, when users are searching for applications for their iPhone (or iPad). In this case, the users are coming from different linguistic backgrounds, express their needs with one or two terms to retrieve an item described by a few keywords or noun phrases. As another challenging issue we can mention the spelling errors. As the topics are extracted from Europeana query logs we can sometimes find typographical errors (e.g., "jean-jaques rousseau" with a spelling error in "Jacques").

Finally, even though our collection is considered as a domain-specific collection but there is a difference between searching this data and a usual domain-specific retrieval. In this search the collection is dedicated to cultural heritage objects but the users are not only the specialists of this domain. Besides, the users do not form a homogeneous group but are coming from different perspectives. We can find students, educators, tourists or "informed citizens". Thus we are dealing with a domain-specific collection with its specific terminology searched by various users who do not necessarily use a specific terminology in their queries which makes the matching process more difficult. This aspect is therefore different from newspaper corpora searched by journalists or patent collection searched by experts.

5.4 Test-Collections

The two main datasets which we use in the experiments are the test-collections which were made available for the CHiC 2012 pilot evaluation lab and CHiC 2013 (Petras et al., 2012) lab at CLEF 2012 and CLEF 2013 evaluation campaigns. In domain-specific IR more researches have been done on patent or medical retrieval rather than on CH domain. Hence the CHiC pilot lab at CLEF 2012 conference started in order to evaluate IR systems for the domain of cultural heritage. The aim of the lab is to provide a standardized and large-scale evaluation of this domain. The data are

extracted from Europeana (www.europeana.eu). Europeana portal is an interface to a digitized collection of Europe's cultural and scientific heritage. It provides access over 23 million objects such as books, paintings, films, museum objects, etc. collected from more than 2200 institutions in 33 countries. Besides providing access to multimedia CH objects, Europeana can be searched using multiple languages. Basically the objects' descriptions correspond to images but we can also find text as well as audio and video. It approximately includes 62% of image, 35% of text, 2% of audio data and 1% of video recordings. Europeana collection is cross-domain and in multiple languages. Europeana is not designated for specific users (e.g., cultural heritage specialists) but it provides all general users with the possibility of exploring its contents.

```
<ims:metadata
ims:identifier=
"http://www.europeana.eu/resolve/record/09405b/8B24F80B16841350BAB1EC58A926259882E23338"
ims:namespace="http://www.europeana.eu/" ims:language="eng">
  <ims:fields>
    <dc:creator>Quintus Caecilius Metellus, moneyer</dc:creator>
    <dc:format>text/html</dc:format>
    <dc:identifier>http://www.fitzmuseum.cam.ac.uk/opacdirect/114568.html </dc:identifier>
    <dc:language>en-GB</dc:language>
    <dc:publisher>The Fitzwilliam Museum, Cambridge, UK</dc:publisher>
    <dc:source>Fitzwilliam Museum</dc:source>
    <dc:subject>coin, semis, Roman Republic</dc:subject>
    <dc:subject>coin</dc:subject>
    <dc:subject>Quintus Caecilius Metellus</dc:subject>
    <dc:subject>semis</dc:subject>
    <dc:title>coin, semis, Roman Republic</dc:title>
    <dcterms:isPartOf>Fitzwilliam Museum</dcterms:isPartOf>
    <dcterms:provenance>bequeathed by Young,ArthurW.,1936-07-07[CM.YG.535-R]</dcterms:provenance>
    <europeana:country>united kingdom</europeana:country>
    <europeana:isShownAt>http://www.fitzmuseum.cam.ac.uk/opacdirect/114568.ht</europeana:isShownAt>
    <europeana:language>en</europeana:language>
    <europeana:object>http://www.peoplesnetwork.gov.uk/dpp/resource/2512018/stream/thumbnail_image_jpeg
    </europeana:object>
    <europeana:provider>CultureGrid</europeana:provider>
    <europeana:type>IMAGE</europeana:type>
    <europeana:uri>http://www.europeana.eu/resolve/record/09405b/8B24F80B16841350BAB1EC58A92625988
    2E23338</europeana:uri>
  </ims:fields>
</ims:metadata>
```

Figure 5.1 Sample of an English record (image of a Roman coin)

The original Europeana index contains several different fields but in the provided collection many of these fields are removed and the documents metadata is mapped to a single XML format. Each cultural heritage object is mainly described by a set of metadata tags providing brief descriptions of the objects (title, keywords, description,

date, provider, etc.) (Petras et al., 2012). However all documents do not have identical tags. The number of tags varies in different documents widely. Some documents contain many different tags whereas fewer can be detected in some others, leading to more sparse content in the latter. This leaves us with short documents, on the average. A sample record of the English collection is shown in Figure 5.1.

In the following sections we first explain the monolingual corpora which cover English, French, German and Polish languages. Afterwards we give some details on the multilingual collection that we use to conduct our experiments.

5.4.1 Monolingual Corpus

The first corpus that we use for monolingual and bilingual retrievals is offered in 3 major European languages, namely English (EN), French (FR) and German (DE). The English corpus consists of 1,107,176 documents; the French one has 3,635,388 ones while there are 3,865,680 documents in the German collection. As mentioned before we can find the objects' descriptions in different media types (image, text audio, video, etc.). Table 5.1 shows the number of documents in each format for the three above-mentioned languages. Nevertheless as far as the experiments in this study are concerned, only human-readable informative texts are of use.

Table 5.1 Documents in the monolingual corpus by language and media type

Language	Sound	Text	Image	Video	Total
German	23,370	664,816	3,169,122	8,372	**3,865,680**
French	13,051	1,080,176	2,439,767	102,394	**3,635,388**
English	5,169	45,821	1,049,622	6,564	**1,107,176**
Polish	230	975,818	117,075	582	**1,093,705**

In the collection there are 50 very short topics (The mean topic size for English topics is less than two terms per topic (~1.8)). Table 5.2 provides some more statistics on the topic lists of each of the languages. These topics are mostly named entities (e.g., people, geographical name, work titles) with, in some cases, indication of a time period. The topics are extracted from Europeana queries logs. Thus they convey the real users' information needs in a cultural heritage search context. Relevant

documents could not however be found for each topic in each language. Among the 50 German topics, 2 have no relevant documents in the collection. This number grows to 11 for the French topics and 14 for English ones. A sample topic from each language is shown in Figure 5.2. As shown in the sample below each topic consists of a title and, sometimes, a description of the content.

```
<topic lang="en">
  <identifier>CHIC-006</identifier>
  <title>esperanto</title>
  <description>Constructed international auxiliary language</description>
 </topic>

<topic lang="fr">
  <identifier>CHIC-004</identifier>
  <title>film muet</title>
  <description />
 </topic>

  <topic lang="de">
   <identifier>CHIC-025</identifier>
   <title> amerikanische sklaverei </title>
   < description />
  </topic>
```

Figure 5.2 Sample of English, French and German topics

Table 5.2 Statistics on the number of distinct indexing terms per topic

	English	French	German	Polish
Mean	1.8	2.24	1.8	2.6
Std dev.	0.60	0.90	0.63	1.16
Median	2	2	2	2
Max	4	5	4	6
Min	1	1	1	1
Topics without rel. items	14	11	2	4

The other monolingual corpus which is used in this study is the Polish corpus. The Polish test-collection is composed of 1,093,705 documents among which 230 documents are audio documents, 975,818 are text, 117,075 are images and 582 video documents. Documents format for Polish is the same as for the previously mentioned languages (as shown in Figure 5.1). Each document, describing one of Europeana's objects includes meta-data in regard to different schema:

- Dublin Core (tags starting with dc: prefix)
- Qualified Dublin Core (tags starting with dcterms: prefix)
- Europeana Semantic Elements (tags with europeana: prefix)

A set of fifty test topics comes with this collection. The set consists of a mixture of topical and name entity queries. We can precisely divide these topics into the following subsets:

1. Chronological topics:
   - 8 topics with explicit time frames (18th or 19th century)
   - 8 topics concerning particular historical period, e.g. *Barok* (Baroque)
2. Name entities:
   - 12 topics with personal names, e.g. *generał Józef Bem* (general Josef Bem)
   - 6 topics with geographical names, e.g. *Kraków* (Cracow)
   - 5 topics of historical names, e.g. *Powstanie Styczniowe* (January Uprising)
3. General entities:
   - 5 topics concerning religion or beliefs, e.g. *diabeł* (devil)
   - 7 topics concerning social groups or functions, e.g. *robotnicy* (workers)

Like in the previous collections the topics are short (in average 2.6 tokens per topic) and they tend to reflect the information needs of Europeana's real users. In the Polish collection we cannot find relevant items for every topic. Topic #17 with 5 relevant objects in the collection ("Czeslaw Milosz" or "Czesław Miłosz") has the minimum number of relevant objects and we find 562 pertinent items for Topic #20 ("PRL (People's Republic of Poland)") that makes this topic the one with the maximum number of relevant objects. In mean, we can find 170.6 relevant objects per topic (median: 125; stdev: 139.6). Statistics on Polish corpus is given in Table 5.17.

5.4.2 Multilingual Corpus

The multilingual collection is composed of 23,300,932 CH object descriptions. The collection with the size of 132 GB consists of records written in the German, French, Polish, Swedish, Italian, Spanish, Norwegian, Dutch and English languages. With fewer objects, we can add the Finnish, Slovenian, Greek, and Hungarian languages, which sum up to 13 different languages. For each object, the given description is available in at least one language, and for many of them, a passage is available in a second language. Table shows the number of documents in each format each of the 13 languages in the collection. Documents format is also the same as in the other collections (as shown in Figure 5.1).

Table 5.3 Documents in the multilingual corpus by language and media type.

Language	Sound	Text	Image	Video	Total
German	23,370	664,816	3,169,122	8,372	**3,865,680**
French	13,051	1,080,176	2,439,767	102,394	**3,635,388**
Swedish	1	1,029,834	1,329,593	622	**2,360,050**
Italian	21,056	85,644	1,991,227	22,132	**2,120,059**
Spanish	1,036	1,741,837	208,061	2,190	**1,953,124**
Norwegian	14,576	207,442	1,335,247	555	**1,557,820**
Dutch	324	60,705	1,187,256	2,742	**1,251,027**
English	5,169	45,821	1,049,622	6,564	**1,107,176**
Polish	230	975,818	117,075	582	**1,093,705**
Finnish	473	653,427	145,703	699	**800,302**
Slovenian	112	195,871	50,248	721	**246,952**
Greek	0	127,369	67,546	2,456	**197,371**
Hungarian	34	14,134	107,603	0	**121,771**
Others	375,730	1,488,687	1,106,220	19,870	**2,990,507**
Total	**455,162**	**8,371,581**	**14,304,289**	**169,899**	**23,300,932**

In the multilingual collection the topic descriptions consist of a mixture of topical and named-entity queries. Information on topic size is given in Table 5.4. The 50 short topics (e.g., “horse couriers”, “Columbus ships”), as mentioned before, tend to reflect information needs as expressed by real Europeana users. The same as in the other collections some topics descriptions contain personal names (e.g., “Marie Sklodowska-Curie”), but we also have topics with geographical names (e.g., “falkland islands”, “rock of Gibraltar”) or with historical names (e.g., “uprisings in 18th century”). In our multilingual experiments we use the new topics which were prepared for the 2013 edition of the data. In CHiC 2012 version of data the topics were extracted only from the Europeana query logs and in some cases there were zero results. The new version of the topics was tested in all languages and so resulted in fewer zero relevant results (Petras et al., 2013). For example for French and German topics of this version we can find relevant documents for all the topics (compare to the 2012 version were we had no relevant results for 11 topics in French and for 2 in German).

Table 5.4 Distinct indexing terms per topic

	Mean	**Std dev.**	**Median**	**Max**	**Min**
German	1.98	0.85	2	4	1
French	2.62	0.93	2	5	1
Swedish	2.3	0.9	2	4	1
Italian	2.12	0.73	2	4	1
Spanish	2.38	0.89	2	6	1
Norwegian	2.12	0.99	2	5	1
Dutch	1.82	0.79	2	5	1
English	2.16	0.57	2	4	1
Polish	2.6	1.16	2	6	1
Finnish	2.24	0.97	2	6	1
Slovenian	3.12	1.05	3	6	1
Greek	2.2	0.52	2	4	2
Hungarian	3.44	1.44	3	9	1

For the English topics only one topic (#64 “Crockery doll houses”) remains with no relevant results compare to 14 topics in last year’s topic lists. Finish corpus with zero relevant documents for 34 topics has the biggest number of topics with no relevant documents. It is followed by Slovenian with 13, Greek with 10, Norwegian and Swedish with 7, Polish and Spanish with 4, Italian with 3 and Hungarian and Dutch with 2. Moreover, the number of relevant documents per topic varies greatly. Topic #53 (“Postage stamp”) has the largest number of relevant items (1,390) while Topic #91 (“Columbus ships”) has the smallest number of relevant documents (19). In mean, we can find 56.7 relevant CH objects per topic (median: 302; stdev: 323).

5.4.3 Relevance Assessment

As explained in Section 3.4.4 considering the big number of documents in large collections, it is not possible to check all the items for relevance. Accordingly for the collections used in this study a pooling technique is used for relevance assessment.

To produce the pool for English, French and German monolingual collections, the 100 top ranked documents from each result list of different systems (submitted by the participants of the lab) were selected. Afterwards for each query the documents were analyzed for the relevance by eight assessors. The documents were marked as “relevant” if it fulfills the information need and “not relevant” if not. The documents could also define as “Europeana relevant”. In this case the document is relevant only as it is represented in the Europeana but not as it was presented in the provided collection. For example some of the objects in Europeana contain thumbnails of the object which are not present in the collection but the assessors could use them for the assessment. For the final evaluation the documents defined as Europeana relevant and not relevant were considered as not relevant and the rest as relevant (Petras et al., 2012).

For the multilingual collection for each language depending on the number of documents different pool depths were chosen. For these records the native speaking assessors for each language (except for English) marked the documents as “highly relevant”, “partially relevant” and “not relevant”. At the end the records defined as

highly and partially relevant were considered as relevant and the remaining as not relevant (Petras et al., 2013).

5.5 Monolingual Retrieval

In our IR group as one of the main tasks we work on design, implementation and evaluation of various indexing and search strategies for a set of different natural languages. Up to this point we achieved to provide groundwork for evaluation and comparison of different tools for monolingual IR, in different languages, using generic test-collections (e.g., newspaper articles). Now our objective is to evaluate different tools considering only a specific field of knowledge in order to integrate domain-specific search into our system. The aim here is to find proper indexing strategies and IR models and to be able to evaluate the impact of document structure and query formulation on retrieval effectiveness. With these finding we will be able to study afterwards the possibilities to improve the search quality in a domain-specific search.

In the monolingual retrieval we use the English, French and the Polish corpora in order to conduct our experiments. Our first objective is to propose and evaluate various indexing and search strategies for these languages when dealing with a corpus containing documents with a specific content. The main goal here is to compare the retrieval effectiveness across different IR models. Our second objective is to measure the relative merit of various stemming strategies when used for monolingual retrieval for the above mentioned languages in the cultural heritage context. In the following sections we will first talk about the experiments on the French and English languages and then we discuss our experiment on the Polish language.

5.5.1 English and French

5.5.1.1 IR Models and Indexing Strategies

As explained before each cultural object in the collection is described by a short list of keywords, usually extracted from a predefined authoritative list. During the indexing process, as mentioned before, we extract only the textual data. We consider the following tags as useful to extract pertinent indexing terms: <dc:contributor>, <dc:creator>, <dc:date>, <dc:language>, <dc:title>, <dc:type>, <dc:subject>, <dc:description>, <dcterms:alternative>, <dcterms:created>, <europeana:country>, <europeana:language>, <europeana:type>, <europeana:year>. For both English and French monolingual retrievals, we apply a stopword removal along with a light stemmer. Our stopword list for English contains 571 terms while the French one has 464 terms. These tools are freely available at members.unine.ch/jacques.savoy/clef/. These lists are composed of terms having a high frequency such as determinants, prepositions, conjunctions, pronouns, and some verbal forms which convey no important meaning. In Table 5.5 we provide some information on the number of indexing terms after this preprocessing phase.

The light stemmer that we use for English removes only the plural '-s' and is called S-Stemmer (Harman, 1991). The stemmer for French removes the inflectional suffixes from plural and feminine forms of the words (Savoy, 1999). Our choice of these light stemmers is based on previous experiments which show that light stemmers tend to be as effective as stemmers based on morphological analysis (Savoy, 2006; Harman, 1991; Fautsch & Savoy, 2009). Moreover applying stemming would not be a good manner to achieve high precision which is the aim in this experiment (Savoy & Rasolofo, 2003).

Table 5.5 Statistics on English, French & German corpora

	English	**French**	**German**
No. of documents	1,107,176	3,635,388	3,865,680
No. of empty docs	456	620	3104
Indexing terms per document			
Mean	35.16	22.78	24.99
Std dev.	40.23	41.03	24.54
Median	25	17	20
Max	1508	3697	2069
Distinct indexing terms per document			
Mean	26.52	17.89	18.79
Std dev.	27.60	18.93	16.01
Median	19	15	15
Max	770	1162	819

In our experiments for the English and French languages we try different weighting schemes in order to compare them and define the most effective ones in terms of achieving a high precision. First we pick the *dtu-dtn* model (Singhal, 2002) as an effective vector-space model. Second, as probabilistic models, we use the Okapi (BM25) (Robertson et al., 2000). Then we try three other probabilistic models extracted from the *Divergence from Randomness* (DFR) family (Amati & Van Rijsbergen, 2002), namely DFR-PL2, DFR-I(n_e)C2, and DFR-I(n_e)B2. The indexing weight (weight of term tj in document d_i) in these models is computed as shown in Table 5.6.

Table 5.6 Formulas used in different models for assigning indexing weight

Okapi	$w_{ij} = \frac{((k_1+1)\cdot tf_{ij})}{(k+tf_{ij})}$ $K = k_1 \cdot \left[(1-b)+b\cdot\frac{l_i}{avdl}\right]$ l_i is the length of document d_i and *avdl* is the average document length.
dtu-dtn	Indexing weight for document terms (*dtu*): $w_{ij} = \frac{(1+\ln(1+\ln(tf_{ij})))\cdot idf_j}{(1-slope)\cdot pivot + slope\cdot nt_i}$ Indexing weight for query terms (*dtn*): $w_{ij} = (1+\ln(1+\ln(tf_{ij})))\cdot idf_j$
DFR	$w_{ij} = Inf_{ij}^1 \cdot Inf_{ij}^2 = -\log_2\left(Prob_{ij}^1(tf_{ij})\right)\cdot\left(1-Prob_{ij}^2(tf_{ij})\right)$ **DFR- I(n_e)C2:** $Inf_{ij}^1 = tfn_{ij}\cdot\log\left[\frac{n+1}{n_e+0.5}\right]$ $Prob_{ij}^2 = 1-\frac{tc_j+1}{df_j\cdot(tfn_{ij}+1)}$ **DFR-I(n_e)B2:** $Inf_{ij}^1 = tfn_{ij}\cdot\log_2[(n+1)/(n_e+0.5)]$ $Prob_{ij}^2 = 1-\left[(tc_j+1)/(df_j\cdot(tfn_{ij}+1))\right]$ **DFR-PL2:** $Prob_{ij}^1 = \frac{e^{-\lambda_j}\cdot\lambda_j^{tfn_{ij}}}{tfn_{ij}!}$ $Prob_{ij}^2 = \frac{tfn_{ij}}{tfn_{ij}+1}$ tf_n is the normalized term frequency. $\lambda = \frac{tc_i}{n}$ (tc_j is the number of occurrence of term t_j in the collection and n is the size of the elite set) $n_e = n\cdot\left(1-\left(\frac{n-1}{n}\right)^{tc_j}\right)$ $tfn_{ij} = tf_{ij}\cdot\log_2\left(1+\frac{c\cdot mean\_dl}{l_i}\right)$($c$ and *mean_dl* (average document length)

5.5.1.1.1 Results and Discussions

Tables 5.7 and 5.8 show the Mean Average Precision (MAP) for, respectively, English and French corpora used for the monolingual retrieval. For both languages, we tried different IR models while applying a light stemmer (LStem) (see previous section) and compared these results with the ones obtained when stemming is ignored. In using the Okapi model the avdl (average document length) is set to 181 for English corpus and 169 for the French one, the constant k_1 to 1.2, for both languages, and we tried three different values for the constant *b*: 0.5, 0.7 & 0.9.

Table 5.7 MAP of different IR models, English corpus

	DFR I(n_e)C2	**DFR I(n_e)B2**	**DFR PL2**	**Okapi (*b*=0.5)**	**Okapi (*b*=0.7)**	**Okapi (*b*=0.9)**	***dtu-dtn***	**Avg.**
NoStem.	0.4244	0.4524	0.4354	0.4289	0.4207	0.4032	0.4320	0.4281
S-Stem.	0.4487	**0.4752**	0.4628	0.4560	0.4429	0.4229	0.4484	0.4510
% Change	+5.7%	+5.0%	+6.3%	+6.3%	+5.3%	+4.9%	+3.8%	+5.3%

Table 5.8 MAP of different IR models, French corpus

	DFR I(n_e)C2	**DFR I(n_e)B2**	**DFR PL2**	**Okapi (*b*=0.5)**	**Okapi (*b*=0.7)**	**Okapi (*b*=0.9)**	***dtu-dtn***	**Avg.**
NoStem.	0.3520	0.3582	0.3623	**0.3627**	0.3602	0.3497	0.3413	0.3552
LStem.	0.3290	0.3360	0.3392	0.3402	0.3348	0.3253	0.3197	0.3320
% Change	-6.6%	-6.2%	-6.4%	-6.2%	-7.1%	-7.0%	-6.3%	-6.5%

As the results show, for the English corpus, with DFR-I(n_e)B2 model we achieve the highest MAP while the best performing model for French is Okapi model (with *b*=0.5). The results show that applying the light stemmer for the English language improves the effectiveness of the search which is not the case for the French collection. As can be seen in Table 5.8 we achieve higher MAP while ignoring the stemming phase for the French language. By making a query-by-query analysis on the results we can find some examples where stemming misleads the retrieval. In Topic #21 the title “chardonne” (Jacques Chardonne, Writer (F.) Or place in Switzerland) is indexed as “chardon” (after applying the light stemmer) which leads the system to retrieve in its top ranks non-relevant documents (in which “chardon” refers to a flower) such as:

- Etude de feuilles de echirops, de sphoerophalus, chardon cultivé, de chardon sauvage de la mer, de fleur lilas, de chardon sauvage
- Sujet ou décor : représentation végétale (fleur, chardon) ; chardon bleu ; Etude de chardon fleuri
- Chardons sur la côte rocheuse

As another example we can mention Topic #9 for which the title "îles malouines" changes to "malouin" after stemming and results in the retrieval of non-relevant documents (where "Malouin" is a proper name) such as follows in the top ranks:

- L'Avare, comédie de Molière en 5 actes, mise en vers, par A. Malouin
- Villas de la Malouine

5.5.1.2 Data Fusion

In our experiment we want to see whether combining different indexing schemes and IR models improves the retrieval effectiveness, as it is supposed to, or not (Vogt & Cottrell, 1999). It is probable that different strategies retrieve the same relevant items in their top ranks rather than the same non-relevant ones. Therefore we consider that by combining different ranked lists, resulting from different IR models, we will gain a list with relevant documents in higher ranks and the non-relevant items in lower ones. In order to produce this combination of ranked lists, different fusion operators can be used. In our study we choose the Z-score scheme which tends to perform the best (Savoy, 2005; Dolamic et al., 2009). More details about the Z-score strategy can be found in (Savoy & Berger, 2005).

5.5.1.2.1 Results and Discussions

In Table 5.9 we can see the results for our data fusion approach for the English corpus. We applied the fusion operators on the results when using the S-Stemmer. We can see that the MAP obtained by combining different result lists enhances slightly the performance. However the difference between the MAP obtained for each model separately and the combined one is rather small.

Table 5.9 MAP of different combinations of IR models, English corpus

Model	Query Expansion (*idf*-based)	Single MAP	Combined MAP Z-Score
DFR-I(n_e)B2 **DFR-PL2**		0.4752 0.4628	0.4715
DFR-I(n_e)B2 **DFR-I(n_e)C2**	5 documents /10 terms	0.4752 0.3918	0.4611
DFR-I(n_e)B2 ***dtu-dtn***		0.4752 0.4484	**0.4758**
dtu-dtn **DFR-PL2**		0.4484 0.4628	**0.4667**
DFR-I(n_e)C2 ***dtu-dtn***		0.4487 0.4484	**0.4518**
DFR-I(n_e)B2 **Okapi(*b*=0.9)**	20 documents /10terms	0.4338 0.4229	**0.4378**
dtu-dtn **DFR-PL2**	5 documents /10terms	0.4484 0.3834	0.4301
DFR-I(n_e)C2 ***dtu-dtn*** **Okapi(*b*=0.9)**	20 documents /10terms 10 documents /10terms	0.4074 0.3677 0.4229	**0.4238**
DFR-I(n_e)C2 ***dtu-dtn*** **Okapi(*b*=0.9)**	20 documents /10terms 10 documents /30terms	0.4074 0.3376 0.4229	0.4171

5.5.1.3 Pseudo-Relevance Feedback

Pseudo-relevance feedback (PRF) or blind-query expansion is considered to be often an effective method for query expansion. Our previous experiments on other corpora, based on newspaper articles, show that this method tends to improve the retrieval effectiveness (improve the MAP of around 5% to 30%) (Akasereh & Savoy, 2013). It is particularly attractive because it does not need any external input. PRF considers the top-ranked documents initially retrieved by the system as relevant ones and uses them as implicit feedback to generate additional terms. In some implementations, we can also consider non-relevant items (e.g., the documents displayed at the bottom of the returned list). We should take this into consideration that some of the top-ranked

documents might be non-relevant which makes them noisy feedbacks (Shokouhi et al., 2009; Xu et al., 2009; Cao et al., 2008).

In our experiments we first apply the Rocchio's approach (Buckley et al., 1996) with $\alpha = 0.75$, $\beta = 0.75$. Here the system expands the initial query by adding the most frequent m terms selected from the k best ranked documents retrieved for the original query. In some cases adding frequently occurring terms produces noise (Peat & Willett, 1991) therefore we also apply an *idf*-based query expansion (Abdou & Savoy, 2008) as a second PRF approach. The reason for trying both approaches is that in some cases adding frequently occurring terms produces noise and consequently Rocchio's approach does not give good results (Peat & Willett, 1991). We employ both methods to different number of documents from which was extracted different number of terms.

5.5.1.3.1 Results and Discussions

Table 5.10 contains the MAP obtained when applying pseudo-relevance feedback. These results reveal that in this experiment the PRF technic did not help to enhance the retrieval performance. The reason is due to the fact that in this experiment we are dealing with relatively short documents (having the average number of distinct indexing terms per document at ~54 for English and ~56 for French).

Table 5.10 MAP of *idf*-based blind-query expansion, English and French queries

		English, DFR_I(n_e)B2 S-Stemmer		**French Okapi NoStem**	
MAP without PRF		0.4752		0.3627	
No. of Documents	**No. of Terms**	***idf*-based**	**Rocchio**	***idf*-based**	**Rocchio**
5	5	0.4382	0.3576	0.3488	0.3682
	10	0.4315	0.3787	0.3483	0.3638
	30	0.3864	0.3440	0.3428	0.3667
	50	0.3656	0.3280	0.3241	0.3645
	70	0.3606	0.3210	0.3110	0.3659
10	5	**0.4557**	0.3348	0.3432	0.3724
	10	0.4250	0.3528	0.3472	0.3738
	30	0.3923	0.3355	0.3300	0.3431
	50	0.3875	0.3346	0.3283	0.3412
	70	0.3913	0.3312	0.3272	0.3401
15	5	0.4545	0.3410	0.3329	0.3717
	10	0.4432	0.3495	0.3166	0.3744
	30	0.3981	0.3339	0.2971	0.3437
	50	0.3878	0.3261	0.2947	0.3442
	70	0.3764	0.3225	0.2916	0.3415
20	5	0.4519	0.3381	0.3404	0.3660
	10	0.4338	0.3207	0.3181	**0.3746**
	30	0.3962	0.3181	0.2900	0.3481
	50	0.3850	0.3152	0.2864	0.3471
	70	0.3798	0.3072	0.2876	0.3412
25	5	0.4456	0.3388	0.3439	0.3649
	10	0.4346	0.3164	0.3231	0.3696
	30	0.3901	0.3243	0.3031	0.3500
	50	0.3789	0.3187	0.2641	0.3486
	70	0.3723	0.3064	0.2608	0.3401

5.5.2 Polish Language

Polish language is a Slavic language with a relatively complex morphology. In our experiments we first evaluate the impact of language independent indexing strategies for this language. Afterwards we repeat our experiments adding a stemming phase to

them in order to evaluate the influence of stemming for this language and compare it with other languages from the same family (e.g., Czech).

5.5.2.1 IR models and Indexing Strategies

In the Polish collection each CH object descriptor is in average composed of around 50 distinct indexing terms. From the various tags in the documents we extracted the following for indexing procedures: <dc:contributor>, <dc:creator>, <dc:description>, <dc:date>, <dc:language>, <dc:subject>, <dc:title>, <dc:type>, <dcterms:alternative>, <dcterms:created>, <europeana:language>, <europeana:type>, <europeana:uri>, <europeana:year>. Some statistics on documents' lengths after processing the collection are given in Table 5.11.

Table 5.11 Statistics on Polish corpus

No. of documents		**No. of empty docs**	
1,093,705		2109	
Number of indexing terms per document			
Mean	**Std dev.**	**Median**	**Max**
72.06	86.39	43	2095
Number of distinct indexing terms per document			
Mean	**Std dev.**	**Median**	**Max**
50.28	47.31	35	1040

For this language we suggest a stopword list consists of 138 words (mainly determiners, prepositions, conjunctions, pronouns and auxiliary verbal forms). For the Polish language as a first indexing strategy, we investigate different text representations based on *n*-gram (McNamee & Mayfield, 2004), as well as trunc-*n*. In *n*-grams approach, as explained in Section 3.1.3, we produce overlapping sequences of *n* characters for each word. While trunc-*n*, is the process of truncating a word by keeping its first *n* characters and cutting of the remaining letters. Such representations usually tend to form good overall baselines when facing with a new language (for which no good stemmer is available or known). The benefit sought of implementing *n*-gram or truncation is to assign low indexing weights to frequent suffixes usually added to indicate grammatical cases, gender modifications, or derivational suffixes. In fact, the Polish language has seven grammatical cases, three genders, and two

numbers, and the corresponding suffixes are attached to both nouns (four possible declensions) and adjectives. As a second indexing strategy we opt the whole words with or without applying a light stemmer (Savoy, 2006). This word normalization is based on a set of grammatical rules trying to remove only inflectional suffixes from nouns and adjectives. For the Czech language (a language of similar morphology as Polish), applying a stemming stage improves the retrieval effectiveness of around 40% (Dolamic & Savoy, 2009). For other languages having a complex morphology, a simple algorithmic stemmer does not provide the expected improvement (Korenius et al., 2004); this is mainly due to numerous exceptions or spelling irregularities.

As IR models, we first consider the classical *tf idf* (with cosine normalization) (Manning et al., 2008). This approach was selected only to provide a baseline. As other effective IR models, we use the Okapi (or BM25) (Robertson et al., 2000), and the DFR-I(n_e)B2 as one implementation of the DFR probabilistic paradigm (Amati & Van Rijsbergen, 2002).

5.5.2.1.1 Results and Discussions

Tables 5.12 and 5.13 show our results for the Polish language. In Table 5.12, we have evaluated different sub-word indexing strategies, showing that the trunc-*n* tends to produce better retrieval effectiveness. Moreover, the value of the parameter *n* must be larger than with the French of English languages, with the best value being equal to 6.

Table 5.12 Result MAP based on *n*-gram or trunc-*n* approaches

	DFR-I(n_e)B2		**Okapi**	
	***n*-gram**	**Trunc-*n***	***n*-gram**	**Trunc-*n***
$n = 4$	0.2350	0.2268	0.2466	0.2532
$n = 5$	0.2610	0.2968	0.2577	0.3038
$n = 6$	0.2611	0.3078	0.2640	**0.3211**

Table 5.13, shows the evaluation of the classical tf idf and the two probabilistic models on the Polish collection. The performance measure indicates that the Okapi probabilistic model proposes the best performance. Moreover, the use of both a stopword list and a light stemmer clearly tends to improve the overall effectiveness.

Table 5.13 Result MAP based on word-based indexing

IR Model	No Stopword No Stemming	Stopword No Stemming	No Stopword Stemming	Stopword Stemming
tf idf	0.2558	0.2566	0.2541	0.2579
Okapi	0.3060	0.3140	0.3258	**0.3433**
DFR-I(n_e)B2	0.2883	0.3028	0.3085	0.3308

When comparing the MAP values depicted in the second (no stopword, no stemming) and third column (stopword, no stemming), we can see an improvement after removing functional words with the two probabilistic models (e.g., from 0.3060 to 0.3140 (+2.6%) for the Okapi model). Applying a light stemmer clearly improves the retrieval effectiveness of both probabilistic models (from 0.3140 to 0.3433 (+9.3%) for the Okapi model).

Studying the results in details for some queries leads us to the conclusion that a unigram indexing strategy can be improved by applying Boolean conjunction to the searching terms. For example, topic #31 "Lech lub Jarosław Kaczyński" contains personal names. However, there is a Polish town called Jaroslaw and many municipal documents are available in Europeana. Thus for this topic there were numerous false positive retrievals concerning the town and not the person. If we search separately for the terms in this topic we find 3,318 documents pertinent to Lech, 1,049 pertinent to Kaczynski, and 9,253 to both Jaroslaw as personal name and as the town. Using classical *tf idf* or Okapi approaches this topic was of the worst relevance ratio – for the 731 assessed documents only 16 items (2%) were considered relevant or partially relevant. Another conclusion regarding personal names is that for any person the last name should be weighted higher than the first name during the ranking process.

5.5.2.2 Pseudo-Relevance Feedback

As an additional strategy to improve the retrieval effectiveness, we again use pseudo-relevance feedback information (see Section 5.5.1.3) in order to generate a new expanded query.

5.5.2.2.1 Results and Discussions

Table 5.14 shows the results of PRF approach with different parameters (different number of documents and terms).

Table 5.14 Result MAP based on Rocchio pseudo-relevance feedback

IR Models Parameters	**DFR-I(n_e) 5-grams Rocchio**	**DFR-I(n_e)B2 Word-based No Stem Rocchio**	**DFR-I(n_e)B2 Word-based No Stem *idf*-based**
Without PRF	0.2610	0.3028	0.3028
5 docs, 5 terms	0.1572	0.2189	0.2784
5 docs, 10 terms	0.1590	0.2119	0.2780
5 docs, 20 terms	0.1552	0.2013	0.2777

We evaluate the DFR-I(ne)B2 search model with different parameter values. As we can see, this search technique tends to hurt the MAP achieved by the original query, using the word-based or 5-grams indexing scheme, Rocchio or idf-based selection schemes. Adding automatically terms in the query is clearly not a useful method in our context.

5.5.3 Conclusion

The results obtained in the English and French monolingual retrievals state that the models derived from the Divergence from Randomness (DFR) family yield the best retrieval effectiveness regardless the underlying language and test-collection. Applying DFR-I(n_e)B2 and DFR-PL2 for both the French and English corpora produced a high MAP compared to other tested models. Our results reveal that the Okapi model (with b=0.5) tends also to be an effective model. The resulting question is to define the best values for the underlying constants.

Our experiment shows that applying a light stemmer (removing only the plural '-s') for English, helps to achieve better results than when the stemming phase is skipped. On the contrary, when using our light stemmer for French (removing plural and feminine suffixes) does not seem to enhance the retrieval performance. A simpler

stemmer for the French language may produce a better effectiveness than the applied light stemmer.

Considering the results from all monolingual experiments (English, French and Polish), we can also conclude that when dealing with relatively short documents, blind-query expansion is not a useful expansion method in order to improve the retrieval effectiveness. In such cases, it seems difficult to select the most appropriate terms to be included in the expanded query.

For the Polish language, we found that the use of a short stopword list and a light stemmer improves retrieval effectiveness. The use of words as indexing units is better than considering *n*-gram or trunc-*n* indexing schemes. However, we cannot specify whether a more aggressive stemmer (affecting also verbs) or a statistical one may further enhance the performance (Majumder et al., 2007; Paik et al., 2011; Paik et al., 2013). Moreover, the effectiveness of a Polish lemmatizer must also be investigated. For this language we suggested a stopword list of 138 words. However a longer list can be created to achieve a broader coverage of functional words in this language (Fox, 1989).

For this language neither vector-space, nor probabilistic models can impose relevant retrieval of all keywords from the query. Using a semi-Boolean approach as logical conjunction of query terms tends to be a better strategy.

5.6 Bilingual Retrieval

Our objective in this part is to assess the effectiveness of query translation methods in a bilingual retrieval.

5.6.1 Experiment Architecture

In our bilingual retrieval we use the same collection that we used in monolingual experiments. We use the German and French topics to search the English corpus (for details on the corpus see Section 5.4.1). Our approach is based on query translation

(QT). Thus we produce the English translations for German and French topics and then we launch the search on the English corpus. To translate the queries we use two different strategies. First we use Google translation which seems to give reasonable results when dealing with very short query formulation (Dolamic & Savoy, 2009). As a second approach we use the combination of Wikipedia and Google considering that a combination of translation strategies improves the retrieval performance (Savoy & Berger, 2005).

In our tests we use Okapi, different DFR models together with the *dtu-dtn* vector-space model while applying the S-Stemmer as used on English corpus in our monolingual experiments (Section 5.5.1.1).

5.6.2 Results and Discussions

The results for the bilingual retrieval are shown in Tables 5.15 and 5.16. We can see that using the combination of Google and Wikipedia results a better performance.

The topics used in this collection are mostly name entities and only the title is used for the search which makes the translation less critical and easier. As a result there are not many differences between translations produced with the two strategies. However, by inspecting the results in details we can find some cases for which a better translation led to better retrievals. In translating Topic #5 (“briefmarke”), from German to English, Google gives us the word “stamp” versus “postage stamp” which resulted from the Google and Wikipedia combination. As a result the system returns 9 relevant documents among its first 10 ranks when searching “postage stamp” while by searching “stamp” the first relevant document only appears at rank 82. Using the French topics for the same topic (“timbre poste”), Google gives us “stamp post” versus “postage stamp” using the combination method. Here again the system retrieves 9 relevant documents among its first 10 ranks using “postage stamp” while by searching “stamp post” it retrieves 5 relevant documents among its first 10 having the first relevant at rank 5.

Table 5.15 MAP of different IR models, German topics on English corpus

	DFR I(n_e)C2	**DFR I(n_e)B2**	**DFR PL2**	**Okapi (*b*=0.5)**	**Okapi (*b*=0.7)**	**Okapi (*b*=0.9)**	***dtu-dtn***	**Avg.**
Google	0.4181	**0.4462**	0.4309	0.4255	0.4101	0.3910	0.4223	0.4206
Google+ Wikipedia	0.4403	**0.4691**	0.4478	0.4322	0.4144	0.4580	0.4459	0.4440
% Change	+5.3%	+5.1%	+3.9%	+1.6%	+1.0%	+17.1%	+5.6%	+5.6%

Table 5.16 MAP of different IR models, French topics on English corpus

	DFR I(n_e)C2	**DFR I(n_e)B2**	**DFR PL2**	**Okapi (*b*=0.5)**	**Okapi (*b*=0.7)**	**Okapi (*b*=0.9)**	***dtu-dtn***	**Avg.**
Google	0.3960	**0.4214**	0.4053	0.4006	0.3908	0.3705	0.4100	0.3992
Google+ Wikipedia	0.4096	**0.4346**	0.4197	0.4137	0.4051	0.3861	0.4218	0.4129
% Change	+3.5%	+3.1%	+3.6%	+3.3%	+3.7%	+4.2%	+2.9%	+3.4%

5.6.3 Conclusion

Our results from the bilingual search confirm the effectiveness of DFR-I(n_e)B2 model and the S-Stemmer (used for English). Furthermore, they show that a combined translation strategy leads to perform better results than a single one. Even though in our experiment, having very short topics (and mostly name entities), the difference between the various translation methods is not remarkable. Thus for practical reasons we suggest using only one translation device.

5.7 Multilingual Retrieval

In this experiment we face with more than 23 million of CH objects described in 13 different languages with their corresponding topics. In our experiments, we have used the 50 topics written in each language. This corpus forms a real multilingual test-collection and various MLIR strategies can be evaluated (Peters et al., 2012) (for

more details on the corpus see Section 5.4.2). Our experiments explore the problem when facing with short text descriptions expressed in various languages having a richer morphology than English. We use two different approaches to perform our search. As a first approach, we built a single big collection with all CH object descriptions. We then search into this single corpus using the 50 multilingual topics. This first approach must be viewed more as a baseline than a realistic implementation. As the second strategy, we build 13 distinct corpora according to the language in use and associate a dedicated server per language. We then search separately each corpus against its corresponding topics. In a final step, the broker needs to merge the 13 different result lists to generate a single ranked list of retrieved items (see Section 5.7.2).

5.7.1 Setup and Indexing

To index the collection we extract only the tags containing textual information. However, we do not use all the available information. In fact, we remove the tags containing general information on the objects such as the publisher or the provider name. To generate a surrogate for each CH object, we only use the following six tags: <dc:contributor>, <dc:creator>, <dc:description>, <dc:subject>, <dc:date>, <dc:title>. This set of tags contains most of the useful information about each CH object.

As mentioned before the documents are relatively short. Once the collection is parsed, considering all the languages, the minimum of distinct terms per record is 12 for Slovenian or Greek, with a maximum of 50 for the Polish language and with a median of 19. Details on the size of each collection are given in Table 5.17. As for topics, we use only the title section of each topic formulation. Nevertheless, we provide two different sets of topics. First, we use the original topics provided in each language. In a second experiment, we use only the English topics and then we automatically translate them into the other 12 languages. We conduct some of our experiments with these two sets of topics to be able to measure the impact of the automatic query translation process.

Table 5.17 Statistics for each language in the corpora

language	Number of empty documents	Number of documents	Distinct indexing terms per document			
			Mean	Std dev.	Median	Max
German	3104	3,865,680	18.79	16.01	15	819
French	620	3,635,388	17.89	18.93	15	1162
Swedish	1	2,360,050	29.70	44.11	14	1166
Italian	44319	2,120,059	20.63	18.74	16	2156
Spanish	3	1,953,124	23.39	19.46	20	1656
Norwegian	1273	1,557,820	14.83	13.14	12	611
Dutch	46169	1,251,027	11.98	11.88	10	762
English	456	1,107,176	26.52	27.60	19	770
Polish	2109	1,093,705	50.28	47.31	35	1040
Finnish	0	800,302	13.97	7.84	13	500 (min=1)
Slovenian	0	246,952	11.88	11.15	9	445 (min=1)
Greek	23	197,371	12.13	22.45	5	575
Hungarian	0	121,771	35.07	55.30	24	1271 (min=1)

For all languages (except for Slovenian and Greek) we apply a stopword removal (Fox, 1989). These lists differ in size for each language (from the longest composed of 747 Finnish words to 138 Polish terms). Table 5.18 shows the number of terms that each stopword list contains for each of the languages. For each language, such a list contains terms having a relatively high frequency and is composed mainly by determiners, prepositions, conjunctions, pronouns, and some verbal forms (these lists are freely available at members.unine.ch/jacques.savoy/clef/).

Table 5.18 Size of stopword list for each language

Language	Size of the stopword list
German	578
French	464
Swedish	386
Italian	430
Spanish	307
Norwegian	176
Dutch	315
English	571
Polish	138
Finnish	747
Hungarian	737

Considering the frequent use of names as one of the characteristics of the CH domain, we suggest applying a light suffix-stripping stemmer for each language. In this perspective, each algorithmic stemmer is designed according to the grammar rules of the corresponding language. More precisely, these light stemmers try to remove only the inflectional suffixes attached to nouns or adjectives to denote the gender, number, and the different grammatical cases. For example, the English light stemmer removes only the plural suffix "-s" (Harman, 1991). The French light stemmer removes the inflectional suffixes denoting the plural and feminine forms while for languages like Polish or Dutch, more rules were needed. Finally, each suffix removal step is controlled by quantitative and qualitative restrictions to guarantee some consistencies of the resulting terms (Savoy, 2006). We can mention that the performance difference between a light and a more aggressive stemmer is not significant for the English language (Fautsch & Savoy, 2009). As a variant when high precision is the main objective, we also index the CH object descriptions without considering the stemming stage.

5.7.2 IR Models and Data Fusion

As an effective IR model, we choose the Okapi (BM25) (Robertson et al., 2000) as our weighting scheme. As we deal with relatively short documents, we consider that this IR model is well adapted when facing with short textual descriptions (Yuanhua & Zhai, 2011) and would provide a high retrieval effectiveness level (Yuanhua & Zhai, 2011). To define the parameter values, we apply the default setting of the Okapi BM25 with $b = 0.75$ and $k_1 = 1.2$. The same set of values was used for all languages. After this step, we have 13 servers, each corresponding to one language. As soon as they receive the query in their corresponding language, each server produces a ranked list of retrieved CH objects.

In order to merge these result lists produced separately, the broker may apply different merging strategies. As a baseline approach, we merge these lists in a round-robin manner (denoted as "RR"). In this case, we take one document in turn from all individual lists and repeat this process (Fox & Shaw, 1993). As an alternative, we also use a biased round-robin approach (Savoy, 2004). In this case, we assume that each server does not contain the same number of pertinent items for each query. In our implementation, we decide to favor languages having a larger number of items, expecting they will also contain more relevant items. To simplify the process, we take, per round, three documents from German and French result lists, two from the Swedish, Italian, Spanish, and one from the rest of the languages. We denote this biased round-robin approach as "bRR".

As other merging schemes, we take into account the document score (or retrieval status value, RSV) computed for each retrieved item. Accordingly, as third merging strategy, we normalize the document scores within each language (or server). To achieve this, we divide each document score by the maximum score (or the score achieved by the first document in each ranked list). We name this strategy "NormMax". For the *i*th collection, the new RSV' for the *k*th document is $RSV'_k = RSV_k / Max^i$, where Max^i denotes the document score having the maximal value in the *i*th result list. As fourth merging approach, we apply a variant of the previous one, called "MinMax". In this case, we normalize the document score by taking into account not only the maximum score but also the minimum one (Savoy, 2004). More

formally, the new RSV' score is computed as: $RSV'_k = ((RSV_k - Min^i) / (Max^i - Min^i))$.

As the final merging strategy, we apply the Z-score operator to merge the different ranked lists (denoted as "Z-score"). In this case, the document score is normalized by considering the average and the standard deviation of the document scores distribution in each result list (Savoy, 2004). Thus, the new $RSV'_k = ((RSV_k - Mean^i) / Stdev^i) + \delta^i$, with $\delta^i = ((Mean^i - Min^i) / Stdev^i)$ used to obtain always positive value.

5.7.3 Results and Discussion

As a first baseline we form a single collection with all the CH object descriptors. As a search query we concatenate all the 13 original topic titles (forming a multilingual query). We do not apply any stopword list and we ignore the stemming stage. The resulting MAP is rather low with a value of 0.0476.

To improve this result, we index the CH descriptors according to their given language and we applied a language-dependent stopword list. When using this indexing strategy with a single inverted file, we can achieve a MAP of 0.1158. In order to verify the impact of an automatic query translation, we conduct the same experiment but using the translated queries instead of the original ones. To achieve this, from the English topic title, we use the Google translate service to automatically translate the submitted English query into the 12 different languages. Finally, we concatenate all query translations with the original English topic. This approach achieves a MAP of 0.1200.

Table 5.19 provides an overview of the retrieval performance according to each language, using either the translated queries or the original ones. Using the original topic formulations, the achieved MAP is higher, but the retrieval performance differences are usually small. In a related vein, we also compare, for each language, the retrieval effectiveness when applying or not a light stemmer. In some cases, the light stemmer improves the mean performance (e.g., with the English or French language). For other languages, the resulting effect is small and negative (e.g., with

the Swedish, Norwegian, or Spanish languages). Table 5.19 reports the MAP over 50 topics for each language. In the second column, we can find the retrieval effectiveness of the original topics when using a light stemmer. In the third column, we ignore this word normalization procedure. In the fourth column, we use the translated topic titles from the English formulation and performed the search without considering the stemming stage. The last column shows the percentage change when using the translated queries (without stemming) compare to the case that the original queries were used.

Table 5.19 MAP for each language, with original or automatically translated queries, with or without a light stemmer

Language	Light Stem. Original queries	No stem. Original queries	No stem. Translated queries	% Change
German	0.2863	0.2963	0.2846	-3.9%
French	0.2596	0.2359	0.2176	-7.8%
Swedish	0.2054	0.2216	0.1664	-24.9%
Italian	0.2402	0.2584	0.2575	-0.35%
Spanish	0.2558	0.3056	0.3057	+0.03%
Norwegian	0.3511	0.3859	0.2830	-26.7%
Dutch	0.3299	0.3223	0.2599	-19.4%
English	0.3022	0.2490	0.2490	0%
Polish	0.3042	0.3035	0.2120	-30.1%

Overall, this first strategy (making a single collection out of all the corpora) owns the advantage to be rather simple to implement and demonstrates the usefulness of a stopword list.

As a second indexing and search strategy, we divide the Europeana corpus according to the language and formed 13 servers. The topic title of the original formulation was then sent to each server. Separately, each server produces a ranked list of retrieved items. Finally, we need to merge the 13 result lists to generate the final answer presented to the end-user. To evaluate the various steps in this multilingual search process, we first evaluate the quality of the various merging strategies and the usefulness of applying a light stemming strategy. When the

submitted topics tend to contain many names, a light stemming may hurt the overall retrieval effectiveness. For example, the name "Baring" becomes "Bare" when using the Porter stemmer.

Table 5.20 Evaluation of different stemming and merging strategies

Parameter setting	Stemming	Stopwords	Language	MAP
Separate indexes, RR	No	Yes	All	0.1388
Separate indexes, bRR	No	Yes	All	0.1402
Separate indexes, NormMax	No	Yes	All	0.1444
Separate indexes, MinMax	No	Yes	All	0.1516
Separate indexes, Z-score	No	Yes	All	0.1545
Separate indexes, RR	Yes	Yes	All	0.1065
Separate indexes, bRR	Yes	Yes	All	0.1386
Separate indexes, NormMax	Yes	Yes	All	0.1515
Separate indexes, MinMax	Yes	Yes	All	**0.1592**
Separate indexes, Z-score	Yes	Yes	All	0.1396

Table 5.21 Evaluation of different server selection approaches

Parameter setting	Stemming	Stopwords	Language	MAP
Separate indexes, bRR	No	Yes	All	0.1402
Separate indexes, NormMax	No	Yes	All	0.1444
Separate indexes, MinMax	No	Yes	All	0.1516
Separate indexes, Z-score	No	Yes	All	0.1545
Separate, bRR	No	Yes	All–{SL, EL, HU}	0.1389
Separate, NormMax	No	Yes	All–{SL, EL, HU}	0.1604
Separate, MinMax	No	Yes	All–{SL, EL, HU}	**0.1735**
Separate, Z-score	No	Yes	All–{SL, EL, HU}	0.1622

As depicted in Table 5.20, we considered all the 13 languages, with a stopword list adapted for each language and five distinct merging strategies. In this set of experiments, when we ignored the stemming stage, the best result (MAP: 0.1545) is based on the Z-score merging operator. When we apply a light stemming strategy,

the best overall performance is obtained with the MinMax merging operator (MAP: 0.1592).

In Table 5.21, we assume that some languages, owning clearly less records than others, can be ignored during the selection of the most useful servers. More precisely, we have conducted a set of experiments where the Slovenian (SL), Greek (EL), and Hungarian (HU) languages were not searched. As we can see, this arbitrary and prior selection seems to work by allowing better overall retrieval performance than searching into all 13 collections. The best result is achieved by a run based on the MinMax merging operator.

5.7.4 Conclusion

Our experiments were rather complex due to the very short descriptions, written with broad terms and the difficulty of having a precise meaning of the real user's information needs. The complexity of the morphology of the various languages used to describe the cultural heritage objects clearly increased the difficulty of our task.

In the multilingual experiments, we have selected the probabilistic Okapi model the results of our experiments show that producing one single index out of all the collections and search this index has the only advantage of simplicity. On the other hand when we search different corpora separately and merge the results to a final ranked list the performance improves. The performance still depends on the choice of the merging operator. Our results show that Z-score and MinMax merging operators (see Section 5.7.2) perform the best. Of course we should not forget the importance of a good stopword list. Applying such lists has a clear and positive impact on the overall retrieval effectiveness.

The other interesting finding is that when we delete the languages with fewer records from our search the overall performance increases. We can see this already when we apply a biased round-robin approach rather than round-robin (see Section 5.7.2). During the merging process when we favor the languages with more records and select more items per round from their results the performance grows (see Table 5.21).

5.8 Query Expansion and Relevance Feedback

As mentioned in Section 5.3, with our test-collection we are dealing with a domain-specific collection which is provided for all users (experts as well as general users). Therefore some users do not use the specific terminology of the domain or they do not manage to convey their information need to the system with their submitted search terms. Besides as a general trend, we can observe that many short queries are submitted to the cultural heritage search engines. Thus the user's information need is provided with few useful search terms. Accordingly finding an effective query expansion method would be a good solution in order to improve the retrieval effectiveness for cultural heritage information systems. One way to improve the retrieval effectiveness is to consider techniques that automatically enhance the submitted request. Based on these, in our study, we try to evaluate the performance of different PRF approaches and query expansion with external resources in the context of cultural heritage retrieval.

5.8.1 Related Work

In the field of domain-specific IR, several collections are so far made available by different evaluation campaigns and various studies are conducted on them. To give some examples we can name PubMed (a bibliographic collection containing references to articles from journals on life sciences used in genomics TREC), GIRT (a collection in social science field used in CLEF 2000-2008) and CHiC collection (composed of records on cultural heritage domain).

Query expansion in domain-specific search has been evaluated by various studies. Petras (2005) compares the technique of Entry Vocabulary Module (EVM) query expansion with blind feedback on GIRT collection and reports that EVM improves over blind query expansion.

Abdou & Savoy (2008) propose an *idf*-based query expansion. The evaluation of this method on MEDLINE collection shows that the *idf*-based suggested method performs statistically better than Rocchio model.

Pseudo-relevance feedback (PRF) is often considered as an effective query expansion approach. However, in this method not all the terms extracted from feedback documents are useful ones. In using PRF Cao et al. (2008) claims that a term classification method (to select useful expansion terms) could be a promising approach in order to compensate the drawbacks of blind query expansion approach.

Wikipedia is considered to be a good source for query expansions by many studies. Li et al. (2007) study the performance of Wikipedia articles as external source for query expansion. Their study reports the improvement of retrieval effectiveness when using Wikipedia.

Xu et al. (2009) suggest different methods of term selection for query expansion with pseudo-relevance feedback using Wikipedia pages as the source of PRF data.

While the above-mentioned works focus on query expansion, Efron et al. (2012) propose applying document expansion in order to improve retrieval effectiveness when dealing with collections of short documents.

5.8.2 Experiment Architecture

For this experiment we use the English corpus from CLEF 2012 collection (see Section 5.4.1). Same as the previous experiments for each document only human-readable texts are used. After this extraction, the average number of distinct indexing terms per document is around 27. In using the topics, only the title field is applied in order to conduct our search. We apply the same stopword list as previous experiments and afterwards we apply a stemmer based on Porter algorithm (Porter, 1997). Finally as a weighting scheme we choose the Okapi (BM25) model (Robertson et al., 2000). As we are dealing with relatively short documents, we apply the default settings of the Okapi BM25 (Yuanhua & Zhai, 2011) while assigning $b = 0.35$ and $k_1 = 1$. Our choice of the Porter algorithm for stemming and the Okapi (BM25) model is based on our previous experiments were we evaluated different indexing strategies along with various IR models (see Section 5.5.1).

For evaluating the retrieval performance of our system we use two different measures. First we use the MAP measure (computed as explained in Section 5.1). As

mentioned before, the topics used in the collection are real queries submitted by Europeana general users. Many of these casual users are not professionals in cultural heritage domain. Consequently these users often do not expect a full and complete answer to their search and they are just seeking some information on a certain topic. In such a search the user only consults the first retrieved results hoping to find a couple of relevant answers to her/his search. Such a user does not usually need all possible answers to her/his search query (Arampatzis et al., 2009). This makes the main difference between the system that a casual Internet surfer needs and a recall-oriented professional search system. In some other domain-specific search systems (e.g., patent or medical retrieval) a professional (e.g., a patent examiner) would prefer to find as much as relevant documents than finding some relevant documents at the top ranks (Kim et al., 2011). Accordingly for evaluating our system we also adopt precision at k measure (with k equal to 5) assuming that only up to k documents is retrieved and returned to the user.

In order to evaluate different methods of query expansion we first conduct a search using the title field of the original topics. At this stage the topics are applied as such and with no further manipulation. We use this experiment as our baseline experiment. Afterwards we aim to improve the search results by expanding each topic with related concepts. As, in the used collection, we are dealing with very short topics, enriching the queries with similar concepts might improve the retrieval effectiveness. The additional terms facilitate the process of matching the relevant documents to the related query. Moreover, it helps to reduce the mismatches between the documents and the queries (Shokouhi et al., 2009). It is worth to mention that during the CLEF-CHiC 2012 campaign, the participants have the opportunity to manually add concepts to the topic descriptions. If the added terms or phrases were clearly related to the semantic content of the topic, their presence usually tended to decrease the retrieval effectiveness. Based on this finding, our aim is to evaluate automatic tools to expand the queries. Good search terms are not always semantically related ones (e.g., Paris and France, Louis XIV and “le Roi-Soleil”) but lexically related (e.g., UK and England, Britain and British) or having a relation based on the context (e.g., Ireland and IRA bombing). In this perspective, we use Wikipedia articles (see next section) as an external source. We also apply pseudo-relevance

feedback (PRF or blind-query expansion) as an alternative expansion approach (as described later).

5.8.3 Query Expansion using Wikipedia

Wikipedia contains often one or more articles for each topic, providing a summary of the most important aspects of that topic. Besides, these articles are regularly updated. Therefore we can consider Wikipedia as an appropriate resource for the aim of query expansion (Xu et al., 2009). In the case of the current experiment, Wikipedia could be a good resource also because of the nature of our topics. As mentioned before, the topics used in the collection are mostly name entities thus there is a high probability to find a Wikipedia article related to each topic.

In expanding the queries, we add three different parts to the original topics. To start our expansion we first try to find some additional search terms for each topic. We construct two different lists of additional enrichments and add them under two separate tags to the original topics. We first find those Wikipedia pages (maximum 10 pages per topic) having titles starting with the same term as the original topic title. Then we add the title of these pages to each topic (under the tag name <WikiTitle>). Afterwards we select those Wikipedia articles that contain (in the title or content) the related topic title. Then we add the title of these pages (maximum 10 additional titles per topic) under the tag <WikiContent> to each topic.

In order to provide more additional information for our topics, we enrich the topics by adding also a description to them. In producing a description for our topics, we first find the appropriate Wikipedia page which represents the topic (using the original topic title). Subsequently we add the information found in the introduction section of that page as a description to our topics. The descriptions were added under <Description> tag to each topic. A sample enriched topic is shown in Figure 5.3.

```
<topic lang="en">
   <identifier>CHIC-001</identifier>
   <title>hiroshima</title>
    <WikiTitle>Hiroshima, Hiroshima Prefecture, Hiroshima Toyo Carp, Hiroshima Big
        Arch,Hiroshima Electric Railway, Hiroshima and Nagasaki, Hiroshima mon amour,
        Hiroshima University, Hiroshima Stadium, Hiroshima Peace Memorial
   </WikiTitle>
   <WikiContent>Hiroshima, Atomic bombings of Hiroshima and Nagasaki, Hiroshima Prefecture,
        Hiroshima Big Arch, Sanfrecce Hiroshima, Hiroshima Station, Hiroshima Airport, Little
        Boy, Hiroshima Toyo Carp, Hiroshima (band)
    </WikiContent>
     <description>Town in Japan is the capital of Hiroshima Prefecture, and the largest city in the
        Chūgoku region of western Honshu, the largest island of Japan
    </description>
 </topic>
```

Figure 5.3 Example of an enriched topic

Expanding a query with appropriate search terms (terms that properly convey user's information need or those morphologically related) may improve the retrieval effectiveness. This improvement can even be achieved by adding only one appropriate term (Petras, 2005). Consequently we wish to find out which terms could be considered as appropriate search terms in our case study, what is the characteristic of these terms and which criteria should be taken into account to choose them. To do so we extract those terms from <WikiTitle> and <WikiContent> lists, which had a positive effect on the retrieval and made a new list out of them. Thereafter a new search is conducted using queries enriched with the terms in this new list. To build our new list, for each topic, we first add each element of <WikiTitle> to its corresponding topic title and search the collection with this new query. At each step we compare the obtained results with the results of the baseline experiment (experiment with only the original topic title). In this way only those terms from <WikiTitle> that improved the baseline results will be added to the new list and those with a negative impact will be discarded from the query. Same procedure is separately applied using <WikiContent>. We address this new list as "WikiTitle+WikiContent".

5.8.3.1 Results and Discussions

Table 5.22 shows the results of our various tests for query expansion using Wikipedia. As mentioned before in our tests we applied Porter stemmer and Okapi BM25 model (with b = 0.35 and k_1 = 1). In this table we can see the MAP and P@5 values for different queries with different types of expansions. We applied the *t*-test to our results to verify whether the difference between the results obtained from different expansions is statistically significant or not. In this evaluation, we considered the Title-only query formulation as the baseline for a two-tailed test with α = 1% and 5% as the significance level (marked respectively with ** and * in Table 5.22).

As the results in Table 5.22 show, adding a description to the topics significantly decreases the performance considering both MAP and P@5 (as shown under the label "Title+Description" and "Title+WikiTitle+Description"). Descriptions added more noise than useful terms to the queries. As we can see under the line "Title+WikiTitle", providing additional titles improves the retrieval effectiveness but the difference is not significant. While in the case of "Title+WikiContent" the results are not improved. On the other hand when the original title is expanded with appropriate search terms (the ones manually extracted from the two different lists) the MAP is significantly improved (as shown under the label "Title+ (WikiTitle + WikiContent)" in Table 5.22).

Table 5.22 Results for different query formulations

Query Type	MAP	P@5
Title only (Baseline)	0.4936	0.5400
Title + Description	0.3838 *	0.3900 *
Title+WikiTitle	0.4990	0.5500
Title+WikiTitle+Description	0.4703	0.5300
Title+WikiContent	0.4721	0.5400
Title+(WikiTitle+WikiContent)	**0.5137** **	**0.5800**

5.8.4 Pseudo-Relevance Feedback

In this experiment we also apply the PRF approach (see Section 3.5.1). We employ Rocchio's approach (Buckley et al., 1996) as well as *idf*-based method to 3, 5, 10, 15, 20, 50 and 70 documents from which was extracted 5, 10, 15, 30, 50 top terms.

Table 5.23 Results for PRF approach

		Mean Average Presision (MAP)	
		Without PRF	0.4990
		***idf*-based**	**Rocchio**
3 documents	5 terms	0.4703	0.5000
	10 terms	0.4426	0.5018
	15 terms	0.4528	0.5049
	30 terms	0.4199	0.5069
	50 terms	0.3850	0.4961
5 documents	5 terms	0.4500	0.4971
	10 terms	0.4482	**0.5076**
	15 terms	0.4418	**0.5078**
	30 terms	0.4186	0.4997
	50 terms	0.3844	0.4786
10 documents	5 terms	0.4535	0.4947
	10 terms	0.4406	0.5018
	15 terms	0.4079	0.4949
	30 terms	0.3983	0.4899
	50 terms	0.3968	0.4519
15 documents	5 terms	0.4533	0.4958
	10 terms	0.4376	0.5010
	15 terms	0.4071	0.4929
	30 terms	0.3817	0.4832
	50 terms	0.3701	0.4617
20 documents	5 terms	0.4397	0.4963
	10 terms	0.4372	0.4954
	15 terms	0.4104	0.4917
	30 terms	0.3843	0.4697
	50 terms	0.3556	0.4565
50 documents	5 terms	0.4067	0.4931
	10 terms	0.3778	0.4946
	15 terms	0.3672	0.4854
	30 terms	0.3293	0.4692
	50 terms	0.3035	0.4553

5.8.4.1 Results and Discussions

Table 5.23 contains the results obtained from PRF approach. We applied the PRF while using the original query expanded with the <WikiTitle> field. The results show that Rocchio's approach (Buckley et al., 1996) performs better than *idf*-based query expansion (Abdou & Savoy, 2008). From our obtained results we can see that using more than 15 documents does not help anymore to enhance the retrieval performance (consequently the results when using 70 documents are not registered in Table 5.23). With Rocchio model the MAP value increases especially when the number of documents and selected terms is not very high (3 or 5 documents with up to 30 terms). However comparing these results with our baseline, the differences are not statistically significant.

5.8.5 Query-by-Query Analysis

Analyzing the results in detail confirms that one word can make a whole change. Nevertheless without knowing precisely the user's information need, while submitting a query, choosing a proper search term is not always simple. For instance to expand a query like "Hiroshima", without knowing the user's intention while submitting this query, one might think that "Nagasaki" could be a good term to add to the original query. In our results for the first Topic ("Hiroshima") we can see that the term "Japan" helps to improve the search results while the word "Nagasaki" does not make any changes. Below there are some more examples found by applying the query-by-query analysis:

- In Topic #27 ("paul colin") adding the word "artist" leads the retrieval of relevant items in higher position which augments the P@5 from 0.6 to 0.8.

- Topic #28 ("etaples") is expanded with "Etaples art colony" and "Etaples Military Cemetery" among the other additional terms. The terms "colony" and "cemetery" cause the retrieval of a non-relevant document at the 3rd position and thus P@5 reduces from 1 to 0.8.

- For Topic #23 ("jean-jaques rousseau") the expanded query returns eight relevant documents among the top ten retrieved items. Without query

expansion, the system is able to retrieve only three relevant items in the top ten results. Here the P@5 improves from 0.4 to 1 while MAP changes from 0.3299 to 0.5232. The additional terms such as “French” or “Henri” are among the terms that caused this improvement. However the term “Henri” comes from the term “Henri Pigozzi” which leads to retrieve a relevant document containing the name “Claude Henri”.

- The additional term “Bestiary” brings a non-relevant document to the first rank for Topic #33 (“physiologus”).

- In expanding Topic #30 (“anguissola”) with “1550-1600 in fashion” and “Boy Bitten by a Lizard” the terms “Fashion” and “boy” degrades the result.

- In Topic #41 (“red kite”) the system extracts non–relevant documents at the first two ranks. While by adding the terms “Bird” and “Milvus”, we will find relevant documents at the top two ranks.

These observations show that in some cases (Topic #41 & Topic #27) adding a category to the proper name helps the retrieval of relevant documents. At the same time, in the case of Topic #30 we notice that expanding the query with another term from the same category decreases the effectiveness of the search. This shows that finding a relation between the original search term and the proper expansion term remains critical.

5.8.6 Conclusion

Throughout this experiment our goal was to experience the possibility of improving the retrieval effectiveness via query expansion when considering only a specific domain of search. In our experiment we were dealing with data coming from cultural heritage domain. We also aimed to find out how we can define and select an appropriate search term in the process of query expansion. We conducted different experiments on our collection, expanding the queries using external resources as well as applying pseudo-relevance feedback (or blind query expansion) approaches.

The obtained results confirm that query expansion helps to improve the search quality. Using external resources to collect similar concepts for enriching the original queries gives better results comparing to blind query expansion. However in order to have significant improvement in results we need to find appropriate terms which is not a trivial task. As we saw in our results nominating a term as a good term is dependent on the structure and characteristic of the documents. Also having a global knowledge of user's information need considerably helps to find these good terms. In our experiment due to the presence of some vague queries and some sparse documents, the results were not always as expected.

Conclusion & Future Work 6

Our aim in this thesis was to design and analyze a professional search system which deals with one specific field of knowledge and handles multilingual datasets. The domain of this study was cultural heritage and our data was written in 13 different languages. Our main objective was to design a system which is compatible with the specific characteristics of data in cultural heritage domain and is able to fulfill the user's information need regardless the language of the data. Moreover we aimed to enhance the performance of our designed system by applying query expansion methods. The main challenges that we were facing in this study were the frequency of names in the corpus and the user queries; the multilinguality of the cultural heritage collections and users who come from different linguistic backgrounds. Finally, having short and broad descriptions for each object and the difficulty of having a precise insight of the user's information need were also very frequent when searching into these collections.

To design and evaluate our system we conducted our project in three major phases. At a forth step we applied query expansion and evaluated its impact on the performance of our system.

First step (monolingual retrieval): In order to design our retrieval system for cultural heritage data, we used English, French and Polish corpora provided by CLEF evaluation champagne (Petras et al., 2012). On this data we evaluated different IR models, indexing strategies and data fusion technics in order to define the best strategies to develop our system.

In this step in the indexing phase we proposed a stopword list for each language. For the English language we applied S-Stemmer (Harman, 1991) and we used a light stemmer for Polish and French languages (removing inflectional suffixes). For the Polish language we also used the *n*-grams (McNamee & Mayfield, 2004) and trunc-*n* text representations. As IR models we evaluated *dtu-dtn* (Singhal, 2002) model,

Okapi (BM25) (Robertson et al., 2000), DFR-PL2, DFR-I(n_e)C2 and DFR-I(n_e)B2 models (Amati & Van Rijsbergen, 2002).

After the evaluation of different IR models and indexing strategies we considered Z-score fusion operator (Savoy, 2005; Dolamic et al., 2009) in order to merge different ranked lists resulting from different IR models. The aim was to produce a result list with relevant documents in higher ranks and the non-relevant items in lower ones.

According to our results for the English language we propose applying the DFR-I(n_e)B2 model along with the S-Stemmer. For the French language the best IR model is the Okapi model (with b=0.5) while the light stemmer misleads the search. Finally for the Polish language, the Okapi model has again the best performance. For this language we also suggest using the trunc-n indexing strategy with n=6 which produces good retrieval effectiveness. The proposed stemming approach improves the performance for Polish. In our results we did not gain a significant improvement when applying data fusion.

Second step (bilingual retrieval): Using our system with the best performance from the first step, we moved to a bilingual level. Our aim in this step was to compare different translation technics

Accordingly we used German and French topics to search the English corpus using query translation approach. We produced English translations for German and French topics and then we searched them in the English corpus. To translate the queries we used Google and the combination of Wikipedia and Google translation.

Given the results we propose using a combination of translation technics for translating. Considering the fact that the topics used in this domain are short and tend to contain name entities, the translation is not very critical and consequently the difference between different translation methods is not remarkable.

Third step (multilingual retrieval): In the third step, we used the multilingual collection (provided by CLEF evaluation campaign on cultural heritage objects) and we tried to adapt our system to a multilingual context.

In this phase we had our data in 13 different languages. As a first approach, we proposed building a single collection out of all the collections. And as the second approach we proposed to build 13 corpora and to produce the final results by merging the results of searching each of them separately. In order to build our indexes for each language we proposed different stemmers according to each language's specific syntax and morphology and we applied a stopword removal. We chose the Okapi (BM25) (Robertson et al., 2000) as our IR model in this phase (with $b = 0.75$ and $k_1 = 1.2$).

When we produced separate indexes for each language in order to merge their result lists we evaluated different merging strategies. First a round-robin approach. Second a biased round-robin approach (Savoy, 2004). As third merging strategy, we normalized the document scores within each language by dividing each document score by the maximum score. As fourth merging approach, we applied a variant of the previous one in which we normalized the document score by taking into account not only the maximum score but also the minimum one (Savoy, 2004). And as the final merging strategy, we applied the Z-score operator in which the document score is normalized by considering the average and the standard deviation of the document scores distribution in each result list (Savoy, 2004).

Our results show that when dealing with several collections in different languages the best is to treat each language separately and merge the results from each system to produce the final results. To do so the best merging operators are Z-score and the method of normalizing the document score by taking into account the maximum and the minimum score. Moreover, we propose favoring the languages with more records during the merging process. When we select more items per round from the results coming from languages with more records the performance increases. Besides, deleting the languages with fewer records from the search improves the overall performance.

Forth step (query expansion): In this step we first evaluated the effect of Rocchio's approach (Buckley et al., 1996) and *idf*-based query expansion (Abdou & Savoy, 2008) on our monolingual corpora (English, French and Polish) as two pseudo-relevance feedback approaches for query expansion. Second we used Wikipedia as an external source to expand the queries.

Given that in CH domain the documents are relatively short, applying PRF does not help to improve the performance of our system. Having short description makes it difficult to add useful search terms to the initial query. But we show that the search results enhance when using external resources to enrich the original. By having a good knowledge of the user's information need we can benefit from this method at the most.

Future work: In our study we saw that the main challenge with cultural heritage data is the lack of descriptions and short user queries. Accordingly by enriching these two we might be able to gain better search results. Different studies confirm that query expansion is a promising method to enhance the retrieval performance (see Section 5.8.1). Accordingly for future work we would like to investigate on query expansion technics. Given the fact that we have very short documents in this collection, for future work we consider trying document expansion technique (Efron et al., 2012). We would also like to try using DBpedia as well as Wikipedia with a differently structured manner in order to take more advantage of their data (Xu et al., 2009). Moreover we will consider using relevance feedback coming from other sources than pseudo relevant documents (Shokouhi et al., 2009).

References

Abdou, S. & Savoy, J., 2008. Searching in Medline: Query expansion and manual indexing. *Information Processing & Management*, 44(2), pp.781-89.

Akasereh, M. & Savoy, J., 2013. Ad Hoc Retrieval with Marathi Language. In *Multilingual Information Access in South Asian Languages*. Springer. pp.23-37.

Amati, & 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.357-89.

Arampatzis, A., Kamps, J. & Robertson, S., 2009. Where to stop reading a ranked list?: threshold optimization using truncated score distributions. In *SIGIR.*, 2009. ACM.

Baeza-Yates, R. & Ribeiro-Neto, B., 1999. *Modern Information Retrieval*. ACM Press.

Ballesteros, L. & Croft, W.B., 1997. Phrasal Translation and Query Expansion Techniques for Cross-language Information Retrieval. *SIGIR Forum*, 31, pp.84-91.

Bikel, D.M. & Zitouni, I., 2012. *Multilingual Natural Language Processing Applications From Theory to Paractice*. Pearson plc.

Boughanem, M., 2008. Introduction à la recherche d'information. In Boughanem, M. & Savoy, J. *Recherche d'information état des lieux et perspectives*. Lavoisier. pp.19-44.

Buckley, C., Singhal, A., Mitra, M. & Salton, G., 1996. New Retrieval Approaches Using SMART. In *TREC-4.*, 1996. NIST Publication.

Büttcher, S., Clarke, C.L.A. & Cormack, G.V., 2010. *Information Retrieval - Implementing and Evaluating Search Engines*. MIT Press.

Cao, G., Nie, J.-Y., Gao, J. & Robertson, S., 2008. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR '08*., 2008. ACM.

Cleverdon, C.W., 1991. The Significance of the Cranfield Tests on Index Languages. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*., 1991. ACM.

Darwish, K. & Orad, D., 2002. CLIR experiments at Maryland for Trec-2002: Evidence combination for arab-english retrieval. In *Proceedings of the text retrieval and evaluation conference*., 2002.

Demner-Fushman, D. & Oard, D.W., 2003. The effect of bilingual term list size on dictionary-based cross-language information retrieval. In *Proceedings of the 36th Hawaii international conference on system science*., 2003.

Dolamic, L., Fautsch, C. & Savoy, J., 2009. UniNE at CLEF 2008: TEL, and Persian IR. In *Evaluating Systems for Multilingual and Multimodal Information Access*. Springer. pp.178-85.

Dolamic, L. & Savoy, J., 2009. How Effective is Google's Translation Service in Search? *Commun. ACM*, 52(10), pp.139-43.

Dolamic, L. & Savoy, J., 2009. Indexing and Searching Strategies for the Russian Language. *JASIST*, 60(12), pp.2540-47.

Dolamic, & Savoy, J., 2009. Indexing and Stemming Approaches for the Czech Language. *Information Processing & Management*, 45(6), pp.714-20.

Dolamic, L. & Savoy, J., 2010. When stopword lists make the difference. *JASIST*, 61(1), pp.200-03.

Efron, M., Organisciak, P. & Fenlon, K., 2012. Improving Retrieval of Short Texts Through Document Expansion. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*., 2012. ACM.

Fautsch, C., 2009. *Domain specific information retrieval social science, blogsphere and biomedicine* . PhD Thesis. Université de Neuchâtel.

Fautsch, C. & Savoy, J., 2009. Algorithmic Stemmers or Morphological Analysis: An Evaluation. *JASIST*, 60(8), pp.1616-24.

Fox, C., 1989. A Stop List for General Text. *SIGIR Forum*, 24, pp.19-21.

Fox, E.A. & Shaw, J.A., 1993. Combination of Multiple Searches. In *The Second Text REtrieval Conference*., 1993.

Grossman, D.A. & Frieder, O., 2004. *Information Retrieval*. Springer.

Harman, D., 1991. How effective is suffixing. *JASIS*, 42, pp.7-15.

Harter, S.P., 1975. An algorithm for probabilistic indexing. *Journal of the American Society for Information Science*, 26(5), pp.280–89.

Kim, Y., Seo, J. & Croft, W.B., 2011. Automatic boolean query suggestion for professional search. In *ACM SIGIR Conference on Research and Development*., 2011. ACM.

Korenius, , Laurikkala, J., Järvelin, K. & Juhola, M., 2004. Stemming and Lemmatization in the Clustering of Finnish Text Documents. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*., 2004. ACM.

Kowalski, G., 1997. *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers.

Li, Y., Luk, W.P.R., Ho, K.S.E. & Chung, F.L.K., 2007. Improving Weak Ad-hoc Queries Using Wikipedia Asexternal Corpus. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*., 2007. ACM.

Majumder, P. et al., 2007. YASS: Yet Another Suffix Stripper. *ACM Trans. Inf. Syst.*, 25(4).

Manning, C.D., Raghavan , & Schütze, H., 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Maron, M.E. & Kuhns, J.L., 1960. On Relevance, Probabilistic Indexing and Information Retrieval. *J. ACM*, 7(3), pp.216--244.

McNamee, P. & Mayfield, J., 2004. Character N-Gram Tokenization for European Language Text Retrieval. *IR Journal*, 7, pp.73-97.

Metzler, D., Dumais, S. & Meek, , 2007. Similarity Measures for Short Segments of Text., 2007. Springer-Verlag.

Mikolov, T. et al., 2013. Distributed Representations of Words and Phrases and their Compositionality. *CoRR*, abs/1310.4546.

Nie, J.-Y. & Savoy, J., 2008. Recherche d'information multilingue. In Boughanem, M. & Savoy, J. *Recherche d'information état des lieux et perspectives*. Lavoisier. pp.139-70.

Paik, J.H., Mitra, , Parui, S.K. & Järvelin, K., 2011. GRAS: An Effective and Efficient Stemming Algorithm for Information Retrieval. *ACM Trans. Inf. Syst.*, 29(4), pp.19:1-19:24.

Paik, J.H., Parui, S.K., Pal, D. & Robertson, S.E., 2013. Effective and Robust Query Biased Stemming. *ACM Transactions on Information Systems (TOIS)*.

Pasi, G., 2010. *Information Retrieval Models: Basic models*. Ecole d'Atomne en Recherche d'Information et Applications 2010.

Peat, & Willett, , 1991. The Limitations of Term Co-Occurrence Data for Query Expansion in Document Retrieval Systems. *JASIS*, 42(5), pp.378-83.

Peters, C., Braschler, M. & Clough, P., 2012. *Multilingual Information Retrieval: From Research to Practice*. Springer-Verlag.

Petras, V., 2005. How one Word can make all the Difference - Using Subject Metadata for Automatic Query Expansion and Reformulation. In *In Working Notes for the CLEF 2005 Workshop*. Vienna, 2005.

Petras, V., Bogers, T., Ferro, N. & Masiero, I., 2013. *Cultural Heritage in CLEF (CHiC) 2013 – Multilingual Task Overview*. [Online] Available at: ceur-ws.org/Vol-1179/CLEF2013wn-CHiC-PetrasEt2013.pdf.

Petras, V. et al., 2012. Cultural Heritage in CLEF (CHiC) Overview 2012. In *Working Notes for the CLEF 2012 Workshop*., 2012.

Porter, M.F., 1997. An algorithm for suffix stripping. In *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc. pp.313-16.

Rijsbergen, C.J.V., 1979. *Information Retrieval*. Butterworth-Heinemann.

Robertson, S.E., 1997. The Probability Ranking Principle in IR. In K.a.W.P. Sparck Jones, ed. *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc. pp.281-86.

Robertson, S.E., Walker , & Hancock-Beaulieu, , 2000. Experimentation as a way of life: Okapi at TREC. *Inf. Process. Manage.*, 36(1), pp.95-108.

Robertson, S.E. & Walker, S., 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*., 1994. Springer-Verlag.

Sahami, M. & Heilman, T.D., 2006. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. In *Proceedings of the 15th International Conference on World Wide Web*., 2006. ACM.

Salton, G., Fox, E.A. & Wu, H., 1983. Extended Boolean Information Retrieval. *Commun. ACM*, 26(11), pp.1022--1036.

Sanderson, M., 1996. *Word sense disambiguation and information retrieval*. Ph.D. thesis. University of Glasgow.

Savoy , J. & Rasolofo, Y., 2003. Report on the TREC-11 Experiment: Arabic, Named Page and Topic Distillation Searches. In *Proceedings of the eleventh text retrieval conference TREC-2002*., 2003.

Savoy, J., 1999. A Stemming Procedure and Stopword List for General French Corpora. *J. Am. Soc. Inf. Sci.*, 50(10), pp.944-52.

Savoy, J., 2004. Combining Multiple Strategies for Effective Monolingual and Cross-Lingual Retrieval. *IR Journal*, 7, pp.121-148.

Savoy, J., 2005. Data Fusion for Effective European Monolingual Information Retrieval. In *Multilingual Information Access for Text, Speech and Images*. Springer. pp.233-44.

Savoy, J., 2006. Light Stemming Approaches for the French, Portuguese, German and Hungarian Languages. In *Proceedings of the 2006 ACM Symposium on Applied Computing*., 2006. ACM.

Savoy, J., 2008. Searching strategies for the Hungarian language. *Information Processing & Management* , 44(1), pp.310-24.

Savoy, J. & Berger, P.-Y., 2005. Selection and Merging Strategies for Multilingual Information Retrieval. In *Multilingual Information Access for Text, Speech and Images*. Springer. pp.27-37.

Shokouhi, M., Azzopardi, L. & Thomas, P., 2009. Effective query expansion for federated search. In *SIGIR '09*. New York, 2009. ACM.

Singhal, A., 2002. AT&T at TREC-6. In *Proceedings of the Eighteenth Annual International ACM SIGIR.*, 2002.

Singhal, A., Salton, G., Mitra, M. & Buckley, C., 1996. Document length normalisation. *Information processing and management*, 32(5), pp.619-33.

Sparck-Jones, K. & van Rijsbergen, C.J., 1975. *Report on the need for and provision of an "ideal" judgements retrieval test collection*. University of Cambridge.

Vogt, C.C. & Cottrell, G.W., 1999. Fusion Via a Linear Combination of Scores. *Inf. Retr.*, 1, pp.151-73.

Xu, , Jones, G.J.F. & Wang, B., 2009. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32Nd International ACM SIGIR*

Conference on Research and Development in Information Retrieval. New York, 2009. ACM.

Yu, H. & Agichtein, E., 2003. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19(1), pp.340-49.

Yuanhua, L. & Zhai, C., 2011. When documents are very long, BM25 fails! In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*., 2011. ACM.