Université de Neuchâtel
Faculté des Sciences
Institut d'Informatique

Information Retrieval of Digitized Medieval Manuscripts

par

Nada Naji

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

Acceptée sur proposition du jury :

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

Prof. Sylvie Calabretto, rapporteur
LIRIS, INSA de Lyon, France

Prof. Elöd Egyed-Zsigmond, rapporteur
LIRIS, INSA de Lyon, France

Prof. Peter Kropf, rapporteur
Université de Neuchâtel, Suisse

Soutenue le 11 septembre 2013

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 Nada NAJI

Titre:

"Information Retrieval of Digitized Medieval Manuscripts"

sur le rapport des membres du jury composé comme suit:

- Prof. Jacques Savoy (Université de Neuchâtel), directeur de thèse
- Prof. Peter Kropf (Université de Neuchâtel)
- Prof. Sylvie Calabretto, (INSA-LIRIS, Université Claude Bernard, Lyon)
- Prof. Elöd Egyed-Zsigmond (INSA-LIRIS, Université Claude Bernard, Lyon)

Neuchâtel, le 29 octobre 2013

Le Doyen, Prof. P. Kropf

Dedication

To my beloved brother, Noor

Acknowledgements

I will forever be thankful to my advisor Professor Jacques Savoy for his constant guidance, wisdom, encouragement, support, kindness and help over the past years.
Doing my PhD under his supervision was such a priceless opportunity that I profoundly appreciate.

I would like to express my thanks to Professor Peter Kropf, Professor Sylvie Calabretto, and Professor Elöd Egyed-Zsigmond for being part of the jury and for the time and energy that they dedicated to evaluate this PhD dissertation.

This dissertation is a part of the HisDoc synergy project of the Swiss National Science Foundation (CRSI22 125220). Being a team member of this project, I had the valuable opportunity to collaborate with Professor Horst Bunke, Professor Rolf Ingold, Dr. Andreas Fischer, and Micheal Baechler whom I would like to thank for this fruitful collaboration. I would like to thank Professor Michael Stolz and Dr. Gabriel Viehhauser for providing the Parzival transcriptions, Professor Anton Näf and Eva Wiedenkeller for their time and valuable advice in the Middle High German language. My thanks to Ljiljana for her help throughout the beginning of my PhD work and for the nice time and discussions that we had together.

I would like to thank all my colleagues in the Computer Science Department of the University of Neuchatel for making my PhD years such an unforgettable experience with all our discussions, coffee breaks, "torrées", and "soupers". I would like to thank all of my friends whose encouragement and support I truly appreciate. I would like to thank my friends Dilek and Derin for the warm company and the nice times we spent together, but also for their support during the hard times.

No words can express my gratitude and love to my parents and my brother for their unconditional love, care, patience, and constant support in each and every way one can possibly imagine. I love them so much and I would have never made it this far without their love.

Abstract

This dissertation investigates the retrieval of noisy texts in general and digitized historical manuscripts in particular. The noise originates from several sources, these include imperfect text recognition (6% word error rate), spelling variation, non-standardized grammar, in addition to user-side confusion due to her/his limited knowledge of the underlying language and/or the searched text. Manual correction or normalization are very time-consuming and resource-demanding tasks and are thus out of the question. Furthermore, external resources, such as thesauri, are not available for the older, lesser-known languages. In this dissertation, we present our contributions to overcoming or at least coping with these issues. We developed several methods that provide a low-cost yet highly-effective text representation to limit the negative impact of recognition error and the variable orthography and morphology. Finally, to account for the user-confusion problem, we developed a low-cost query enrichment function which we deem indispensable for the challenging task of one-word queries.

Keywords: Information Retrieval of Noisy texts, Information Retrieval of Handwritten Documents, OCR, Text Recognition, Digital Libraries, Middle High German, Medieval Manuscripts.

Contents

List of Figures

List of Tables

1

Introduction

There has been a growing interest in building Digital Libraries (DL) during the last decades. The main motivation is to preserve our Cultural Heritage (CH) and make this valuable material available worldwide and easily accessible by users. Museums, libraries, linguists, historian, paleographers, as well as the general public are examples of users of DL whose interests, skills and backgrounds vary greatly. Having access to the original historical documents such as manuscripts, maps and music scores is usually limited to a certain group of users due to the fragility and delicacy of the writing media of this precious heritage, and consequently a wide range of users is left deprived. Therefore, the need has arisen to transfer the experience of examining and browsing these historical documents into the digital world by building portals and search engines devoted for CH.

1.1 Motivation

From an Information Technology (IT) perspective, handling historical manuscripts is a significantly challenging task and the currently available solutions still need to be improved. The work presented in this dissertation constitutes a part of a larger project named HisDoc: Historical Document Analysis, Recognition, and Retrieval. Our work addresses the third and last module of HisDoc, the Information Retrieval (IR) one. In this project, historical manuscripts are automatically transformed from ink on parchment or paper into fully searchable digital texts while keeping human intervention to the least extent possible.

The manuscripts suffer from degradation due to aging factors such as stains, holes, faded writing and ink bleeding. Moreover, they were written during times when the writing material (be it paper or parchment) was expensive. Therefore, stitching a tear in a page was a quite common practice to keep the parts together and thus mending a disintegrating page rather than reproducing it. For the same *economical* reason, additional texts were to be added within marginal spaces as well as in-between existing text-lines. All these degradation elements make the automatic processing of these manuscripts a tougher task and augment the magnitude of difficulty that we encounter in HisDoc. A higher recognition error rate (i.e., a lower recognition accuracy level) would definitely impact the performance of IR. This issue becomes more crucial when the documents and/or queries are rather short especially when affecting rare words, such as proper names, thus making it impossible to retrieve the sought piece of information using the traditional IR methods.

Text recognition of historical manuscripts is a challenging task given that even the best performing systems known today are still incapable of delivering a 100% correct transcription. In addition to all the difficulties described so far, the non-uniform handwriting produced by several writers who wrote a single manuscript adds yet another challenge especially when it comes to the quality of text recognition, which once suffering, will penetrate its imperfections into the IR process. Moreover, there are certain issues that arise when particularly dealing with older languages such as the high orthographic and morphological flexibility and the regional and dialectical variations (more in Section 2.1.5) which, once again, cannot be handled by the simple IR methods.

Language tools, rules or resources from the modern age are therefore less interesting when it comes to applications on older languages. Modern dictionaries and thesauri, text recognition systems, and text processing elements offer limited use when it comes to historical documents. It is, therefore, necessary to develop better digital services, methods, and practices to convert these documents into the convenient digital formats with the main aim of making them available and effectively searchable by the public.

1.2 Objectives

The main objective of this dissertation is to evaluate the traditional IR practices when dealing with noisy texts in general. More precisely, studying the behavior of IR procedures when facing with different sources of noise and spotting the differences in these behavior patterns with respect to the source of noise.

The noise may originate from imperfect text recognition (character-based or word-based), corrupted queries due to user term confusion and false memories, the orthographic and morphological variability of the underlying language, or a combination of some or all of these noise sources. The first goal is thus to quantify the quality of the answers returned by the IR system. This stands for the percentage of degradation in IR effectiveness due to the issues described above.

Our second objective is two-fold: Firstly, we want to assess the robustness of our IR methods and practices with the lack of the necessary external resources such as thesauri. We aimed at assessing the usefulness of using and upcycling the limited in-house data that we have on hand (basically the by-product of the text recognition process - which is usually discarded) to build up the needed resources such as expansion term pools and corpus surrogates. Secondly, we want to find out the extent to which performing high-quality IR would still be feasible without the need to manually or automatically correct the recognition errors, if applicable, (as suggested by some studies, Section 2.2).

Our contributions, findings and recommendations of the methods and strategies open the horizon to even further research and raise curiousity (ours, at least) of whether these practices would still be as lucrative for other languages and datasets.

1.3 Organization of the Dissertation

This dissertation consists of eight chapters which are structured as follows:

This first chapter is a brief introduction to the main drivers to conducting this research, the search problem and questions that were put to experimenting and investigation throughout the course of our work.

Chapter 2 provides an overview of the related research achieved so far as well as some

key concepts and essential definitions related to information retrieval with just enough details to establish a comprehensive common-ground for a better reading experience through the chapters that follow.

In Chapter 3, we present the test-collections upon which the practical part of this research was based. Written in different languages and originating from different eras of time, yet both collections are based on imperfectly digitized text images.

The second part of this dissertation is composed of five chapters, all of which are devoted to the technical and empirical aspects of this dissertation, presenting detailedly the experiments that we conducted and the results obtained with the following structure:

Chapter 4 is concerned with the results obtained from the experiments based on the TREC-5 confusion track. While the main focal points of Chapters 5 and 6 are the different approaches adopted in the case of the Middle High German Parzival test-collection, in particular when facing with clean queries. Chapter 7 extends the the preceding chapters by introducing a new dimension to the problem domain by closely inspecting and approaching retrieval with noisy (corrupted) queries. It also presents our experiments and proposed solution for this critical issue.

Finally, Chapter 8 concludes this dissertation and highlights its main contributions and findings, challenges and limitations, and last but not least, its outlook and future work.

2

State of the Art

This chapter presents a brief overview of IR utilities and methods in general. It also presents previous and state-of-the-art research addressing the problem of noisy text retrieval and the handling of historical languages.

2.1 Key Concepts and Definitions

2.1.1 Information Retrieval

Whether it is looking for a contact email address, finding the lyrics of a music track on one's smartphone, using a Web search engine to find a recipe with certain ingredients, or trying to find out the birth place of a famous figure. These ostensive examples of Information Retrieval (IR) are undoubtedly convenient since IR has become an inseparable part of (almost) everyone's everyday life. However, in a proper academic fashion, Information Retrieval can be defined as follows:

> *Information Retrieval is the process of searching a stored set of information items to find ones that help solving the user problem.*

A typical situation can be depicted as a user having certain information needs, which is formulated as a topic that is then passed to an IR system (a search engine). The IR system's job now is to find from the corpus (a bunch of documents which may come in different formats, e.g., text, sound, images, etc.) items that satisfy these information needs the most. This applies to a wide spectrum of users and IR systems. A familiar example is the World Wide Web search, wherein a user query is submitted to the search

engine in use, which in turn searches the documents available in the World Wide Web picking the most relevant ones and returns, as results, a ranked list of these (supposedly) relevant documents.

Personal Information Management (PIM) is another example, that is, when the user is trying to find a certain file on her/his machine by feeding in the file name and/or other related specifications such as the file format or its creation date. The expected (and optimal) result is the file that the user is aiming to find. In practice, however, and depending on the user query, the system may return that file together with a number of other files that might be of less or no interest to the user.

The traditional (and most common way) to describe the user's search needs is via text. Other query types also exist, such as query by example (image) and query by humming (singing). These non-textual querying methods are not covered here as they lie beyond the scope of this dissertation which is solely concerned with searching texts. It is worth-mentioning at this point that, during a certain phase of our experiments, the final documents were actually image files whose textual content was being subject to searching. Thus, we have performed image retrieval where the underlying representation of the images is purely textual and the information needs were formulated as text as well. The term *document* in this dissertation only refers to text documents whenever mentioned from now on. The set of documents forms a *corpus*.

A document may constitute a paragraph extracted from a novel, a book chapter, a newspaper article or simply a scientific article's abstract. It can even be as short as a line of text which might represent a poetical verse, an image caption, or the contents of a cell in a table.

The simplest way to perform search is to match the given topic to the documents' text by scanning the latter sequentially until a match is (or several matches are) found. Though simple and inexpensive to implement, the cost in terms of performance is very high as this method is as naïve as inefficient. The performance worsens even more as topics get more complicated and the documents' size and/or number grow(s). This is where *indexing* comes in to provide an efficient solution. The corpus should thus be indexed as a first step after which an inverted file of this index is produced.

An inverted file is analogous to a book index. A list of terms (such as words and acronyms) that are sorted alphabetically, each associated with the page numbers containing that term. This concept applies exactly to the inverted file, wherein indexing terms point to documents in which they appear (instead of pages). Documents are thus represented by their indexing terms. Figure 2.1 depicts the overall IR process.

We used automatic indexing throughout our experiments. Our system is capable of handling full text documents and topics and converting them into optimized indexes, some or all of the following steps are involved in this process:

1. Tokenization: transformation of documents with continuous texts into a set of tokens (terms) by separating them at white spaces and punctuation symbols.

2. Case normalization: conversion to lower-case letter.

3. Special words handling: such as emails, acronyms.

4. Stoplist: Incorporating stop-word lists to eliminate non-content-bearing terms (e.g., "and", "the"). Depending on the corpus, very short terms (less than three characters length) and very frequent uninformative terms might also be considered as stop-words.

5. Accent handling: Substituting an accented letter by its unaccented form (e.g., café → cafe, Zürich → Zuerich).

6. Word normalization and decompounding: Instead of whole words, terms can be normalized into their stems or lemmas. Examples of normalization methods are:

   - Stemming: usually removing suffixes such as plural signs and derivational and inflectional suffixes (books → book, running → run, Motoren → Motor)

   - Truncation: retaining the sequence of n letters at the beginning of each token and discarding the rest (computer —(Trunc-4)→ comp).

   - n-gram: overlapping sequences of letters of each token by a sliding window of n letters (dictionary —(4-gram)→ dict, icti, ctio,..., nary).

   - decompounding; transforming compound words into their composing terms ("Apfelsaft" [apple juice] → "Apfel" [apple] + "Saft" [juice])

The selection of these steps varies according to the language of the underlying corpus.

As can be seen, an IR system is not only responsible for performing the search, it should first of all, process the documents and store them in a convenient format (indexes). Once the indexing phase is finished, the corpus is efficiently searchable, thus user topics can be submitted to the IR system. These are then indexed as queries in the same fashion as the documents and thus passing through some processing steps similar to those undergone by the documents. After that point, the IR system attempts at finding the documents that are the most *similar* to the query, in other words, ones that, supposedly, best satisfy the user information needs or answer his/her initial questions. Finally, the IR system returns a ranked list of the matching documents to the user as a search result. The ranking is essentially based on the similarity between the documents and the query, with the highest ranked document being the most similar (i.e., the best match) to the document. However, in the World Wide Web for instance, ranking is affected by other factors such as page click-through rate and popularity.

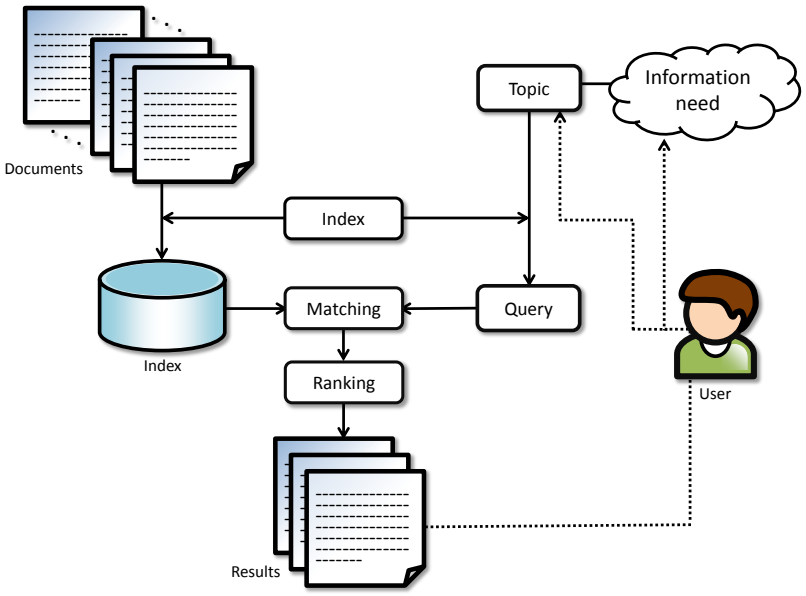

Figure 2.1: The IR process

2.1.2 Retrieval Models

IR models specify precisely how the topics and the documents are represented and how the indexing is performed. IR models must also specify how these documents' surrogates are matched to the queries (the topic representations). In this dissertation we are interested in two major groups of IR models, these are: the vector space (VSM) and probabilistic models. A basic description of several implementations of these two classes is presented in the following sections.

Vector Space Model

The Vector Space Model (VSM) [53] represents queries and documents as vectors (hence the name) in a multidimensional space. Each term t represents a dimension in the space and belongs to either a document d or a query q. Figure 2.2 depicts a three-dimensional vector space, with three terms t_1, t_2 and t_3 which represent the indexing terms of a query q. The figure also shows the document d_1, among whose indexing terms are t_1 and t_2 (but not t_3). Document d_2, on the other hand, contains all three terms t_1, t_2 and t_3 in its index.

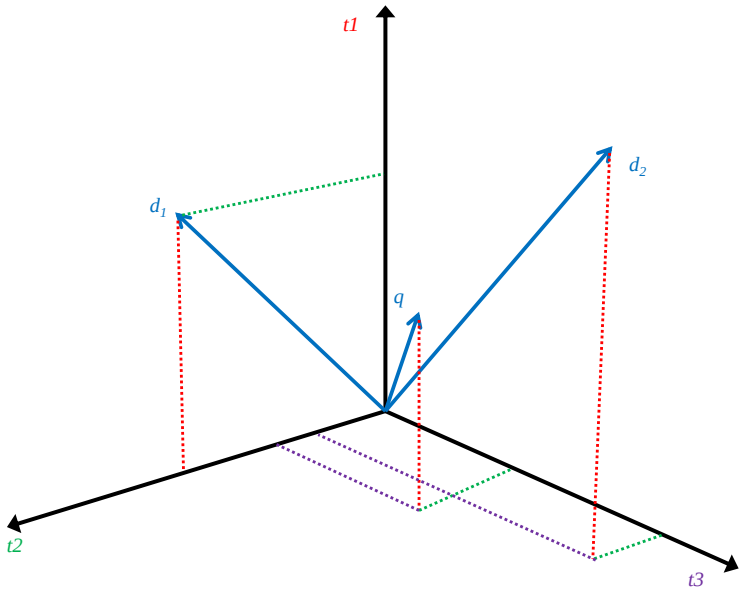

Figure 2.2: The vector space model, a generic example

For each term t_i in a document d_j, a weight w_{ij} should be assigned to reflect how much importance this term conveys with regard to the semantic content of that document. There are several formulas according to which term weights can be calculated. The simplest and one of the most widely-used formulas is given in Equation 2.1 below:

$$w_{ij} = tf_{ij} \cdot idf_i \tag{2.1}$$

Where tf_{ij} represents the number of occurrences of a term t_i in a document d_j and idf stands for the *inverse document frequency* of the term t_i. The basic formula to calculate the idf is as follows:

$$idf = log\frac{n}{df_i} \tag{2.2}$$

Where n is the total number of documents in a corpus c and df_i is the *document frequency* of the term t_i (i.e., the number of documents in which the term t_i appears).

When calculating term weights, queries are treated as documents, and hence term weights within the query are calculated using the same formula as for the documents. Once the query term weights and the document term weights are obtained for each term t_i, the similarity between the query q and the document d_j can then be calculated. The similarity between the document and the query is equivalent to the dot product of their corresponding vectors as shown in the following formula:

$$sim(d_j,\ q) = \sum_{i=1}^{|q|} w_{ij}\ .\ w_{iq} \tag{2.3}$$

where $|q|$ represents the number of indexing terms in the query q.

Alternatively, the similarity can be obtained by computing the cosine angle between the query and the document vectors:

$$sim(d_j,\ q) = \frac{\sum_{i=1}^{|q|} w_{ij}\ .\ w_{iq}}{\sqrt{\sum_{n=1}^{t} w_{nj}^2} \times \sqrt{\sum_{m=1}^{|q|} w_{mq}^2}} \tag{2.4}$$

Where t indicates the number of indexing terms. The similarity reflects the relevance of a document to the given query. In order to obtain similarity values ranging between 0 and 1 (where 0 means that the query and the document have no terms in common), the cosine measure can be employed (Equation 2.4 above).

There exist several variations that take into consideration the document length (i.e., the number of terms) when calculating the similarity levels. In certain implementations, the occurrence of a term in a relatively short document gives more power to that document in comparison to longer ones having the same term with the same term frequency.

Meanwhile, there exist other models that prevent the length-based competitiveness among documents, such as the *Lnu-ltc* [9] and *dtu-dtn* [55] models. In *Lnu-ltc*, the document term weight is referred to as (*Lnu*) which can be calculated using Equation 2.5.

$$w_{ij} = \frac{\frac{[ln(tf_{ij})+1]}{[ln(meantf)+1]}}{(1-slope)\cdot pivot + slope \cdot nt_i} \tag{2.5}$$

Where *meantf* is the mean term frequency in the document d_i, nt_i represents the number of distinct indexing terms in a document d_i and the *pivot* and *slope* represent factors that adjust the value of term weight normalization based on the document length. The weight of the query term (*ltc*) is calculated using Equation 2.6.

$$w_{qj} = \frac{(ln(tf_{qj}) + 1) \cdot idf_j}{\sqrt{\sum_{k=1}^{t}((ln(tf_{qk}) + 1) \cdot idf_k)^2}} \tag{2.6}$$

In the *dtu-dtn* model, the document term weight is calculated using Equation 2.7.

$$w_{ij} = \frac{[ln(ln(tf_{ij}) + 1)] \cdot idf_j}{(1 - slope) \cdot pivot + slope \cdot nt_i} \tag{2.7}$$

while the query term weight is calculated according to Equation 2.8.

$$w_{qj} = [ln(ln(tf_{qj}) + 1)] \cdot idf_j \tag{2.8}$$

This formulation prevents the retrieval system from favoring short documents over their longer counterparts, more precisely, ones exceeding the mean length based on the reported value of *pivot*.

Probabilistic Model

Reported in 1977 by Robertson [49], the *Probability Ranking Principle* (PRP) is the root of the probabilistic model. The idea behind this model is to discriminate between two classes of documents: relevant and non-relevant. Documents are ranked according to their estimated *probability* of being *relevant* to a given query. The PRP, whose formula is shown in Equation 2.9, suggests that for a given query q, a document d_j can belong to one of two sets: a set of relevant documents (denoted R), or another of non-relevant ones (denoted $\bar{R}$)

$$score(d_j,\ q) = \frac{P(R|d_j)}{P(\bar{R}|d_j)} \tag{2.9}$$

where $P(R|d_j)$ is the probability that the document d_j is relevant to the query q and thus belonging to the set R. On the other hand, $\bar{R}$ is the complementing

set to R, and hence $P(\bar{R}|d_j)$ = 1-$P(R|d_j)$ which represents the probability of d_j being non-relevant with respect to q.
By applying the simple staement of Bayes' theorem:

$$P(R|d_j) = \frac{P(d_j|R) \cdot P(R)}{P(d_j)} \tag{2.10}$$

we obtain:

$$score(d_j,\ q) = \frac{P(d_j|R) \cdot P(R)}{P(d_j|\bar{R}) \cdot P(\bar{R})} = \frac{P(d_j|R)}{P(d_j|\bar{R})} \cdot \frac{P(R)}{P(\bar{R})} \tag{2.11}$$

Since P(R) and P($\bar{R}$), representing the prior probabilities of a document being relevant or non-relevant respectively. These estimates are usually the same for all documents in the corpus, thus the documents can then be ranked according to the following formula:

$$score(d_j,\ q) = \frac{P(d_j|R)}{P(d_j|\bar{R})} \tag{2.12}$$

The rest of this section highlights three different implementations of the probabilistic model, all of which were employed in our experiments.

Okapi Given a query q and a document d_k, this model takes into consideration both the term frequency tf_{ik} of a term t_i and the document length l_k for determining the probability that d_k is relevant to q [47, 48]. In our experiments, we adopted the BM25 (Best Match 25) whose scoring formula is shown in Equation 2.13 below:

$$score(d_k,\ q) = \sum_{t_i \in Q} qtf_i . log\left[\frac{n - df_i}{df_i}\right] . \frac{(k_1 + 1).tf_{ik}}{K + tf_{ik}} \tag{2.13}$$

$$K = k_1 .[(1-b) + b.\frac{l_k}{avdl}] \tag{2.14}$$

where b and k_1 are constants that are usually adjusted according to the corpus being used, qtf_i represents the frequency of a term t_i in the query q, n is the total number of documents in the corpus, df_i is the *document frequency* of the term t_i (i.e. the number of documents in which the term t_i occurs at least once) and last but not least, *avdl* denotes for the average document length in that corpus.

Divergence from Randomness (DFR) The main idea behind the DFR paradigm [4] is that terms encountering random distributions across the corpus bear only little information. In a given document, the term frequency of an informative term diverges from its frequency within the corpus. The level of this divergence is proportional to the degree of informativeness that the term contributes to that document. The relevancy score is provided in Equation 2.15 below:

$$score(d_k,\ q) = \sum_{i=1}^{|q|} w_{ik}\ .\ qtf_i \tag{2.15}$$

where qtf_i is the frequency of the term t_i in the query q and $|q|$ is the number of terms in the query. Term weighting in the document w_{ik} is based on two components:

$$w_{ik} = Inf^1_{ik}.Inf^2_{ik} \tag{2.16}$$

where $Inf^1{}_{ik} = -log(Prob_1)$ which is the probability of having tf_{ik} occurrences of the term t_i in the document d_j based on the selected model of randomness and $Inf^2{}_{ik} = 1 - Prob_2$ where $Prob_2$ represents the risk of accepting a term as a document descriptor. The term frequency tf_i should be normalized before being used to calculate $Prob_1$ and $Prob_2$, which is usually referred to as the *second normalization* [4].

Language Model (LM) Instead of modeling the probability of relevance of a document to a query, the language model approach creates for each document in the corpus a probabilistic language model and ranks the documents based on the probability that the underlying model generates the submitted query [36]. In other words, a document d_k is likely to be relevant to a given query q if the document language model M_{d_k} is likely to have generated that query.

$$score(d_k,\ q) = P(q|M_{d_k}) \tag{2.17}$$

Several variations were proposed to calculate $P(q|M_{d_k})$. In our experiments we adopted the approach proposed by Hiemstra [27].

2.1.3 Retrieval Performance: Measuring Effectiveness

IR effectiveness has to do with the quality of the retrieval results, that is the pertinence of the retrieved documents to the given query/ies and the user satisfaction of these results. In an ideal world, an IR system should *only* retrieve *all* the relevant documents with respect to a given query.

A definition of *relevance* or *relevancy* is indispensable at this point. Relevance has to do with subjective, topical pertinence to a certain matter. A document is relevant to a query if it encompasses a degree of relativeness with respect to that query, that is to say, both are concerned with the same topic. From an IR point of view, documents that are highly similar to a given query are considered as relevant while from a user perspective this concept still applies but however relevancy gets purely subjective and *situational* [8, 15, 64] and may change over time and knowledge acquisition. Therefore, in addition to its similarity and the associated scores, other factors also contribute to shaping the user's evaluation of relevance of a given document such as broadness, timeliness, and information novelty.

Since this world is far from being perfect, a more realistic expected performance from an IR system would be to return many relevant documents (all of them or

at least as many as possible) and at the same time keeping the number of non-relevant ones to a minimum. Despite its being more lenient, this aim is not as easy to achieve as it seems. One simple reason is the definition of relevance and how it can be different from one user to another (subjectivity), or even change over time for the same user (timeliness).

There are two key statistics for measuring the effectiveness of an IR system which are calculated based on the results (documents) returned by the IR system for a certain query/set of queries:

(a) **Precision** The ratio of the number of the relevant documents that the system succeeded to retrieve for a given query to the number the retrieved documents (both relevant and non-relevant).

$$Precision = \frac{|Relevant \cap Retrieved|}{|Retrieved|} \tag{2.18}$$

(b) **Recall** The ratio of the number of the relevant documents that the system succeeded to retrieve for a given query to the number of the relevant documents for that query (retrieved or left out).

$$Recall = \frac{|Relevant \cap Retrieved|}{|Relevant|} \tag{2.19}$$

To calculate the precision and recall, all the relevant documents must be determined for each query beforehand. When using empirical test-collections (e.g., TREC and CLEF) a list stating the relevant documents for each query (relevance assessments/judgment) is usually made available together with the document set (corpus) and the queries.

2.1.4 Relevance Feedback and Query Expansion

After examining the retrieved documents for a given query, the user may learn from their content and reformulate the initial topic seeking enhanced results (usually novel documents) by broadening or narrowing down the topical scope.

Sometimes, the system itself does this query reformulation without the need to

interact with the user. There are several approaches that help the system in its assumptions of what could be a good expansion term.

These approaches can be split into two main classes [36], namely *global methods* and *local methods*. The global methods, which are independent of the retrieval results, include (blind) query expansion using either an external or a manual thesaurus, an automatically generated thesaurus based on the corpus or by using different spelling correction techniques (if needed). Local methods on the other hand, include relevance feedback and its different applications (blind/pseudo and indirect) and attempt to reformulate the query by taking into account the retrieval results, i.e., the presumably relevant document(s) based on their similarity to the query. Both approaches can be performed automatically or, alternatively, by involving the user. An example of an *automatic* implementation of a *global* method is expanding queries by adding terms based on a thesaurus (*global*: not based on prior search results, and *automatic*: performed without *asking* the user).

2.1.5 Middle High German

A significant part of our work is concerned with Middle High German (MHG) which is a linguistic period spanning the AD years 1100 through 1350 as reported by [26], the MHG period is also said to fall between 1050 and 1350[1], while according to [41] the term may cover an extended period up to 1500. The chronological order of the MHG comes second to Old High German (OHG: 8^{th} century to approx. 1100), and precedes the Early New High German (ENHG: 1350 - 1600) which is, lastly, followed by the New High German period (NHG: since 1600).

Unlike some languages (e.g., Spanish and French), German spelling was not standardized until 1901/1902, prior to that date, spelling was done phonographically [26] (i.e., writing like one talks, or should we rather say *"raitin laik wun tawks"*). Before 1900, German spelling varied highly (with many dialects) depending on the time and the region [19, 32, 43]. Certain letters of the alphabet were used interchangeably in MHG; *v* and *f* (*fogel vs. vogel*. It is worth-mentioning here that capitalization of nouns' initials was not present in MHG), and *z* and *c* (*Parzifal*

[1] http://en.wikipedia.org/wiki/Middle_High_German, accessed August 27^{th}, 2012 15:20

vs. Parcifal) are among others.

There is a high rate of graphical variants in MHG representing the different phonological variations depending on the dialects and styles. Several grapheme variants are not present anymore in the corresponding modern language (e.g., *å* and *á* for the grapheme *a*). All these aspects allowed more flexibility to the writers thus making part of each one's writing style in addition to regional variations, and more surprisingly, even within the same region or locality, significant writer-wise variations also existed. Normalized editions of manuscripts can also be found, such editions eliminate a great deal of the aforementioned irregularities - this might help enhance the overall IR performance. However, these normalized version are not always available as they require manual transcription by the experts, which is a very time-consuming task. Even when available, utilizing the normalized versions is not always desirable if the aim is to index the unedited versions of the original texts.

The German language is also known for its compound construction (e.g., world-wide, handgun). For instance, the word *Kühlschrank* (refrigrator) is composed of two words, namely *kühl* (cold) and *Schrank* (cupboard). Word compounding, however, was not used as frequently in MHG as it is in modern German. The percentage of compound nouns to nouns in the first half of the 13^{th} century was around 6.8%, this ratio increased over the centuries reaching 25.2% in the modern German language [23].

2.2 Related Work

In the task of allowing effective search and access to digitally available media, there have been several projects of different scales (content size and type, budget, target users, etc.) interested in building DL and CH portals such as the European Library[1], Europeana[2], Gallica[3], and e-codices (the Virtual Manuscript Library of Switzerland)[4].

[1] http://www.theeuropeanlibrary.org
[2] http://www.europeana.eu
[3] http://gallica.bnf.fr
[4] http://www.e-codices.unifr.ch

The first challenge encountered when digitizing historical documents is the quality of digitization and text recognition of scanned images. In this regard, there are only a few large-scale studies that tackled the loss of effectiveness caused by imperfect character recognition [12, 39]. In [58], it was shown that when using high-definition images and a high quality OCR system, the error rate could be limited to around 2%. Dealing with historical handwritten manuscripts instead of typed text or with colored paper and writing media suffering impurities such as stains, holes and stitches instead of a high-contrast black-and-white, the final error rate would definitely exceed the estimated 2%.

In order to explain the impact of noisy data on the retrieval quality of an IR system, it is necessary to realize the robustness that redundancy provides to any natural language when facing with a low recognition error rate. Due to this fact, the event of incorrectly-recognizing a certain term several times is quite rare and as a result the retrieval system would still be able to find the target items. If somehow the error affects a searched term more than once, the retrieval quality will definitely suffer as a result.

A corpus based on OCR'd text tends to produce an index with low frequency terms and many *hapax legomena* (terms occurring only once). This phenomenon was addressed by [58, 60] who pointed out that the proportion of such terms may be as high as 70% which exceeds the theoretically predicted 50% by Zipf's Law. As another consequence, the frequencies will tend to be smaller thus generating higher *idf* values. The higher the error rate, the worse the overall IR outcome would be, this issue may get even more critical if the errors occur in relatively short documents [40].

When analyzing other IR system components in the case of handling noisy texts, no clear conclusion seems to be possible. Taghva *et al.* [58] for example, suggest ignoring all stemmers, or at least applying only a light stemmer (eliminating only the plural "-s" suffixes [25]). On the other hand, the best TREC-5 system [7] used an aggressive stemmer [45]. Knowing that recognition errors mostly affect short words, which means words of two or three letters in length, the systematic elimination of these short words is worth considering [59].

Other authors suggest to automatically correct noisy data before proceeding with the indexing process. In this regard, several correction methods have been introduced [44], some of which rely on the Web to identify correct spellings [17]. When handling less popular or ancient languages however, using the Web is of limited value. Moreover, automatically correcting the whole corpus is usually a costly task. This strategy of limiting spelling correction to search terms was adopted by major commercial search engines. Despite all the approaches that suggest correcting OCR results [33, 56, 61] language-specific matters such as historical printing style and language evolution were usually ignored [26].

Alternatively, indexing can be based on consecutive sequences of n characters [37] rather than words. Ballerini *et al.* [7] suggest using sequences superimposed on trigrams or 4-grams. When the error rate is around 10%, an n-gram-based indexing scheme tends to provide better quality (+10% in average) than a word-based one [24]. Various studies show that information retrieval quality tends to decrease in proportion to the recognition error rate.

The German language is known by its compounded structure, according to CLEF evaluation campaigns [42], splitting compound words in German has shown to be effective for IR purposes as the same concept can be expressed using different forms (e.g., *Computersicherheit vs. Sicherheit mit Computern*). Compounding is far less pronounced in MHG with respect to modern German (Section 2.1.5).

As discussed earlier in Section 2.1.5, it is obvious that historical languages can be characterized by their highly flexible grammar, multifold orthographic variations, and minimal strictness on abbreviation rules. Certain older languages lack a great deal of standardization compared to their modern counterparts. Various orthographic forms (all referring to the same entity) may co-exist within the same textual item (e.g., a manuscript). Several approaches have been suggested to deal with spelling variability as for example the plays and poems in Early Modern English (1580 - 1640) [16, 44]. Shakespeare for instance had his name spelled as "Shakper", "Shakspe", "Shaksper", "Shakspere" or "Shakspeare", but never as the currently known spelling.

As for the German language, Ernst-Gerlach and Fuhr [19] performed a rule-based

search method to map historical spelling variants of words to the corresponding modern version and use the former set for expansion either automatically or by interacting with the user. Word pairs of contemporary inflected and derived forms and the corresponding historic variants are manually collected to serve as the training set, which is a time consuming process. The resulting matches were inspected manually and further rules can be fed by the user via an interactive graphical interface. The test-collection used is based on texts from the 15^{th} to the 19^{th} century. However, in this study the authors *"..are not performing retrieval experiments the usual way"* but rather adopted a recall-oriented word-spotting approach since a precision of 1 seems to be quite trivially achieved in this study.

Hauser *et al.* [26] based their experiments on early prints (14^{th} - 17^{th} century) in ENHG, which are numerous (unlike the case of manuscripts). This study suggests building *special dictionaries* of entries of modern and historic variants (with the possibility of incorporating meta-information such as place and source) to be used later for automatic or interactive expansion. Similarly to [19], this approach requires a great deal of manual work by experts in the domain (e.g., linguists) to check the individual entries at multiple stages of the study (corpus generation, handling spelling and compound variation, incorporating morphology, syntax, structure, and meta-information [26]). In addition to their high-demand of manual work, these studies also share their high-dependency on external resources such as thesauri and somehow *"arbitrary"* texts to accumulate possible variants.

Approximate matching is another choice according to Hauser *et al.* [26], either by using word similarity or rule-based matching. Previous studies also suggested various applications based on the Levenshtein distance for approximate name matching, performing IR on languages lacking fixed orthography or improving pseudo-relevance feedback, and correcting automatic mistranscribed spoken documents in [34, 57, 66].

2.3 Summary

In this chapter we presented the classical similarity models and indexing strategies in the Information Retrieval (IR) domain. We also reviewed different approaches

to representing and retrieving noisy documents from the litterature as well as state-of -the-art projects.

The state-of-the-art research is mainly focusing on complementary tools and methods to support or enhance the performance of these models and strategies according to the problem on hand. Overall, we saw that there is no shortage in works and projects done on historical documents retrieval. However, most of these projects involve significant manual work for correction, taggig, normalization, and/or transcription. Some studies suggest approximate and/or rule-based matching to overcome spelling variation experienced in older languages.

The problem of representing and retrieving noisy historical documents is still an unsolved one, and with less resources (budget, thesauri), the challenge grows even bigger. It is thus worth examining the behaviour of the traditional retrieval and indexing approaches to spot and define the points of strength and weakness. In a following step, the methods and tools needed to support them (e.g., translation, spelling correction, enrichment) can be introduced.

The next chapter presents the datasets that we used throughout our experiments and the metrics to quantify the performance obtained using IR systems. Afterwards, we present our experimental setups and results when solely using these traditional approaches. Later chapters present our contributions that aim at enhancing the performance of those approaches.

3

Evaluation

This chapter is divided into two major sections. The first, namely, the *evaluation test-collections*, introduces the test-collections employed in the experimental part of this dissertation. The first test-collection is the result of digitizing a printed text in Modern English using character-based text recognition tools (OCR). The second test-collection is based on a well-known poem called *Parzival*, a manuscript written in Middle High German and digitized using word-based recognition tools.

These two test-collections originate from different providers but share two main features, the first is that both were built for testing IR systems in the presence of text recognition errors. The second is their capability of handling *known-item* search; a common search problem in IR. In this task, the user is looking for a previously seen document (or a few documents) and thus knows it exists but, however, does not know how to access it again. In order to retrieve the target document, the user attempts at recollecting one or more terms seen in that document, builds a topic using these terms and feeds it to the IR system. The expected result is a ranked list of documents that are relevant to the user's request based on their similarity to the query (as calculated by the IR system). However, if the target document is not among the returned documents, the search fails for that query since the user information need is not answered.

The second section, *performance measures*, defines and discusses the measures that are standardly used to assess IR performance when facing with the known-

item paradigm (Mean Reciprocal Rank MRR, and Average Precision AveP). Finally, it introduces our proposed metric that extends the standard MRR.

3.1 Test-collections

The empirical nature of IR requires realistic datasets, these are, however, not widely available. For this reason, standardized test-collections were built by initiatives like, for instance, TREC[1] (Text REtrieval Conference) and CLEF[2] (Conference and Labs of the Evaluation Forum - formerly known as Cross-Language Evaluation Forum). These initiatives offer different tasks and test-collections to support research in IR. The alternative (yet expensive and resource-demanding) approach is to build test-collections that comply with the standards yet meet the specific research needs on hand.

Up to this point, we can summarize the characteristics of the test-collections required for our experiments in the following checklist:

(a) **Search problem:** The test-collection should consist of the standard building elements necessary for performing known-item search: the corpus, the query set, and the relevance judgments.

(b) **Parallel versions:** There should be at least two parallel versions of the corpus: an error-free version to be used as the assessment baseline in addition to (at least) one *corrupted* version *suffering* from imperfect text recognition.

(c) **Language irregularities:** The documents should be based on texts written in a non-normalized old language, one in which orthography and/or grammar were not standardized.

During the course of writing this dissertation, we were introduced to the CLEF2012[3] Cultural Heritage track [3], namely, CHiC[4]: Cultural Heritage (CH) in CLEF (and

[1] http://trec.nist.gov/
[2] http://www.clef-initiative.eu/
[3] http://clef2012.org/
[4] http://www.promise-noe.eu/chic-2012/home

then with CHiC2013[1] in later stages of writing this dissertation). The CHiC corpora are extracted from Europeana[2]. This digital library provides access to over 23 million searchable items referring to books, paintings, manuscripts, museum objects, plays, notable characters in history, etc.. Despite their richness, multilingualism and being tightly bound to CH, these Europeana-based datasets still do not represent the test-collections we need. The reason is that the textual contents of the documents are provided in modern languages such as English, French and German, which, most of the time, only describe the items in question thus providing the related *meta-data*. For instance, a search with query terms extracted from the actual contents of the already mentioned manuscript *Parzival* will fail since the verses within the manuscript were not made available in the Europeana indexes. Focusing mainly on meta-data, this collection lacks all three characteristics in our checklist.

Succeeding two-thirds of our checklist, the TREC-4 and TREC-5 confusion tracks [30, 62] constitute a good starting point. However, with documents written in Modern English, the *language irregularities* are obviously missing. The next sections present these campaigns in more detail.

3.1.1 The TREC Confusion Tracks

Initially named the *corruption track*, the TREC confusion track originated at an informal meeting held during TREC-3 [29], the idea was to adopt various schemes for retrieval based on OCR applied on *scanned legacy text*. The range of methods included the use of overlapping n-grams with varying lengths, the transition matrix of this *corruption* algorithm was known only to the TREC organizers but not to the participants [29].

The experiments conducted during the **TREC-4 confusion track**, however, are not particularly realistic since the documents were randomly corrupted by NIST (the National Institute of Standards and Technology) without providing a clear description of error sources; random character deletions, insertions, and

[1] http://www.promise-noe.eu/chic-2013/home
[2] http://www.europeana.eu/

substitutions were introduced. The distortion was applied to a certain percentage of the characters (either 10% or 20%) such a distortion model, however, does not reflect our goals due to its lack of fidelity in simulating real-life OCR applications which might consequently yield unreliable evaluation results. Obviously, TREC-5 confusion track is the most suitable standard test-collection for our research interests. The paragraphs below provide a brief description of this test-collection while the results we obtained are presented in Chapter 4.

The **TREC-5 confusion track** shares the same structure as other standardized test-collections; a corpus, a set of topics that represent the user information needs, and a relevance judgment file. The topics are searched against the corpus for which answers are returned in the form of a ranked list of documents.

The relevance judgments file is essential to the evaluation process. For each topic, it states the relevant documents, typically, according to prior assessments of these results by users (or judges). For the case of known-item search, it identifies the particular document(s) that the user wishes to find. In other words, the previously-seen documents that s/he is trying to retrieve. The three building components of a standard test-collection are presented quite detailedly below:

(a) **The corpus** Each of the three versions of the TREC-5 corpus consists of 55,630 documents of which 125 entries are empty documents. The baseline corpus was extracted from the Federal Register (year 1994); a daily publication of regulations and other information from U.S. federal agencies and organizations. In average, each document contains 414.05 indexing terms (with an average of 202.5 distinct terms). Figure 3.1 shows an excerpt from one of these documents, where each document begins with the tag <DOC> followed by a tag <DOCNO> which specifies a unique document identifier, and a <TEXT> tag indicating the document text. As these short samples clearly demonstrate, despite the 5% recognition error rate the text is still readable. While with an error rate of 20%, text comprehension becomes a lot more difficult; rather than naturally reading the text, one needs to decrypt it. This might give an impression of some of the

representational and retrieval-related challenges encountered when dealing with significantly noisy text.

(b) **The topics** The TREC-5 confusion track features a set of 50 topics numbered 1 through 50 that is provided together with the documents. Topic #29 lacks a correct answer (i.e., none of the documents in the corpus is *officially* deemed relevant to this topic); therefore, it was eliminated during the evaluation and thus only the remaining 49 queries should be used in experiments. These topics cover a variety of subjects, such as economics and finance (Topic #7 "Excessive mark up of zero coupon treasury bonds"), politics (Topic #9 "Gold watches accepted as gifts to the US government."), science and technology (Topic #47 "What areas of the human body are the main targets for lead?" and Topic #10 "Patent applications for nonlinear neural network oscillators"). It is worth noting that some queries are limited to only one word (Topic #35 "Bugle") or two words (Topic #34 "saucer bounce" or #39 "jellyfish abundance"). Figure 3.2 shows three sample queries of the collection. The queries are structured as follows: each query begins with a <NUM> tag followed by a unique identifier for that query, a description tag <DESC> comes next preceding the textual content of that query.

(c) **The relevance judgement file** The judgments here are binary, that is, for a given query any document in the collection is either relevant (1) or non-relevant (0).

As can be seen, there are not, unfortunately, any standardized collections based on historical documents, nor are there ones that resemble the TREC-5 confusion track but written in languages other than (Modern) English. We have thus created from scratch a dataset with parallel *clean* and *corrupted* versions as well as query sets of different specifications together with the relevance assessment files. This in-house corpus is based on the earlier-mentioned medieval manuscript *Parzival* which was written in Middle High German and digitized via a word-based text recognition tool, this dataset is the focus of the next sections.

<DOC>
<DOCNO>FR940317-1-00199
<TEXT>
In withdrawing the riskless principal mark-up disclosure proposal in the 1978 Release, the Commission stated that it would "maintain close scrutiny to prevent excessive mark-ups and take enforcement action where appropriate. "Since 1982, the Commission and NASD have undertaken a number of enforcement actions against broker-dealers involving ...

<DOC>
<DOCNO>FR940317-1-00199
<TEXT>
ln withdrawlng the risyless principal mary-up disclosure proposal in the 191W helease1 the Commission stated that it would 44maintain close scrutiny to prevent excessive mary-ups and taye enforcement action where appropriate.:: 20 Since 19W21 the Commission and NASL have undertayen a number of enforcement actions against broyer-dealers involving ...

<DOC>
<DOCNO>FR940317-1-00199
<TEXT>
fa -thtlrawing the WfUefqs priucipA mary-up dRclosure proposA in the 191@ M,lease, the ComMssioa stated that it would amUntdn close scrutAy to preveat excessive m=y-upqe at nd tttes eaforcemebt actioa where approphate.. 2e 0 Since 19S2, tlte CommLsion and htASO have .ndertayea a a0mber of eaforc,ement actioaf agUnst broyer-de,Uers involUng ...

Figure 3.1: A paragraph excerpt from TREC-5 in three versions: the original clean version on the top, the 5% recognition error rate in the middle, and the 20% error rate at the bottom

<NUM> CF2
<DESC>
Mutual fund risk disclosure for futures related issues.

<NUM> CF5
<DESC>
National Science Programs in radio astronomy.

<NUM> CF30
<DESC>
characterization and symptoms of the milk-alkali syndrome.

Figure 3.2: Three sample Topics of TREC-5

3.1.2 Historical Manuscripts (The Parzival Test-collection)

This test-collection is based on a well-known medieval epic poem called Parzival. This poem is attributed to Wolfram von Eschenbach. The first version dates to the first quarter of the 13^{th} century and was written by several writers in the Middle High German language with ink on parchment using Gothic minuscules. Currently, several versions (with variations) can be found but the St. Gall collegiate library cod. 857 is the one used for experimental evaluation [22]. An excerpt of the manuscript is shown in Figure 3.3. An error-free transcription of the poem was created manually and made available by the experts (Figure 3.4).

This error-free version forms our ground-truth (GT) text and was used to assess the performance levels throughout our experiments. To generate the corrupted parallel versions of the corpus, each page of the manuscript has been scanned and automatically subdivided into smaller images each corresponding to a single line (verse) of the poem. Following medievalists' tradition, each verse (line) of the poem represents a document in the corpus. These images were then processed for recognition. At this level, the recognition system determines the best set of possible recognitions for each word. The recognition system provides a set of seven

possible word recognitions. Within each set, the seven alternatives are graded and sorted in terms of their likelihood to be correct. For each word, the seven resulting *recognition#likelihood* pairs are stored as $[W_1\#L_1,\ W_2\#L_2,\ \ldots,\ W_7\#L_7]$ where W_1 is the most likely word whose corresponding grade is L_1 representing the highest log-likelihood value within its 7-recognition (7-r) set.

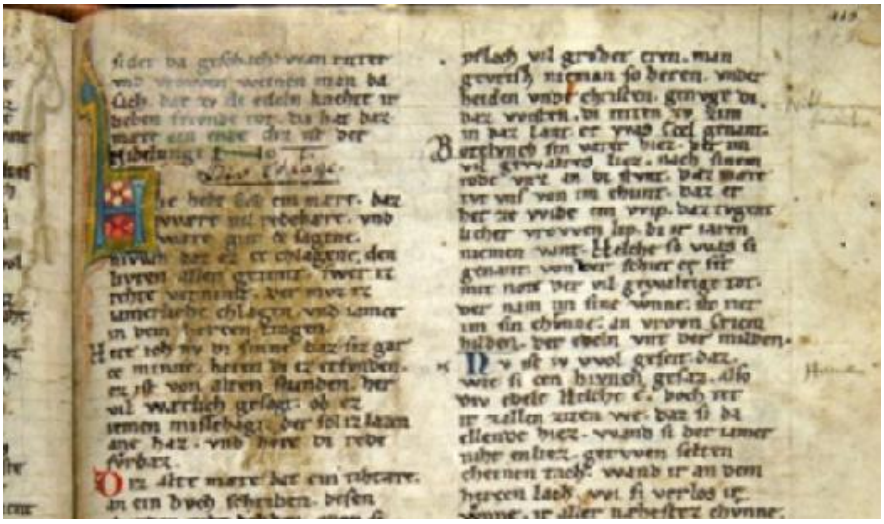

Figure 3.3: A small excerpt of the Parzival manuscript

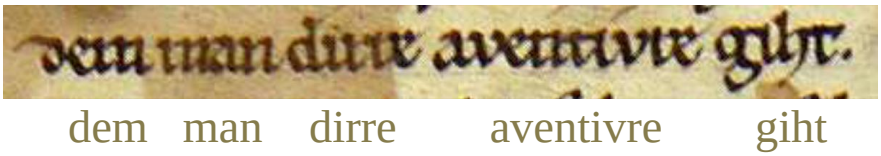

Figure 3.4: A sample verse from Parzival with its error-free transcription (ignoring punctuation)

As a concrete example, we can inspect the word *"man"* in the verse *"dem man dirre aventivre giht"* for which the recognition pairs are: *["man"#36006.78, "min"#35656.51, "mat"#35452.53, "nam"#35424.69, "arm"#35296.25, "nimt"#35278.23, "gan"#35265.75].* In this case, the system succeeded to recognize this occurrence of the word *"man"* correctly, obviously as it is appearing in the first position within its respective 7-r set. Table 3.1 shows the complete

Table 3.1: Sample Parzival verse with word recognition#log-likelihood pairs

dem		**man**		**dirre**		**aventivre**		**giht**	
dem	36825.68	*man*	36006.78	*dine*	39858.95	*aventivre*	81463.72	*giht*	38721.64
zein	36585.68	*min*	35656.51	*dirre*	39849.29	*aventiure*	81067.47	*gibt*	38345.91
zem	36541.31	*mat*	35452.53	*chrîe*	39771.29	*daventivre*	80951.07	*gaher*	38026.35
dan	36493.48	*nam*	35424.69	*dirz*	39606.98	*Aventivre*	80846.73	*gaheṣ*	37713.65
den	36469.12	*arm*	35296.25	*dane*	39586.98	*minnete*	80192.64	*glaṣt*	37605.4
gein	36402.84	*nimt*	35278.23	*Amis*	39494.81	*creatvre*	79974.09	*gehert*	37547.32
win	36389.51	*gan*	35265.75	*dîner*	39448.86	*ṣwennez*	79926.76	*gir*	37539.32

7-r sets for the entire verse above.

The manuscripts were transcribed to the digital format using a *Hidden Markov Model* whose basic features are described in [22]. This recognition system is based on the closed vocabulary assumption implying that each word in the test set is known or has already appeared in the training set. This system has evolved in terms of performance and achieved a word-accuracy close to 94%. Thus, the produced transcription has a word-error rate of around 6%.

The accuracy of the system is calculated based on the resulting W_1 for each word: the system succeeds if W_1 is the correct word. Otherwise, the system misses if the correct word is W_2 through W_7 or even non-appearing in the 7-r set at all, and thus falling within the 6% error rate.

3.1.2.1 Corpora Generation

The traditional output of a text recognition system is the sequence of *W1*'s for each word. This corresponds to the first corrupted version of the Parzival corpus. We will refer to this version as *BW1* (**B**est first **W**ord). Conventionally, what remains from the 7-r set is a by-product of the recognition process and is always discarded. We decided to think outside the box by experimenting with this by-product and making use of these sets to build further corpus surrogates. By examining Table 3.1, we can see that the token *"dirre"* was mistranscribed. For this word, the second likely recognition (*W2*) corresponds to the correct transcription word, whereas *W1* corresponds to an incorrect recognition. With this and similar situations in mind and considering that each document (verse) is quite short, the

existence of a recognition error will make the retrieval of the corresponding verse an unattainable task without some sort of spelling normalization or approximate matching between the search keywords and the text representation. Therefore, we generated three additional corpora denoted *BW3*, *BW7* and *BWδ*.

Instead of being limited to only the most likely recognition per word (token), in *BW3*, the three most likely recognitions W_1, W_2 and W_3 are retained.

Similarly, we created our third corpus surrogate, *BW7* in which -as the name tells- the entire 7-r set is present. This version corresponds to the highest intensity of spelling (and recognition) variants.

It is worth-mentioning that the 7-r sets for a given token are not unique over the different occurrences of that token in the different verses. This means that the sets may contain different as well as similar word classes with different likelihood scores (and hence rankings) within each occurrence. Referring to our example, the 7-r set of the first occurrence of the word *"man"* is *["man" #36006.78, "min" #35656.51, "mat" #35452.53, "nam" #35424.69, "arm" #35296.25, "nimt" #35278.23, "gan" #35265.75]*, while another appearance of *"man"* yielded the following 7-r set *["man" #44254.71, "nam" #43964.23, "nimt" #43854.35, "manz" #43828.35, "min" #43694.35, "mat" #43661.38, "mann" #43660.27]*.

However, having as many word alternatives per a token occurrence can be quite risky due to possible semantic and orthographic shifts in the less likely words per set. A wiser approach is thus needed to balance the restrictiveness of *BW1* and the unconditionality of *BW7*. In *BWδ*, recognition alternatives are included as long as the difference between the candidate's log-likelihood value L_n and that of the most likely term L_1 is less than or equal to δ (where $n = 2, 3, 4, 5, 6, 7$ and δ is set to 1.5% in our experiments). Using our previous example, only the alternative *"min"* is present in addition to the term *"man"* in the *BWδ* version for that occurrence of the term.

The benefit sought from incorporating more than one recognition alternative is to overcome the word recognition errors as well as the non-normalized spelling. For instance, the term *"Parzival"* appeared in the original manuscript as *"Parcifal"*,

"Parcival" and *"Parzifal"*. All of these variants are possible and must be considered as correct spellings. During the recognition phase, some of these variants may appear in the 7-r sets and using the BW3, BW7 or BWδ corpora, some or all of them can be retained in the text representation.

At this lexical level, one should also consider the inflectional morphology where various suffixes were possibly added to nouns, adjectives and names to indicate their grammatical case (e.g., as in Latin and Russian). With this additional aspect, the proper name *"Parcival"* may appear as *"Parcivale"*, *"Parcivals"*, or *"Parcivalen"*, increasing the number of potential correct spelling variants.

The corpus used in our IR experiments is comprised of 1,328 documents, this corresponds to only a subset of the complete Parzival transcription. The remaining 4,477 verses form the training set used during the recognition phase and thus are not included in the search evaluation. The number of tokens per verse ranges between 2 and 9 with a mean length equal to 5.3 words.

3.1.2.2 Automatic Known-item Topic Generation

The manual construction of user queries together with their relevance judgments is another expensive task. A cheaper alternative is to generate them automatically. This issue had been the subject of many studies in order to obtain comparable quality between the automatically generated queries and those built by real users [6, 11, 28]. In the context of simulated query building and known-item search, we have adopted the approach suggested by Azzopardi & de Rijke [6]. This approach is based on a probabilistic framework (see Figure 3.5) simulating the behavior of a user who wants to retrieve a previously seen document, trying to aggregate terms that s/he recollects from this target document. In adopting this algorithm, we excluded all short words (whose length is less than four characters) from being potential search terms. Moreover, words belonging to the list of the 146 non-content-bearing terms in the corpus were eliminated automatically. Finally, we generated three sets of 60 queries each, these are denoted QT1, QT2, and QT3 and respectively contain single-, 2- and 3-term queries. To define the probability of selecting the document d_k (Prob$[d_k]$), we used a uniform distribution. For

choosing the query term t_i given the query θ_d (Prob$[t_i|\theta_d]$), each word has a chance proportional to its length (in characters), the longer the term, the higher the probability of it being selected. This random process was used to generate the QT1 set.

For longer queries, we decided to augment the source of the search keywords. A classical user search need is to retrieve a certain verse or passage that is known to her/him. Thus s/he would build a query by combining a few words that s/he believes had previously seen in that verse. Knowing that the user usually reads a poem as a sequence of verses rather than just a single line at a time, it is possible that s/he confuses words from adjacent verses. Hence, for a multi-term query, retrieving the line that immediately precedes or follows the target verse should not be regarded as non-relevant, but one should also consider that these adjacent verses do not carry the same degree of relevance/importance to the user's query compared to that of the target verse Accordingly, the underlying language model for generating multi-term queries would consider the verse defining the known-item itself, and possibly the preceding and the following lines. In this random process, the verses were not assigned the same probability since the words occurring in the central verse were given twice the chance of being selected.

As a final modification, when generating the QT3 set, the third query term had the possibility to originate from the short words list and the 146 frequent terms. This makes it possible to obtain nominal and prepositional phrases by having two information-bearing terms combined with a preposition.

```
Initialize an empty query Q = {}
Select the document d_k to be the known-item with probability Prob[d_k]
Select the query length s with probability Prob[s]
Repeat s times {
   Select a term t_i from the document model of d_k with probability Prob[t_i|θ_d].
   Add t_i to the query Q. }
Record d_k and Q to define the (known-item / query) pair.
```

Figure 3.5: The basic known-item query generation algorithm according to Azzopardi & De Rijke [6]

3.2 Performance Measures

When retrieving a set of documents, the precision and recall measures are well defined but tend to vary in opposite directions. For the task of evaluating a ranked list of retrieved items, different performance metrics have been suggested, based on the precision, the recall and the rank of the relevant retrieved items.

For the *ad hoc* task, the Average Precision (AveP) is used to compare the retrieval effectiveness obtained from applying different indexing and search strategies. The user model behind this evaluation corresponds to a searcher interested in retrieving all relevant items. Such a model reflects the search behavior of, for example, a lawyer who wants to find all court decisions that are somehow related to a certain legal issue, or a user trying to verify the possible validity or coverage of a submitted patent proposal.

To measure the retrieval effectiveness in the presence of noisy text, the TREC evaluation campaign chose the MRR (Mean Reciprocal Rank). Such a measure adequately reflects the concerns of users wishing to find one or (a few) pertinent response(s) to any given request. Section 3.2.1 presents the MRR measure in detail while Sections 3.2.2 and 3.2.3 propose two measures that extend the classical MRR measure.

3.2.1 Mean Reciprocal Rank (MRR)

For a given query, the reciprocal Rank (RR) is equal to the inverse of the rank of the earliest (i.e., arriving at the highest rank) relevant item retrieved for that query [10].

As an example, for a given query if the first known-item d arrived at rank 2, then the RR for that query is equal to $1/Rank = 1/2 = 0.5$. Any other pertinent items arriving at later ranks are ignored and thus not taken into consideration of this calculation. The values of RR range between 0 and 1, where 1 corresponds to the ultimate RR since the known-item appeared at the first rank. On the other hand, a RR equal to zero means that the system failed to find any known-items (usually not found in the top 1000 retrieved documents).

For multiple queries, the mean of the reciprocal rank values over all queries should be calculated (MRR). This measure reflects the concern of a user interested in finding one or a few pertinent documents to her/his request.

3.2.2 Graded Mean Reciprocal Rank (GMRR)

The concept of partial relevance introduces a more complicated situation; that is, having items that are neither fully relevant nor non-relevant at all. In homepage search for example, the target homepage is the fully relevant item. A hub page with a hyperlink to that homepage should not be considered as fully-relevant. However, such a page is informative enough not to be deemed non-relevant either (since it leads directly to the target homepage). A similar situation occurs when searching a certain XML node extracted from a structured document. If the returned answer resides within the paragraph that for instance precedes the target one, we might consider this answer to be partially relevant since the end-user can see the target passage right below it. Finally, a partially relevant item may correspond to content that partially fulfills the user's information need.

In the NTCIR[1] Web track evaluation, the *rigid RR* measure was used which considers non-fully relevant documents as totally non-relevant. This approach is very strict since non-fully relevant documents can still have some informational value for the end-user. Excluding these from the relevance circle is severe, and, in our opinion, is not quite fair, this would consequently result in a lower MRR value. On the other hand, by adopting the *lenient* evaluation, the *relaxed RR* will yield unrealistically high MRR values since it conflates the fully relevant as well as less relevant documents into the same set.

In practice, one would aim to evaluate an IR system by comparing the performance over a set of queries each having a set of pertinent items. It is possible that some requests have only one (or very few) document(s) judged to be fully relevant and the rest of the information items are simply considered as non-relevant. This is the classical binary relevance case. For other requests, relevance can be of different degrees (full relevance *vs.* high or partial and marginal relevance, etc.).

[1] http://research.nii.ac.jp/ntcir/

For the former set of requests, the classical binary RR measure would perfectly fulfill the evaluation job. On the other hand, RR would be of limited use for the latter subset since it does not handle graded relevance.

Several studies focused on the importance of and the need for graded relevance measures [31, 50, 52, 63], after decades of evaluation based on binary relevance [52]. Despite the existence of a number of IR metrics supporting graded relevance [13, 18, 50, 51], none of them has been as widely used as traditional binary relevance metrics, furthermore, none of them is fully compatible with MRR. This means that results obtained from MRR cannot be compared to those obtained from any of these metrics. This can be problematic if the underlying test-collection requires both binary and graded relevance with the necessity of using MRR (which is the case for our Parzival test-collection).

This section presents a measure that extends the traditional MRR and is capable of handling graded (n-ary) as well as binary relevance. We refer to this measure as Graded Reciprocal Rank, or simply (GRR) for a single query. When computing the average of GRR values over a number of queries the mean of GRR values should be calculated or Mean GRR. We prefer to stylize the name as GMRR. It is shown later in this section that the classical RR as well as the *rigid* and *relaxed* RR are special cases of the GRR measure. Throughout our numerous experiments, GMRR showed to provide reasonable and well-balanced results compared to the extreme *rigid* and *relaxed* RR measures. GRR does not overlook non-fully relevant documents yet eliminates the unwanted tolerance of the *relaxed* RR. It incorporates partially relevant information items into the evaluation but limits their contribution proportionally to their associated degree of relevance. According to the number of relevance levels and the primary setting of the respective coefficients, GMRR may sometimes favor an item with higher relevance despite its appearance in a later rank than that of a less relevant one.

For a given query, and using graded relevance, the search could be unsuccessful. In this case, the evaluation of that query is 0 (in both RR and GRR). Second, when a fully relevant item appears before any partially relevant items in the ranked list, the query is evaluated based on the reciprocal rank of that fully relevant item.

In these first two cases, the results returned using the GRR are similar to those produced when using the classical RR.

In a third possible situation, a non-fully relevant item may appear in a much better rank than that of a fully relevant one. In this case, the GRR measure evaluates the query based on:

(a) the rank of that non-fully relevant item

(b) the degree of relevance of that item

(c) the rank of the earliest fully relevant item

More formally, for a known-item search, we can assign different ordered degrees of relevance for the items as, for example, *fully relevant*, *moderately relevant*, *weakly relevant*, and of course *non-relevant*. We denote the number of possible relevance levels by k. In the classical binary case, k is equal to 2 (e.g., an item is either relevant or non-relevant). In general, k could be any positive integer greater than or equal to 2, such as $k = 4$ in our previous example.

Based on such user-defined ordinal graded relevance scale, the suggested GRR measure for a query q is defined as:

$$GRR(q) = \begin{cases} Max[\alpha_i \cdot 1/\mathrm{R}_i] & i = 1, ..., k-1 \\ \\ 0 & otherwise(i = k) \end{cases} \tag{3.1}$$

Where k is the number of levels corresponding to the predefined degrees of relevance, and α_i is a coefficient reflecting the degree of relevance for the i^{th} level (or class). Moreover, for each relevance level i = 1, 2, ..., k, we can identify the retrieved item belonging to this level and having the best rank (denoted R_i) compared to other retrieved items of the same level i.

The values associated with the different coefficients α_i must respect the following criteria. For the fully relevant class denoted with $i = 1$, the underlying coefficient α_1 is fixed to 1. The last class corresponding to the non-relevant items, i.e., when

$i = k$, the coefficient α_k is fixed to 0. For the remaining levels, the constraint that $\alpha_i \geq \alpha_{i+1}$ for all i = 1, ..., k-1 should be met. The ordered sequence of coefficients α_i forms a monotone non-increasing sequence.

We can directly see that the classical binary RR is a special case of GRR having k = 2, i.e. two levels of relevance. According to our rules, the coefficients are fixed as $\alpha_1 = 1$, and $\alpha_2 = 0$. If the first relevant retrieved item appears in rank 4 (R_1 = 4), the GRR value is Max $[1 \cdot 1/4] = 0.25$. Which is identical to that obtained from the classical RR measure. When facing with more than two levels, say three, thus $k = 3$. Having the coefficients α_i set as follows: $\alpha_1 = 1$, $\alpha_2 = 0$, and α_3 = 0. In this case, the resulting values using GRR would be identical to those obtained when adopting the *rigid* assessments in the NTCIR Web track where the partially relevant items are considered as non-relevant. The *relaxed* RR is obtained by setting $\alpha_2 = 1$ (*ceteris paribus*) which has the impact of considering all the fully relevant and the partially relevant items to belong to the same level despite their different actual degrees of relevance. We hence concluded that both the *rigid* and *relaxed* RR measures are also special cases of GRR.

Let us now consider a more complicated case, a search task with $k = 5$ which corresponds to five degrees of relevance. Each retrieved item could thus be judged as either fully relevant, highly relevant, moderately relevant, weakly relevant or non-relevant. To reflect the difference in quality or pertinence, we set the coefficients as $\alpha_2 = 0.7$, $\alpha_3 = 0.4$ and $\alpha_4 = 0.2$, and as mentioned before $\alpha_1 = 1$ and $\alpha_5 = 0$. Suppose for a given query q, the earliest appearance of a fully relevant item in the ranked list was at Rank 12, thus $R_1 = 12$. For the highly relevant class, the best rank obtained was ten, $R_2 = 10$. While the earliest moderately relevant item arrived at rank $R_3 = 5$, and the best achievement ranking-wise for the weakly relevant items was $R_4 = 50$. The GRR evaluation of this query q is:

GRR(q) = Max [$1 \cdot 1/12$; $0.7 \cdot 1/10$; $0.4 \cdot 1/5$; $0.2 \cdot 1/50$]
= Max [0.083; 0.07; 0.08; 0.004] = 0.083

The traditional *relaxed* RR would have been based on $R_3 = 5$ due to its being the earliest appearance of a relevant item (regardless the degree of relevance).

Despite the fact that the first fully relevant item appeared in a rank later than R_2=10 and R_3=5, it is still not late enough to be overlooked. Conversely, R_2 = 10 was a bit too late and thus the earliest-appearing moderately relevant item coming at Rank 5 has defeated that highly relevant item.

The number of levels representing different degrees of relevance can be set according to the requirements of the experiment being carried out. But one should keep in mind that adding too many levels may cause the discrimination to fade out between closely ranked documents having close degrees of relevance due to the decreased differences between the coefficients' values. Moreover, it is known that human-beings are unable to differentiate too many ordered classes [38] (e.g., judging satisfaction levels as users, deciding difficulty scales, etc.).

Ternary GRR, i.e., $k = 3$, is one of the interesting cases. Working with the Parzival test-collection, we evaluated our multi-term queries using this setting of GRR: fully relevant (the target verse), relevant (any of the two adjacent verses of the target verse) and non-relevant. In our experiments, we deemed setting α_2 = 0.5 would be adequate. For example, having $R_1 = 3$ and $R_2 = 2$, the query is evaluated as Max[$1 \cdot 1/3$; $0.5 \cdot 1/2$] = 1/3. The evaluation is thus based on the rank of the fully relevant document. With $R_1 = 5$ and $R_2 = 2$, the query evaluation is Max[$1 \cdot 1/5$; $0.5 \cdot 1/2$] = 0.25, in this case, the query evaluation depends on the partially relevant item appearing in the second rank.

3.2.3 Best-Rank-Oriented GMRR (broGMRR)

Best-Rank-Oriented Mean Reciprocal Rank is a variation of GMRR. With broGRR, a query q can be evaluated by firstly selecting the item with the best rank (i.e., the one of the smallest (positive) magnitude). With this rank and the respective degree of relevance of that item, we can compute the corresponding RR weighted using the appropriate α_i coefficient. The principle behind this approach is somewhat similar to that of the WRR measure [18].

$$broGRR(q) = \begin{cases} \alpha_i \cdot \text{Max}[1/\text{R}_i] & i = 1, ..., k-1 \\ 0 & otherwise(i = k) \end{cases} \tag{3.2}$$

For the same previous example with $\text{R}_1 = 12$, $\text{R}_2 = 10$, $\text{R}_3 = 5$ and $\text{R}_4 = 50$, $\alpha_1 = 1$, $\alpha_2 = 0.7$, $\alpha_3 = 0.4$, $\alpha_4 = 0.2$ and $\alpha_5 = 0$, broGRR(Q) is evaluated as:

$\text{Max}[1/\text{R}_1; \quad 1/\text{R}_2; \quad 1/\text{R}_3; \quad 1/\text{R}_4]$
$= \text{Max}[1/12; \quad 1/10; \quad 1/5; \quad 1/50]$
$= \text{Max}[0.083; \quad 0.1; \quad 0.2; \quad 0.02] = 0.2$
$\text{broGRR(Q)} = \alpha_3 \cdot 1/\text{R}_3 = 0.4 \cdot 0.2 = 0.08.$

3.3 Summary

Empirical datasets with certain standards are mandatory for evaluating Information Retrieval (IR) systems. Within such datasets, the user's information needs as well as the good answers to these needs (represented by the pertinent documents) are priorly chosen and reported in relevance judgments. The quality of performance of an IR system is then reflected by its capability of finding these good answers and reporting them in early ranks when the same query is fed to the system being tested for performance.

In this chapter, we presented the TREC5 confusion track dataset as well as our Parzival dataset. Both datasets are based on scanned documents that have been digitized using imperfect text recognition tools. Albeit, these datasets vary greatly in terms of the language, length, writing method and other characteresitics as discussed in Section 3.1.

The TREC-5 confusion track chose the Mean Reciprocal Rank (MRR) as its evaluation measure. Through this measure, however, the documents are seen in black and white, that is, they are considered either fully relevant or non-relevant at all. We proposed two extensions to this measure (namely, GMRR and broGMRR

short for Graded MRR and best-rank-oriented GMRR respectively) which introduce a gray scale to the evaluation image. These extensions allow graded relevance levels such as moderately and weakly relevant in addition to the standard fully- and non-relevant levels.

4

Retrieval of Digitized Printed Modern English Texts

(Character-based Text Recognition)

This chapter presents our experiments using the TREC-5 confusion track test-collection. This standard test-collection is based on printed texts written in Modern English. Standard test-collections are created for the purpose of providing a reliable benchmark of IR systems. In Chapter 3 we presented a detailed checklist specifying the three characteristics that we look for in a test-collection as far as noisy text retrieval is concerned. These are (briefly):

(a) Support for the known-item search paradigm.

(b) The availability of an error-free version of the text in addition to one (or more) realistically noisy/corrupted yet parallel version(s) .

(c) The presence of orthographic and lexical variations as a characteristic of the old language in which the text is written.

Of all the available standard test-collections, the TREC-5 confusion track constitutes the best match (practically, the only match) to this checklist (discussed more detailedly in Section 3.1.1), though lacking the third requirement it consists of an error-free corpus provided alongside two parallel corrupted version of 5%

and 20% recognition error rates. It is worth reminding here that no other similar test-collections are available; neither ones in a language other than English nor ones written in older languages.

The rest of this chapter is organized as follows: The first section provides a detailed overview of the experimental setup for the entire series of experiments that we conducted using the TREC-5 confusion track test-collection. Section 4.2 discusses the results obtained from these experiments when using the various techniques. A selection of interesting cases are then discussed in Section 4.4. These cases are presented in the form of individual queries and their respective retrieval responses. We have thoroughly analyzed these queries and their results with the aim of depicting the strengths and weaknesses of the various IR models and strategies in the presence of recognition error in different rates (0% or error-free, 5%, and 20%). Finally, Section 4.3 evaluates the impact of pseudo-relevance feedback to, hopefully, overcome the issues associated with searching noisy OCR'd texts.

4.1 Experimental Setup

The combination of various document representations (Section 2.1.1) and retrieval models (Section 2.1.2) provides a broad view of the possible level of retrieval performance [36]. This proceeding is especially favorable given the fact that no clear conclusions seem to be possible when handling noisy texts as it can be seen in Chapter 2, Section 2.2. Taghva *et al.* [58] for instance, suggest ignoring all stemmers, or at least applying only a light stemmer [25]. Meanwhile, the best-performing TREC-5 system [7] opted for an aggressive stemmer [45]. Bearing in mind that recognition errors mostly affect short words (i.e., ones consisting of two or three characters), the automatic elimination of these short words can be beneficial [59].

As a first representation of documents and topics, we adopted the bag-of-words model. After segmenting the text into word tokens, the 571 non-content-bearing and most frequent words are automatically eliminated (the SMART stoplist). As a second approach, we applied three stemming variations in order to process suffixes in the collection terms, namely: no stemming, a light stemmer based on the Harman's algorithm [25] for deleting the final "-s" denoting the plural form in English (Figure 1, Appendix B), and finally, the more aggressive stemming

algorithm such as the one proposed by Porter [45], which tries to remove some derivational suffixes as well [1].

The stemming procedure is not error-free and thus choosing the more aggressive strategy involves the risk of conflating under the same stem those words having different meanings (e.g., "organization" becomes "organ"), and ignoring some pertinent conflations (e.g., "merger" and "merging" would not be unified under the same form).

Alternatively, terms could be represented by overlapping sequences of n letters, this representation is known as the character-based *n-gram*. When setting $n = 4$ for instance, the word "computing" generates the indexing terms "comp", "ompu", "mput", "puti", "utin" and "ting". When adopting this representation strategy the stemming procedure can thus be ignored, letting the weighting scheme assign low weights to very frequent sequences (e.g., "ting", or "ably"). In our experiments, we set the value of $n = 4$, a value that resulted in the best performance compared to $n = 3$ and $n = 5$, other studies also showed that setting $n = 4$ usually yields better performance.

As an alternative, we suggest considering only the first n letters of each word, better known as *Trunc-n*. When setting $n = 4$, the word "computing" (or "computer") would generate the single indexing term "comp". Knowing that a significant word could appear several times [14], the likelihood of no matches will be reduced. With a high recognition error rate, the chance for a long word being altered is high. Working with a noisy corpus whose error rate is as high as 20%, the probability that a sequence of three letters would be preserved is equal to $(1 - 0.2)^3 = 0.512$ and only $(1 - 0.2)^4 = 0.41$ for a sequence of four letters. This calculation thus demonstrates that a low value of n (three or less) would be adequate. Meanwhile, a lower value of n (e.g., $n = 2$) may induce an excessive number of semantically incorrect matches between the n letters of the search terms and those appearing in the document surrogates of the corpus.

As IR models, we selected two vector-space schemes (*Lnu-ltu* and *tf·idf*), two probabilistic models (*Okapi*, *DFR-PL2*, *DFR-I(n_e)B2*) and a language model (*LM*) (see Section 2.1.2 for more details on IR models).

[1] http://tartarus.org/~martin/PorterStemmer/def.txt

4.2 Evaluation

To assess the performance levels obtained throughout our experiments, the TREC-5 campaign opted for the mean reciprocal rank (MRR) measure, and so did we. As explained in the previous chapter (Section 3.2.1), the MRR measure reflects the behavior of users who want only one or very few pertinent responses to their information needs, such a measure fits perfectly to the task on hand, that is, the known-item search. We also made it clear that the MRR measure is a special case of our extended measure, namely, the GMRR. Since the test-collection on hand does not support graded relevance, GMRR, broGMRR and the classical MRR variations will produce identical results. However, we will use MRR as a *pars pro toto* for GMRR in this chapter for conventionality-related reasons.

Using the same performance measure and having the same test-collection, our performance levels can thus be compared with those obtained during the TREC-5 campaign. To determine whether the difference between any two models is statistically significant, we opted for a bootstrap non-parametric test [54] (significance level $\alpha = 5\%$).

First of all, we evaluated various strategies based on the clean (error-free) version. Section 4.2.1 presents the performance achieved by six different retrieval models based on three different stemming strategies, using three different text representations. Based on these results, we evaluated and analyzed the impact of a recognition error rate varying from 5% to 20% which are presented in Section 4.2.2.

4.2.1 Evaluation of the Clean Corpus

Since our evaluation measure focuses on high precision, we deemed a bag-of-words representation without any morphological normalization (performance shown under the label "None" in Table 4.1 would be adequate. Alternatively, a light stemmer [25] should also provide comparable or even better performance levels since it simply removes plural forms ending with "-s". Finally, the elimination of some derivational suffixes, as it is the norm in many IR empirical studies, will be shown under the label "Porter" [45] in Table 4.1. As an alternative to the bag-of-words model, we implemented the n-gram model with a value of $n = 4$. Another option was to apply the Trunc-n scheme with $n = 4$, a strategy found

Table 4.1: MRR of six IR models, three representations and three stemmers (49 queries)

	Word-based				
IR Model	None	Light	Porter	4-gram	Trunc-4
Okapi	0.7620	0.8231	0.8513	0.7730*	0.7826
DFR-PL2	0.7238	0.7458	0.7936	0.6500*	0.6353
DFR-I(n_e)B2	**0.7958**	**0.8343**	0.8737	**0.8240**	0.8126
LM ($\lambda = 0.35$)	0.7893	0.8077	**0.8807**	0.7029*	**0.8209**
Lnu-ltu	0.6776	0.7352	0.7703	0.5913*	0.6921
tf·idf	0.3775*	0.4539*†	0.4214*†	0.3495*	0.3429*
Difference%		+6.6%	+11.3%	−5.7%	−1.0%

effective with different corpora [37]. The evaluation results shown in Table 4.1 demonstrate that the DFR-I(n_e)B2 model yields the best (or second best) results, regardless of the stemmer used (columns "None", "Light" or "Porter") or the representation (4-gram or Trunc-4). In this table, statistically significant differences in performance compared to the best approaches (depicted in bold) are denoted with an asterisk (*). As can be seen, the performance of other probabilistic models did not differ significantly when compared to the DFR-I(n_e)B2 or LM models when the text representations were based on words. For both the 4-gram representation and *tf·idf* model, all performance differences were statistically significant. As another baseline, we could choose the performance obtained when no morphological normalization was introduced (column "None"). As shown in the last line of Table 4.1, when light or Porter stemmers were applied, the differences in performance levels were relatively small, with an augmentation of only +6.6% using the light stemmer, and +11.3% when Porter is employed.

For the 4-gram indexing strategy, the performance levels decreased to around −5.7% in average, and remained at a similar level when considering Trunc-4 (−1.0%). When applying a statistical test to verify if these differences were truly significant, we found that only when the *tf·idf* model was combined with light or Porter stemmer could the performance differences be seen as significant when compared to the baseline "None" (denoted with '†' in Table 4.1).

Finally, compared to the best performance achieved at the TREC evaluation campaign (MRR = 0.7353) (comparison required according to some authors [5]), the

values in Table 4.1 clearly show that the probabilistic models provide higher levels of performance (e.g., LM with the Porter stemmer achieves a better MRR = 0.8807, an improvement of 19.8% compared to the best TREC result).

4.2.2 Evaluation of the Noisy Corpora

Table 4.1 shows the results obtained with the clean corpus. These results are reported again in Table 4.2 under the columns entitled "Light" and "Porter". In the presence of recognition error rates of 5% and 20%, we obtained the MRR values shown in the columns labelled "5%" and "20%" respectively, the results obtained with the light or Porter stemmers. The last row of Table 4.2 shows the degradation compared to the "Light" and "Porter" baselines, showing that with the 5% error rate, the average decrease is around 22%, and about 58% with an error rate of 20%. These data also show that the use of a light or a more aggressive stemmer tended to produce similar degradation levels with noisy texts. According to the results obtained from the statistical tests that we conducted and using the clean corpus as a baseline, we found that all performance differences were statistically significant. This first analysis used a word-based indexing strategy for both documents and queries. Table 4.3 shows the results obtained when using 4-gram and Trunc-4 as document and query representations. As in Table 4.2, the performance degradation is clearly marked with an error rate of 20%. However, the 4-gram indexing scheme has proved to be more resistant to errors when compared to the word-based one, as shown in the last row of Table 4.3, and thus the decrease in performance level was around -44% instead of -58%.

When considering the Trunc-4 as an indexing strategy, Table 4.3 reports lower performance levels in all cases. As in Table 4.2, the statistical test indicated that the performance differences between the clean version and the noisy versions were always statistically significant. During the TREC evaluation campaign, the average performance on the clean collection was 0.5546. In the case of the 5% recognition error rate however the average was equal to 0.3401, indicating a deterioration of around 38.7%.

As can be seen in Tables 4.2 and 4.3, the current evaluation indicates a mean performance drop of around -20%, a difference that originates from the best performing IR models (Okapi, LM or DFR), which usually show smaller degradation levels. For the corpus suffering a 20% error rate, the average performances

Table 4.2: MRR of six IR models using words for clean and noisy (5% and 20%) corpora

IR Model	Light	5%	20%	Porter	5%	20%
Okapi	0.8231	0.6181	0.3323	0.8513	0.6113	0.3534
DFR-PL2	0.7458	0.6209	0.3112	0.7936	0.6193	0.325
DFR-I($\mathbf{n}_e$)B2	0.8343	0.5899	**0.3456**	0.8737	0.6157	**0.3477**
LM ($\lambda = 0.35$)	0.8077	**0.6631**	0.3226	0.8807	**0.6917**	0.3221
Lnu-ltu	0.7352	0.5538	0.2955	0.7703	0.5912	0.3146
tf·idf	0.4539	0.374	0.2511	0.4214	0.3703	0.2255
Difference %		−22.30%	−57.80%		−22.80%	−58.90%

Table 4.3: MRR of six IR models using 4-gram and Trunc-4 representations for clean and noisy (5% and 20%) corpora

IR Model	4-gram	5%	20%	Trunc-4	5%	20%
Okapi	0.7730	0.6472	0.4013	0.7826	0.5698	0.2785
DFR-PL2	0.6500	0.5063	0.3661	0.6353	0.5498	0.3340
DFR-I($\mathbf{n}_e$)B2	0.8240	**0.6691**	0.4083	0.8126	0.5814	**0.3702**
LM ($\lambda = 0.35$)	0.7029	0.5822	**0.4135**	0.8209	**0.6174**	0.2862
Lnu-ltu	0.5913	0.5128	0.3672	0.6921	0.5009	0.3134
tf·idf	0.3495	0.3029	0.2032	0.3429	0.2817	0.1194
Difference %		−17.2%	−44.5%		−24.1%	−58.4%

observed during the TREC campaign were 0.2663, thus a relative loss of 52%. In our experiment, the results obtained indicated similar levels of deterioration. According to Tables 4.2 and 4.3, the best performance results were obtained when using the 4-gram indexing strategy combined with either the DFR-I(n_e)B2 or LM probabilistic models.

With the presence of an error rate of 5%, the DFR-I(n_e)B2 scheme improved the performance by 17% when compared to results obtained by the best run at TREC (0.6691 *vs.* 0.5737), which is a statistically significant performance difference.

4.3 Evaluation of the Noisy Corpora when Using Pseudo-Relevance Feedback

It is known that the use of a blind query expansion technique [9] to automatically expand queries tends to improve the retrieval effectiveness. In our context such an approach seems *a priori* to be an attractive one, since the system will extract correctly-spelled words from the documents retrieved at the top of the ranked output list. To verify this assumption, we implemented Rocchio's approach [46] with the constants $\alpha = 0.75$ and $\beta = 0.75$, including between 10 and 50 new n-gram terms extracted from the first 3 to 5 retrieved items. The results obtained for the three versions of our test-collection are depicted in Table 4.4. These results are quite disappointing since the performance levels have decreased in all cases, with or without recognition errors. Moreover, these differences are always statistically significant with respect to the performance achieved before applying the blind query expansion approach (values marked with a (†) symbol in Table 4.4).

Table 4.4: MRR after Blind Query Expansion with 4-gram representation for noisy text having 5% and 20% error rate (49 queries)

IR Model	4-gram	5%	20%
DFR-I(n_e)B2	0.8240	0.6691	0.4083
3 docs / 10 terms	0.4341†	0.3265†	0.2563†
3 docs / 20 terms	0.4193†	0.3438†	0.2449†
3 docs / 50 terms	0.5071†	0.3696†	0.2667
5 docs / 10 terms	0.2697†	0.2280†	0.1607†
5 docs / 20 terms	0.3183†	0.2419†	0.1594†
5 docs / 50 terms	0.3415†	0.2615†	0.1629†

4.4 Query-by-Query Analysis

This section analyzes a selection of topics with their respective retrieval results as an attempt to depict the impact of the error rate in the retrieval process using different indexing strategies.

To illustrate the difference between word-based and n-gram indexing strategies,

we chose Topic #36 "headband" as an example. Combining the word-based representation with any kind of stemmer, the IR system was always capable of retrieving the known-item in the first rank. When this query was represented using the n-gram strategy with $n = 4$, the sequences "head", "eadb", "adba", "dban" and "band" were generated. Knowing that the only relevant article contains the term "headband" only once, this document had to compete with other documents containing multiple occurrences of the 4-grams "head" and/or "band" (e.g., as with the words "broadband," "baseband", ...). The adequate answer (the known-item) thus appeared in Rank 85 when indexing by 4-grams, resulting in a very poor performance for this query.

Nevertheless, the 4-gram strategy still resulted in better performance for many other queries, as for Topic #13 "I am looking for theft data on the Chevrolet Corsica." We know that the term "chevrolet" appeared fifteen times in the known-item while the words "theft", "thefts" and "corsica" appeared only once each, and the word "data" never occurred in that document. An inspection of the index shows that document frequencies (df) for the terms "chevrolet", "theft", "thefts" and "corsica" are 36, 254, 16 and 1 respectively. The terms were misspelled in many different ways in the degraded versions, e.g., "che.vrolet", "chevrotet", "fwhevrolet" and "L-orsica". When based on a 4-gram representation, several matches between the query and the known item can be found, as for example, the terms "evro", "vrol" extracted from "chevrolet" or "rsic" from "corsica". Whether searching in the original collection or in the 5% or 20% degraded versions, the known-item was always retrieved in the first rank using the 4-gram indexing approach. These examples show how differently indexing strategies can perform, keeping in mind that certain indexing strategies can outperform other ones for some queries, but not for all.

4.5 Summary

In this chapter we saw that character-by-character text recognition (OCR) introduces misspelled words, more precisely, ones that are not valid with respect to the underlying language word-stock (also known as out-of-vocabulary errors). Such an issue renders the matching of whole words impossible in the case of misrecognition. A good remedy for this problem is this to try to match subwords as opposed to entire ones. Such an approach will not only overcome OCR

imperfections but will also make it possible to capture morphological variations of a given word (e.g., "translate" and "translation" both yield "transl" using a character-level 6-gram. Other text analysis approaches, such as stemming, are not as interesting in this case as they are not always capable of recovering from the recognition error.

We analyzed retrieval effectiveness degradation when facing with two noisy versions of a corpus. Based on six different retrieval models, three indexing strategies and three stemming approaches, there is a relative decrease of more than 20% when having an error rate of around 5%. When indexing documents and queries using n-gram, we can recover some loss in retrieval quality. From these experiments, we found that our best-performing IR system (LM model combined with the 4-gram representation) tends to perform around 20% better than the best TREC-5 results when processing a corpus whose recognition error rate is 5% (0.6631 *vs.* 0.5737).

5

Retrieval of Digitized Historical Manuscripts I

(Word-based Text Recognition)

In this chapter we present the series of experiments that we conducted using the Parzival test-collection. As seen in Section 3.1.2, the Parzival corpus and topics are written in the Middle High German (MHG) language which lacks standardization in both spelling and grammar. A clean (error-free) version of the corpus constitutes our ground-truth for evaluating the corrupted corpora. These corrupted corpora are the result of digitizing the scanned manuscripts (Section 3.1.2.1) and are parallel to the ground-truth. The topics and the relevance judgments were generated automatically as explained in Section 3.1.2.2 and support the known-item search paradigm. This test-collection is thus fully-compliant with our research needs' checklist presented and discussed earlier in Section 3.1 (page 24).

This chapter is organized as follows: The first section (5.1) provides a detailed overview of the experimental setup of our experiments using the Parzival test-collection. Section 5.2 evaluates the error-free as well as the noisy versions of the corpus using various indexing strategies and retrieval models. In Section 5.3 we examine our evaluations even closer with a detailed query-by-query analysis for a selection of topics.

5.1 Experimental Setup

Having the TREC-5 confusion track as our benchmark, we implemented the earlier seen (Chapter 4) IR model *vs.* indexing strategy combinations with the Parzival test-collection. However, we did not assume that the best performing combinations for the TREC-5 confusion track would perform as well on Parzival for several reasons that we discuss shortly in this section.

Being written in two different languages and originating from eras that are several hundred years apart, these two test-collections bear more differences than they share in common. The average document and query lengths also differ notably. The average document length in the TREC-5 confusion track is 414.05 indexing terms (per document). The Parzival documents, on the other hand, are poetical verses whose lengths range between two and nine words. Stopword removal and stoplist length may thus impact the results differently in each case.

Furthermore, the text recognition methods used to generate the automatic transcriptions are far from being similar. This particular aspect makes the n-gram technique seem to be less interesting for the case on hand. Apart from morphologically related words (ones that share the same stem or root), partial string matching is less likely to happen in mis-recognized words. In the TREC-5 confusion track, both the 5% and 20% corpora are the result of a character-by-character text recognition tool (OCR). Such a system is likely to produce, in case of mis-recognition, some out-of-vocabulary terms, i.e., ones that do not belong to the dictionary of the language of the corpus on hand (e.g., as can be seen in Figure 3.1, "appropriate" *vs.* "approphate" in the original and the 20% noisy versions respectively). In this case, using n-grams (with n=*4* for instance) partial matching is profitable for finding the similar yet pertinent matches (e.g., "appr", "ppro"). The Parzival corpus, on the other hand, is based on word-by-word recognition which in case of mis-recognition yields incorrect yet valid (vocabulary-wise) words. The n-gram method may thus seem of less use whereas stemming could be a more interesting technique. However, these are mere speculations and loudly spoken (written) thoughts that do not necessarily reflect what turn things may take in practice, and as we said earlier we are avoiding pre-assumptions as the case on hand possesses its own unique aspects. We present the different stemming and other text processing methods that we adopted in the next section.

5.1.1 Stemming

The stemming strategy is one (among others) that usually provides good performance levels for the German language. Fautsch *et al.* [20] for instance, found that performing light stemming for German yielded better results than with the absence thereof. In our experiments, we used three stemming variations to which we refer as: No stemming, light, and aggressive stemming (respectively denoted with "None", "Light", "Aggr" in the Tables in this chapter). Obviously, the first technique completely ignores stemming hence the label "None". We developed two stemmers for the MHG language basically aiming at removing the most widely occurring affixes yielding stems whose length is no less than three characters.

5.1.1.1 Middle High German Light Stemmer

The light MHG stemmer is somewhat similar to its English counterpart presented earlier [25] (see Section 2.2) in the sense that it only applies a small set of rules for removing the plural sign. In the German language (Middle High as well as modern German) three suffixes can be removed using this stemmer, namely *"-e"*, *"-en"*, or *"-er"*.

5.1.1.2 Middle High German Aggressive Stemmer

We have developed an aggressive stemmer for the Middle High German language analogous to the Porter algorithm [45] in attacking derivational and inflectional affixes alike. This MHG-compatible stemmer features a set of rules which was created manually with the help of experts in the domain. This stemmer is multi-tiered in the sense that it strips n affixes from a given word in an semi-iterative fashion leaving a final stem that consists of no less than three characters. The term *affixes* is used intentionally in this context since our aggressive algorithm not only removes suffixes but also prefixes such as *"ne-"* and ones that are associated with suffixes together constituting the past participle form of verbs (e.g., *"ge-"* and *"-en"* as in *"gebaken"* (baked)). The algorithm of this stemmer is shown in Figure 2 in Appendix B.

5.1.2 Stoplists

We generated two lists of non-content-bearing words. The first one initially contained 150 items featuring very short word types (three characters at most) as well as very frequent word types in the corpus. This list was generated automatically, however human intervention was needed at this point to spot very short and/or very frequent yet content-bearing words (e.g., proper nouns). In our case, after manually reviewing the list, we detected four content-bearing words and hence excluded them from the stoplist leaving it with 146 words as oppodes to 150 ones. The second stoplist consists of four word types only, namely: *"der"*, *"daz"*, *"ir"* and *"er"*. These words correspond to the four most frequent ones from the previous stoplist. The first stoplist was not used during indexing but was essentially created for the automatic query generation process (Section 3.1.2.2). The shorter stoplist was incorporated during the indexing in a subset of our experiments which contributed to a slight improvement in performance.

5.1.3 Decompounding

The German language is one that is characterized by its compound construction. Such a structure makes it possible to introduce new concepts by combining two or more words into one. Interestingly, it is usually acceptable to describe a certain concept by the corresponding compound word as well as by its separate building words (e.g., *"Bankpräsident" vs. "Präsident der Bank"*) both meaning "bank president". As mentioned earlier, word compounding was not as frequently used in the Middle High German Language as it is in modern German. The percentage of compound nouns to nouns in the first half of the 13^{th} century was around 6.8%, this ratio increased over the centuries reaching 25.2% in the modern German language [23]. We, nonetheless, performed a series of runs that supported decomponding. Our decompounder is based on a set of rules that were manually compiled with the help of experts in Middle High German. We then transformed these rules into a fully automatic decompounder which we integrated into our IR system.

Our decompounder inspects each token in a text (document or query), verifies whether it can be broken down (decompounded) into two or more valid words with the condition that the token is more than k characters in length. According to [20] k=5 yields best performance in German, however we found that k=4

worked best in our case. A valid word is one that belongs to the vocabulary set of the underlying language. A token is decompounded if and only if all of its building words are valid ones.

Due to the lack of a predefined lexicon for our corpus, we had to compile a vocabulary set (wordstock) based on the transcription. Each word that appears at least once in the transcription is a valid entry in this vocabulary set, which is equivalent to the entire set of words (recognitions) in BW7. But we also included indexing terms that resulted from both of our aforementioned stemming algorithms, namely, light and aggressive. The benefit sought from this is to integrate word forms without the added derivational or inflectional affixes as they are very unlikely to appear in the unprocessed (unstemmed) transcription as individual tokens (i.e., not a part of a compound word). Further, some valid words correspond to affixes too (e.g., *"en"*, *"e"*, *"er"*). As a result, the decompounder not only decomposes words but also provides further stemming-like and variable n-gram-like functionalities. This aspect may justify the better performance associated to a lower value of k (i.e., k=4 as opposed to 5).

As a concrete example from our test-collection, let us inspect the token *"anderstvnt"* from Query #23 in QT3. This token was successfully decomposed into *"ander"* and *"stvnt"*. Obviously, this case corresponds to the traditional output of a decompounder. Another example which illustrates the implicit ultralight stemming performed by our decompounder is the word *"behalten"*. This word was decompounded into *"behalt"* and *"en"* which corresponds to some sort of stemming. The second building word (namely *"en"*) also corresponds to the infinitive sign in verbs in the German language. It is worth mentioning that unlike stemmer-based indexing which only keeps the resulting stems, decompounders retain the resulting building words in addition to the original undecomposed tokens in the index. With the token *"behalten"* for instance, *"behalten"*, *"behalt"* and *"en"* are all present in the index when the decompounder is used, whereas only *"behalt"* is retained in a stemming-based index. It is then up to the underlying IR model to weigh and remove very frequent words which are very likely to be affixes that denote a derivation or a certain grammatical class (such as *"-en"* in our example).

5.2 Evaluation

The TREC-5 evaluation campaign selected the MRR (mean reciprocal rank) as a retrieval effectiveness measure. In our case, having this campaign as our benchmark, we opted for this same measure. As the relevant items are verses, returning the verse immediately before or after the correct one should not be regarded as fully-impertinent, particularly when knowing that in practice users usually reading a few verses at a time (a passage) comes more naturally than reading a single line isolatedly. Considering the immediate neighbor verses to be fully-relevant is inadequate either. To allow some degree of relevance for adjacent lines, we need a measure that supports graded relevance. But since we must adhere to MRR in order to have comparable results with those obtained with the TREC-5 confusion track, and to the best of our knowledge, none of the available measures for graded relevance is fully-compatible to MRR. Therefore, we adapted the strict MRR computation to support graded relevance as seen in Chapter 3. We used the GMRR measure in our evaluation.

The next section presents the results for the clean corpus (the error-free, manually transcribed text). As mentioned earlier, this version of the corpus constitutes our baseline for evaluation and we refer to it as the ground truth corpus (GT). Section 5.2.2 evaluates the noisy versions of the corpus. As explained in Section 3.1.2, these noisy versions are based on the automatic recognition results of a HMM-based, word-by-word recognition system having a word error rate of around 6%.

To determine whether the performance differences obtained using the various combinations of retrieval models *vs.* indexing strategies accross the different corpora are statistically significant or otherwise, we opted for a bootstrap non-parametric test [54] (significance level $\alpha = 5\%$).

5.2.1 Evaluation of the Clean Corpus

As seen earlier in Section 4.2, we opted for a word-based representation without any morphological normalization as our baseline representation. Having our focus on high precision, we will continue in this vein for evaluating the Parzival corpora.

For the modern German language, stemming and decompounding usually provide better performance levels than the bag of words representation according to [20].

Light stemming is usually deemed effective since it simply removes a very limited number of basically plural suffixes (columns labelled "Light"). The evaluation when eliminating some derivational suffixes, as it is the norm in many IR empirical studies, is another possibility and is shown under the label "Aggr" in our results. Finally, the decompounding technique (labelled "Dec") can also offer as good performance levels as stemming techniques generally do.

As an alternative to the word-based model, we selected the n-gram model with $n = 4$ which usually provides good performance levels. Finally, we used the Trunc-n strategy with $n = 4$, one which is found effective for different corpora [37].

In terms of IR models, we chose two vector-space schemes (Lnu-ltu and *tf·idf*), probabilistic models (Okapi, DFR-I(n_e)B2 and DFR-PL2), and a language model (LM). As can be seen so far, this experimental setting is similar to that presented in the previous chapter with the TREC-5 confusion track with the exception of the decompounding strategy.

Table 5.1 shows the GMRR values obtained for the single-term query set (QT1) which is composed of 60 topics. Obviously, the baseline representation (no stemming) offers the best results overall. Decompounding yielded the least degradation in average performance over the different IR models. Significant performance differences compared to the baseline levels are denoted with (†). We can see that the differences were always statistically significant when light or aggressive stemming or truncation were used.

Within each representation (each column), statically significant differences between each IR model and the best performing one(s) (depicted in bold) are denoted with an asterisk (*). Obviously, these differences are not statistically significant in this case.

Moving on to the 2-term query set (QT2, see Table 5.2), the overall performances seem to have degraded but however the differences from the baseline have notably decreased. Trunc-n continues to significantly underperform almost always. Setmming seems to perform better here but still not as well as decompounding does given that the latter achieved around 1.88% amelioration with respect to the baseline. The 4-gram technique seems to combine best with the *tf·idf* model so far.

Table 5.1: GMRR for the ground-truth corpus (GT) using six IR models and six representations (QT1, 60 queries)

IR Model	Word-based			4-gram	Dec	Trunc-4
	None	Light	Aggr			
Okapi	**0.7749**	**0.6825**†	0.6123†	0.7261	**0.7667**	0.5883†
DFR-PL2	**0.7749**	**0.6825**†	0.6123†	0.7182	0.7631	0.5883†
DFR-I(n_e)B2	**0.7749**	**0.6825**†	0.6123†	0.7178	0.7598	0.5883†
LM ($\lambda = 0.35$)	**0.7749**	**0.6825**†	0.6123†	0.7265	0.7617	0.5883†
Lnu-ltu	**0.7749**	0.6822†	0.6121†	0.7189	0.7625	0.5881†
tf·idf	0.7560	0.6717†	**0.6271**†	**0.7461**	0.7486	**0.5956**†
Difference %		−11.80%	−20.34%	−5.98%	−1.47%	−23.62%

Finally, with the 3-term queries (QT3, see Table 5.3), decompounding seem to maintain a good relative performance, also, the 4-gram strategy shows a continuous stability in terms of average performance whereas Trunc-4 continues to disappoint. Stemming, on the other hand, illustrated highly fluctuating performance levels especially the aggressive variety with statistically significant differences (columns labelled "Aggr").

Table 5.2: GMRR for the ground-truth corpus (GT) using six IR models and six representations (QT2, 60 queries)

IR Model	Word-based			4-gram	Dec	Trunc-4
	None	Light	Aggr			
Okapi	0.6688	0.6469	0.6249	0.6472*	0.6896	0.5878†
DFR-PL2	0.6688	0.6510	0.6291	0.6458*	0.6951	0.6003
DFR-I(n_e)B2	0.6781	**0.6640**	0.6406	0.6454*	0.6813	**0.6018**†
LM ($\lambda = 0.35$)	0.6781	0.6637	0.6403	0.6451*	**0.6993**	**0.6018**†
Lnu-ltu	0.6760	0.6634	0.6398	0.6436	0.6760	0.6011†
tf·idf	**0.6785**	0.6586	**0.6448**	**0.6781**	0.6829	0.5926†
Difference %		−2.48%	−5.65%	−3.53%	1.88%	−11.43%

Table 5.3: GMRR for the ground-truth corpus (GT) using six IR models and six representations (QT3, 60 queries)

IR Model	Word-based			4-gram	Dec	Trunc-4
	None	Light	Aggr			
Okapi	0.7054	0.6618	0.6594†	0.6797	0.7111	**0.6609**
DFR-PL2	0.6970	0.6597	0.6573†	0.6788	0.7111	0.6595†
DFR-I(n_e)B2	0.7097	0.6674	0.6572†	**0.6931**	**0.7278**	0.6402†
LM ($\lambda = 0.35$)	0.7097	0.6675	0.6642†	0.6819	0.7111	0.6569†
Lnu-ltu	0.7097	0.6674	0.6644†	0.6792	0.7028	0.6569†
tf·idf	**0.7111**	**0.6683**†	**0.6679**†	0.6806	0.7097	0.6366†
Difference %		−5.91%	−6.42%	−3.52%	0.73%	−7.82%

To recap these results, we can say that our decompounding technique performs very satisfactorily, and to the contrary of the unspoken pre-assumptions and predictions (that we earlier promised to avoid), the n-gram strategy (with $n = 4$) provided quite good results (even better than stemming). It is also noteworthy that using no stemming at all (None) also constitutes a very good strategy especially for the shortest queries (single-termed) which makes it a good solution as it provides good performance level at a lower budget than that needed for either n-gram or decompounding.

5.2.2 Evaluation of the Noisy Corpora

This section evaluates the four noisy versions of the Parzival corpus presented earlier in Section 3.1.2. These are namely, BW1 (the classical output of a text recognition system - that is to consider only the most likely recognition per word), BW3 (retaining the three most likely recognitions per word), BW7 (keeping all seven word recognitions) and BWδ (for each word, the most likely recognition is retained and possibly one or more highly likely ones, too). The idea of incorporating more than one recognition per word is to overcome the issues that the text suffers from (imperfect recognition and lack of language standardization). This can be seen as some sort of document expansion.

With the error-free corpus GT as our baseline, we evaluated each of these versions using the various combinations of IR models *vs.* indexing strategies and the three

query sets QT1, QT2 and QT3 as seen in the previous section.

As a starting point, we will present and compare the performance of all the corpora (BW1, BW3, BW7, and BWδ) with respect to the baseline (GT). The aim is to conclude which corpus surrogates are the most robust against the aforementioned noise sources, and which IR models and indexing strategies combinations are the most effective ones.

First of all, we inspect the overall retrieval performance obtained from each corpus surrogate using the three query sets individually (QT1, QT2 and QT3) without any morphological normalization (no stemming). In particular, we inspect the difference in average performance with respect to our baseline (GT) for each query set at a time.

Table 5.4 shows the results for single-term queries (QT1). The first observation is that BWδ outperforms the other corpora with a degradation of only 6.30% with respect to the baseline GT. Secondly, the degradation in performance with BW1 is 10.20%, which is not far below that of BWδ, however the differences for each IR models with BW1 were statistically significant from their GT counterparts while they are not with BWδ. The third observation is that BW3 and BW7 resulted in very poor performance with degradation levels of 45.34% and 64.52% respectively. Another phenomenon that we have experienced earlier with GT but was not highlighted back then, is the identical performance levels for different IR models within each of GT and BW1. This may be due to the shortness of both the queries and the documents (verses) with the homogeneity of document lengths across the corpus and the rarity of multiple occurrences of a given term within one verse. Within each corpus, the differences between the best performing IR model(s) (in bold) and the rest were not statistically significant.

Moving on to 2-term queries (QT2) (Table 5.5), once again, BW3 and BW7 underperformed both BW1 and BWδ. The differences in average performance with respect to GT are smaller than those obtained with QT1 but they are nonetheless statistically significant. Overall, the performance levels are lower (even for GT) than those obtained in the QT1 experiment. This is (partially) due to the additional query term that is likely to have originated from an adjacent document (verse) to the target one (the known-item). This aspect is two-edged: it adds a slight level of difficulty to the query knowing that this term belongs to a different document than the known-item. Yet at the same time, retrieving the (somewhat

relevant) adjacent document due to the other term will contribute to eventually finding the target one. As can be seen from Table 5.5, the difference between BW1 and GT are not statistically significant whereas for BWδ, the differences from GT with both DFR-PL2 and *tf·idf* were statistically significant.

Table 5.6 shows the GMRR values using 3-term queries (QT3) across the corpus versions. As the queries get longer, there is a clear tendency of decreasing differences in average performance with respect to the baseline GT. Another observation is that when DFR-PL2 or LM are combined with BW3, the differences are not statistically significant from GT. The BW7 corpus, once again, yielded less than satisfactory results. Having also experimented with other indexing strategies (light and aggressive stemming, decompounding, n-gram and Trunc-n) BW3 and BW7 representations did not thrive into satisfactory results.

The vicissitude of DFR-PL2 here is noteworthy as it went from being significantly underpeforming in BWδ with QT2 to the best performing with QT3 for the same corpus, however we do not have an explanation for that. Another interesting observation is BWδ's stable difference in terms of average performance with respect to the baseline GT whereas the corresponding differences between BW1 and GT are diminishing as queries grow in length.

An early conclusion is that the intensity of retaining word alternatives is crucial to the final IR effectiveness. It is clear that BW3 and BW7 do not constitute good surrogates. This is due to the inclusion of too many word recognition hypotheses (especially BW7) and the lack of the discriminative nature of BWδ between likely- and less likely-correct word recognitions. Since the threshold at which representative word hypotheses are kept is fixed for both BW3 and BW7 alike, terms with relatively low log-likelihood values were nonetheless included, thus forcing likely incorrect word recognitions into the text representation. This aspect is very taxing on retrieval effectiveness as it eventually resulted in more harm (noise) than improvement (enrichment) in terms of IR performance. With the BWδ representation, this noise is quite limited (but not completely eliminated though) by discarding word recognitions with relatively low log-likelihood values of being correct. Finally, BW1, also provided good performance levels especially for longer queries.

Table 5.4: GMRR for different recognition corpora together with the GT as the baseline using six IR models, no stemming (QT1, 60 queries)

IR Model	GT	BW1	BW3	BW7	BWδ
Okapi	**0.7749**	**0.6971**†	0.4218†	0.2681†	0.7166
DFR-PL2	**0.7749**	**0.6971**†	0.4218†	0.2681†	0.7166
DFR-I(n_e)B2	**0.7749**	**0.6971**†	0.4218†	0.2681†	0.7166
LM ($\lambda = 0.35$)	**0.7749**	**0.6971**†	0.4218†	0.2681†	0.7166
Lnu-ltu	**0.7749**	**0.6971**†	**0.4312**†	0.2702†	0.7275
tf·idf	0.7560	0.6727†	0.4125†	**0.3005**†	**0.7447**
Difference %		−10.20%	−45.34%	−64.52%	−6.30%

Table 5.5: GMRR for different recognition corpora together with the GT as the baseline using six IR models, no stemming (QT2, 60 queries)

IR Model	GT	BW1	BW3	BW7	BWδ
Okapi	0.6688	0.6465	0.5039†	0.4141†	**0.6346**
DFR-PL2	0.6688	0.6549	0.4952†	0.4114†	0.6054†
DFR-I(n_e)B2	0.6781	**0.6558**	0.5092†	0.4160†	0.6113
LM ($\lambda = 0.35$)	0.6781	**0.6558**	0.5109†	0.4172†	0.6182
Lnu-ltu	0.6760	0.6454	0.5083†	**0.4185**†	0.6290
tf·idf	**0.6785**	0.6451	**0.5288**†	0.4076†	0.6054†
Difference %		−3.57%	−24.50%	−38.62%	−8.50%

At a first look at those results, one can easily see that BWδ constitutes the most effective representation compared to the other corpora when retrieving single-term queries whereas BW1 has shown to be best for longer ones. At this point, it is not easy to decide which IR model is the best nor can we pick one out of BW1 and BWδ over the other. As a solution, we will combine the three sets of queries (QT1, QT2 and QT3) into one large set of 180 queries of mixed yet equally distributed lengths and we will refer to it as QTAll. The objective of this step is to obtain a global picture of effectiveness in an environment that, though not optimal, but is close enough to practice since real users (and standard test-collections such as TREC-5 confusion track) usually feature queries of mixed lengths.

Table 5.7 shows the results we obtained with the BW1 corpus using light and aggressive stemming, 4-gram, decompounding and Trunc-4 with the 180 queries of

Table 5.6: GMRR for different recognition corpora together with the GT as the baseline using six IR models, no stemming (QT3, 60 queries)

IR Model	GT	BW1	BW3	BW7	BWδ
Okapi	0.7054	0.6859	0.6340*†	0.5587†	0.6706
DFR-PL2	0.6970	**0.6915**	0.6422	**0.5677**†	**0.6718**
DFR-I(n_e)B2	0.7097	0.6903	0.6415†	0.5589†	0.6161*†
LM ($\lambda = 0.35$)	0.7097	0.6903	**0.6529**	0.5523†	0.6647
Lnu-ltu	0.7097	0.6903	0.6459†	0.5578†	0.6544†
tf·idf	**0.7111**	0.6771†	0.6336†	0.5259*†	0.6313†
Difference %		−2.77%	−9.25%	−21.72%	−7.87%

QTAll. As a baseline, we use a no-stemming representation shown in the column labelled "None". The complete results with BW1 and the three query sets (QT1, QT2 and QT3) run individually are provided in Tables 1, 2 and 3 in Appendix A. We performed the statistical tests when using the individual query sets (QT1, QT2 and QT3) but not when combined (QTAll).

The results shown in Table 5.7 show that a representation without any morphological preprocessing (column labelled "None") provides the best results overall with this representation which is also the baseline for assessment in this table. The second best strategy is to use our decompounder while the 4-gram strategy comes third. A closer look on Tables 1, 2 and 3 in Appendix A shows that these indexing strategies are also the best performing with each query set when tested individually and that no statistically significant differences from the baseline were experienced. Just like with the GT corpus, Trunc-4 is always significantly poor in performance and stemmers are once again fluctuating with high and low effectiveness levels.

Table 5.8 shows the results obtained with the BWδ corpus using the different stemmers, 4-gram, decompounding and Trunc-4 with the 180 queries of QTAll. As done previously with BW1, the no-stemming representation (None) is used as a baseline.

As can be seen in Table 5.8, our decompounder provides the best results, thus outperforming the baseline by 1.64% followed by the 4-gram strategy. A closer look at Tables 4, 5 and 6 in Appendix A shows BWδ's similar behavior to that experienced with BW1 in the sense that these indexing strategies are also the

Table 5.7: GMRR for the BW1 corpus with six IR models and six indexing strategies, and queries of different lengths (QTAll, 180 queries)

	Word-based					
IR Model	None	Light	Aggr	4-gram	Dec	Trunc-4
Okapi	0.6765	0.6346	0.6166	0.6453	0.6788	0.5819
DFR-PL2	**0.6811**	0.6352	0.6163	0.6455	**0.6822**	0.5843
DFR-I(n_e)B2	**0.6811**	**0.6404**	0.6182	0.6547	0.6762	0.5826
LM ($\lambda = 0.35$)	**0.6811**	0.6403	**0.6203**	0.6465	0.6813	0.5818
Lnu-ltu	0.6776	0.6360	0.6169	0.6446	0.6716	**0.5982**
tf·idf	0.6650	0.6367	0.6140	**0.6592**	0.6638	0.5798
Difference %		−5.89%	−8.86%	−4.10%	−0.21%	−13.63%

best performing with each query set when tested individually with no statistically significant differences from the baseline. Just like with the GT and BW1 corpora, Trunc-4 is always significantly poor in performance and stemmers are far from stable.

As experienced earlier, decompounding or 4-gram are best for single-term queries (QT1, see Table 4 in Appendix A). Whereas differences in performance with both stemming techniques and Trunc-4 are all statistically significant with respect to the whole-word, stemming-free baseline. However, these differences diminish in longer queries.

As for IR models, it is quite noticeable that there seems to be an association between the performance of a given IR model and the query length. For single-term queries, *tf·idf* was generally outperforming the other IR models. This can be due to the multi-hypotheses per word nature of BWδ.

For 2-term queries (QT2), the Okapi model was ahead of its counterparts regardless of the strategy used, while it shared being the best performing model with DRF-PL2 in QT3.

To set the scope on the noisy corpora and how well they influence the final effectiveness, we will compare the two relative baselines (whole word, no stemming) to see whether the novel one (BWδ) is as good as the conventional one (BW1). Table 5.9 shows that BWδ has a degradation of 1.93% from BW1. However no statistical differences exist between the two corpora but BWδ is far superior to

Table 5.8: GMRR for the BWδ corpus with six IR models and six indexing strategies, and queries of different lengths (QTAll, 180 queries)

	Word-based					
IR Model	None	Light	Agg	4-gram	Dec	Trunc-4
Okapi	**0.6739**	**0.6506**	**0.6234**	0.6732	**0.6938**	**0.5947**
DFR-PL2	0.6646	0.6364	0.6095	**0.6750**	0.6905	0.5892
DFR-I(n_e)B2	0.6480	0.6256	0.5984	0.6580	0.6618	0.5793
LM ($\lambda = 0.35$)	0.6665	0.6380	0.6126	0.6634	0.6730	0.5847
Lnu-ltu	0.6703	0.6417	0.6203	0.6547	0.6664	0.5923
tf·idf	0.6605	0.6317	0.6210	0.6599	0.6635	0.5646
Difference%		−4.01%	−7.50%	0.01%	1.64%	−12.03%

BW1 when it comes to single-term queries. With that being said and with a very small and non-significant difference in average performance, our multi-hypotheses method (BWδ) is, so far, more beneficial than the conventional output of text recognition (BW1).

Table 5.9: GMRR for the BW1 (baseline) and BWδ corpora with six IR models, no stemming, and queries of different lengths (QTAll, 180 queries)

IR Model	BW1	BWδ
Okapi	0.6765	**0.6739**
DFR-PL2	**0.6811**	0.6646
DFR-I(n_e)B2	**0.6811**	0.6480*
LM ($\lambda = 0.35$)	**0.6811**	0.6665*
Lnu-ltu	0.6776	0.6703
tf·idf	0.6650	0.6605
Difference %		−1.93%

5.3 Query-by-Query Analysis

Single-term Queries #4, #11, #25, and #43 are composed of a rare term each, having, according to the GT corpus, *df* values equal to 1 (term appearing only in the known-item). Using the BW1 corpus and the Okapi model, none of these four queries got any pertinent items retrieved.

Query #4 *"machete"*, for instance, has the verse *"machete daz er gein ir sp(ra)ch"* as its known-item. Using the BW1 corpus, the most likely recognition of the term *"machete"* is *"machen"* which is an incorrect recognition. Since this is the only occurrence of the term in the collection, the system failed to retrieve any documents in that case. Using the BWδ corpus, the known-item was retrieved at Rank 1 since the term *"machete"* was the second likely recognition with a log-likelihood difference of less than 1.5%. This second possible recognition was therefore included in the text representation. Using the BW3 or BW7 corpora, the known-item was also retrieved but in lower ranks, precisely, in Rank 3 for BW3 and 18 for BW7.

The underperformance of BW3 and BW7 is due to the fact that these two corpora include more alternatives which merely acted as noise in this case as well as in many other cases. Non-relevant documents containing the search term alternatives are now competitors as the final ranking depends on the *tf* values. A non-relevant item may thus appear before the target verse in the ranked list of retrieved items.

Applying the automatic decompounding strategy clearly provides some successful improvement overall, especially when combined with BWδ regardless of the query length. Decompounding introduces potential variations of a given term which contributes to capturing morphologically-related or orthographically-similar hypotheses from the BWδ representation. As an example, let us inspect Query #7 from QT1 *"Grale"*. With the decompounder, this term will be represented in its full form (i.e., *"Grale"*) as well as its compounds *"Gral"* and *"e"*. Clearly, the decompounder here acted as a light stemmer which was eventually beneficial. The known-item possesses several hypotheses for the term *"Grale"* including the correct term itself as well as *"Gral"*. The decompounding contributed to increasing the *tf*s of both hypotheses which led to stronger matching with the query.

This aspect is clearly missing in the BW1 representation, this is why the BWδ once coupled with decompounding retrieved the known-item one rank earlier than BW1 did (Rank 3 *vs.* Rank 4). However, this same aspect caused the late arrival of the known-item since the first document retrieved contains more variations of the word *"Gral"* in its hypotheses (namely, *"Grals"*, *"Grale"*, and *"Grales"*) which permitted it to compete and eventually win the *tf* race against the known-item.

5.4 Summary

In this chapter, we presented our results for the Parzival dataset. In this dataset, the documents are based on the automatic transcription of a word-based recognition system that uses a Hidden Markov Model (HMM) and produces a word error rate of 6%. To cope with this error rate and the various language irregularities (spelling variation, grammar flexibility) we proposed a special multi-hypotheses representation which makes use of the recognition results. We showed that our special multi-hypotheses representation provides superior results to those obtained using the tradiditonal output of a text-recognition system, that is, the best word hypothesis. We also found that including too many recognition hypotheses per word is not the remedy for surviving recognition error as they tend to incorporate incorrect and impertinent word recognitions into the text representation to some degree.

In terms of text representations, our decompounder wich also performs ultralight stemming and variable n-gram-like functionalities has shown to be one of the best representations followed by whole-word representation without any stemming and then by an n-gram representation with n=4. With modern High German, on the other hand, stemming is usually an effective approach according to [20] as discussed in Section 5.2.1.

6

Retrieval of Digitized Historical Manuscripts II

(Line-based Text Recognition)

The main objective of the HisDoc project is to transform manuscripts from ink on parchment into fully searchable digital documents with the least human intervention possible. In the retrieval module of this project, we develop IR methods that maximize the effectiveness of retrieval with the presence of the various issues and sources of noise that the texts suffer from in terms of the transcription quality and the available resources.

The major part of the research done within the frame of this dissertation uses the recognition results obtained from the recognition system developed at the University of Bern, Switzerland. This system uses the Hidden Markov Model (HMM) as presented earlier in Chapter 3. During the latest stages of the recognition module, another recognition system was developed using special neural Networks (NN) and performs line-based text recognition (a detailed description is available in [21]). This NN recognition system is indepenedent from its HMM counterpart.

As far as information retrieval is concerned, we are interested in finding scalable and transferable retrieval methods and text representations. In order to avoid overfitting, we need to validate our methods on different corpora and/or under different circumstances whenever possible. It is important to see whether the

multi-hypotheses representation would be as fruitful as it proved to be with the HMM Parzival recognition. That is, when applied to other corpora (i.e., other manuscripts using the HMM-based word-by-word recognition) and to other recognition systems (e.g., using NN-based line recognition for Parzival).

In addition to the classical best word (BW1) representation, we thus applied our special multi-hypotheses method to the recognition results obtained from the NN-based system. The original manuscript is, once again, the Parzival poem. The test set verses are the same as the ones used in the previous HMM-based system. However, one major difference is that not all tokens possess a set of seven possible recognitions. Some tokens may possess less than seven. Therefore, each token possesses any number of recognition alternatives between one and seven inclusive.

The reason of the variable size recognition set is due to the underlying method used (described in details in [35], [65] and in Chapter 8 in [21]).

One more dissimilarity between the NN-based and the HMM-based recognition results is that in NN, the recognition scores are probabilities rather than log-likelihoods. The score ranges are thus not the same as those experienced with HMM. This means that the value δ needs to be re-adjusted and fine-tuned accordingly.

Another dissimilarity between the two systems, or more particularly, their recognition results, is that the alternatives produced for a given token do not necessarily overlap with those produced by the other system for a given occurrence of that token. To illustrate this, let us compare the transcription of both systems to our previous example verse *"dem man dirre aventivre giht"* (Figure 3.4). Table 3.1 shows the recognition results of the HMM, word-based system whereas those obtained from the NN, line-based system are shown in Table 6.1.

As can be seen from Tables 6.1 and 3.1, the recognition alternatives are quite different. There are hardly any overlapping recognitions, yet the rankings within each set are not similar either. We generated two corpora analoguous to BW1 and BWδ using these line-based recognition results, we refer to these as BW1_{NN} and BWδ_{NN} respectively. For evaluation, we used the same query sets that we have been using so far which are generated from the HMM-based transcription.

The rest of this chapter is organized as follows: The first section describes the general setup of the experiments that we conducted. The second section presents,

Table 6.1: Sample Parzival verse with word recognition#probability pairs

dem		**man**		**dirre**		**aventivre**		**giht**	
den	-1.7865	man	-1.6409	di	-2.1981	aventivre	-1.7204	giht	-0.9293
de	-2.1160	ia	-1.7434	du	-2.3865	wenne	-2.2305	gibt	-1.9892
din	-2.2193	an	-2.2721	die	-2.4208	venie	-2.2888	niht	-2.4704
dem	-2.2909	namn	-2.3236	den	-2.6151	denne	-2.3478	aht	-2.6295
drin	-2.3183	nam	-2.4058	dir	-2.6366	ave	-2.3598	siht	-2.6730
denn	-2.3506	iar	-2.5662	da	-2.6670	nennet	-2.3611	iht	-2.8819
ein	-2.5170	san	-2.6512	An	-2.7765	denn	-2.4362	git	-2.8920

evaluates and analyzes the results obtained.

6.1 Experimental Setup

We have seen so far that the decompounder representation is (one of) the most effective indexing strategies in our experiments. The no-stemming strategy served as our baseline and also showed to be effective as well.

However, it is not clear to us which IR model is the best. Here, we decided to combine the same six IR models that we have been using so far (namely, Lnu-ltu, *tf·idf*, Okapi, DFR-I(n_e)B2, DFR-PL2, and the language model LM) with the best two indexing strategies mentioned above. We ran the QTAll query set against BW1$_{NN}$ and BWδ_{NN}. After performing several runs each time setting δ to a certain value (precisely, 50, 60, 70 and 80%), we found that δ =70% yields the best retrieval effectiveness.

6.2 Evaluation

From a text-recognition point of view, the NN-based system is more accurate than the HMM one, thus suffering a recognition error rate inferior to the that produced with HMM. This should be reflected on the retrieval quality overall, too. However, the improvement is less pronounced retrieval-wise. This is true for BW1$_{NN}$ since several single-term queries did not match any documents.

Table 6.2 shows the GMRR for both corpora with both indexing strategies and using the QTAll set (180 queries of different lengths). The five queries that did not retrieve any matching documents are given a GRR=0 each.

As for IR models, the various models tended to produce very close results to each other but the DFR family members and *Lnu-ltu* seem to provide good performance levels overall. By examining Table 6.2, we can see that using BWδ_{NN} with the decompounder is a successful combination with several IR models. Tables 6.3, 6.4, and 6.5 report the GMRR values for each query set individually. One can notice that the multi-hypotheses representation tends to work well for both QT1 and QT3 even with the absence of decompounding. Whereas performing decompounding with either BW1$_{NN}$ or BWδ_{NN} seems to work best for 2-term queries.

Table 6.2: GMRR for the BW1$_{NN}$ and BWδ_{NN} corpora with six IR models, two indexing strategies, and queries of different lengths (QTAll, 180 queries)

IR Model	BW1$_{NN}$ None	BW1$_{NN}$ Dec	BWδ_{NN} None	BWδ_{NN} Dec
Okapi	0.6636	0.6793	0.6879	0.7029
DFR-PL2	0.6663	0.6840	0.6910	**0.7064**
DFR-I(n_e)B2	**0.6679**	**0.6856**	0.6821	0.6946
LM ($\lambda = 0.35$)	0.6665	0.6807	**0.6929**	0.7022
LNU-ltu	0.6668	0.6746	0.7091	0.6984
tf·idf	0.6548	0.6788	0.6920	0.6904
Difference %		2.44%	4.24%	5.24%

Table 6.3: GMRR for the BW1$_{NN}$ and BWδ_{NN} corpora with six IR models, two indexing strategies, and single-term queries (QT1, 60 queries)

IR Model	BW1$_{NN}$ None	BW1$_{NN}$ Dec	BWδ_{NN} None	BWδ_{NN} Dec
Okapi	0.7004	0.7062	0.7632	0.7525
DFR-PL2	0.7004	**0.7167**	0.7632	0.7437
DFR-I(n_e)B2	0.7004	0.7133	0.7632	**0.7545**
LM ($\lambda = 0.35$)	0.7004	0.7082	0.7632	0.7437
LNU-ltu	**0.7013**	0.7153	**0.7733**	0.7415
tf·idf	0.6826	0.6983	0.7728	0.7469
Difference %		1.73%	9.87%	7.09%

Table 6.4: GMRR for the BW1$_{NN}$ and BWδ_{NN} corpora with six IR models, two indexing strategies, and 2-term queries (QT2, 60 queries)

IR Model	BW1$_{NN}$ None	BW1$_{NN}$ Dec	BWδ_{NN} None	BWδ_{NN} Dec
Okapi	0.6200	0.6638	0.6140	0.6616
DFR-PL2	0.6283	0.6688	0.6235	**0.6699**
DFR-I(n_e)B2	**0.6286**	0.6590	0.5968	0.6408
LM ($\lambda = 0.35$)	**0.6286**	0.6660	0.6138	0.6656
LNU-ltu	0.6272	0.6417	**0.6513**	0.6690
tf·idf	0.6149	**0.6703**	0.6197	0.6575
Difference %		5.92%	−0.76%	5.79%

Table 6.5: GMRR for the BW1$_{NN}$ and BWδ_{NN} corpora with six IR models, two indexing strategies, and 3-term queries (QT3, 60 queries)

IR Model	BW1$_{NN}$ None	BW1$_{NN}$ Dec	BWδ_{NN} None	BWδ_{NN} Dec
Okapi	0.6708	0.6799	0.6864	0.6944
DFR-PL2	0.6708	0.6785	0.6864	**0.7056**
DFR-I(n_e)B2	**0.6753**	**0.6965**	0.6863	0.6886
LM ($\lambda = 0.35$)	0.6711	0.6799	0.7017	0.6972
LNU-ltu	0.6725	0.6788	**0.7028**	0.6847
tf·idf	0.6674	0.6794	0.6833	0.6669
Difference %		1.62%	2.95%	2.72%

6.3 Query-by-query Analysis

Single-term Queries #1, #23, #33, #43 and #53 failed to retrieve any documents using BW1_{NN} in the absence of any stemming technique. Four of these queries feature a *hapax legomenon* term each, that is, they appear only once in the entire corpus and that single occurrence is obviously in the known-item. Whereas the only term in Query #53 has a document frequency equal to three.

The reason for this performance degradation is the imperfect recognition. Another reason is the fact that some terms are rare in the corpus which makes them difficult queries. Although the last one is less difficult with a df=3 as opposed to 1. This is a living proof of the insufficiency of having only one recognition per word, hence the inability of the classical representation on its own to survive the recognition errors in similar cases.

On the other hand, using the multi-hypotheses corpus $\text{BW}\delta_{NN}$, all 180 queries were capable of finding matching documents including the known-item at different ranks. This is another indication supporting that the multi-hypotheses representation ($\text{BW}\delta_{NN}$) is more effective than the conventional one. Also, when using the decompounder with either BW1_{NN} or $\text{BW}\delta_{NN}$ matching documents were found for all queries.

One of the queries reported above is Query #43 *"stens"*. This single-term query did not find any matches when BW1_{NN} (without stemming nor decompounding) was used, while incorporating decompounding contributed in retrieving the known-item at Rank 3. The misrecognition of this term into *"sten"* caused the poor performance of the basic BW1_{NN}. Using the $\text{BW}\delta_{NN}$ representation, with or without decompounding, the known-item is retrieved at Rank 1. This term is represented with four hypotheses in $\text{BW}\delta_{NN}$ with the correct one as the third-likely word hypothesis.

6.4 Summary

In this chapter we conducted a series of experiments with a setup similar to that seen in Chapter 5. Further, the corpus being used is also extracted from the same manuscript (Parzival). However, the recognition system used to generate the transcription uses special Neural Networks (NN) and is totally independent

from the HMM one featured in Chapter 5.

The results we obtained using the NN-based corpus support our findings in the previous chapter. These results emphasize the lucrativeness of using our multi-hypotheses representation over the traditional output of a recognition system which retains only the most probable word transcription. Further, it proves once again that our decompounder results in better retrieval perfomance when combined with either representation.

7

Corrupted Query Retrieval

Older languages, such as Middle High German, tend to be high in both orthographical and morphological variations. Referring to our historical corpus, we revisit our example term *"Parzival"* which appears in the original manuscript as *"Parcifal"*, *"Parcival"* and *"Parzifal"*. These possible variants must be considered as correct spellings as they refer to the same concept or entity. In terms of inflectional morphology, various suffixes were possibly added to nouns, adjectives and names to indicate their grammatical case, for the same running example, *"Parcivale"*, *"Parcivals"* or *"Parcivalen"* are other possible variants.

In practice, when the user tries to recollect a certain term from her/his verse of interest, s/he may be confused about how exactly to spell the query term as it appears in the target text-line. Term confusion may also be due to the user's limited knowledge of the underlying language. Interestingly, it is sometimes not possible even for native German speakers to fully understand the unedited versions of the Parzival poem.

We can classify term confusion into two types:

(a) valid term confusion

(b) out-of-vocabulary term confusion

Also referred to as real-word and non-word errors respectively. Valid term confusion means that the user confuses two different yet valid words (i.e., both of which belong to the underlying language vocabulary). These *cognitive errors* occur due to a high orthographical and/or phonetical similarity between the terms

on hand. For example, the word "center" might be confused with "cents" and both words are valid in the English language vocabulary. The same principle applies to orthographical and morphological variants such as "center" *vs.* "centre" and "center" *vs.* "centering" or "centers" respectively. Confusing homophones is another possible case (e.g., "to", "too" and "two"). An out-of-vocabulary term simply means mistyping a word into one that does not exist in the language on hand. Misspelling "center" as "centar", which does not constitute a valid English word, is an example of an out-of-vocabulary term confusion.

In this dissertation, we will only tackle the problem of cognitive errors. The out-of-vocabulary term confusion problem is a solved one and can simply be resolved with an edit-distance function. With the High German consonant shift[1] or even the influence of modern German or the native/known languages of the user, spelling errors may occur. Taking account of these shifts and Grimm's law[2] (e.g., *"-ph" vs. "-pf* and *"d" vs. "t"*) and the most common foreign language influences (e.g., *"sch" vs. "sh"*) the out-of-vocabulary term confusion problem can completely be overcome.

Also, in practice, dictionaries are used for spelling correction, in our case we can use our *upcycled* thesaurus (presented in the next section) due to the unavailability of an external thesaurus. User logs are also incorporated in commercial systems into query completion and/or suggestion, but once again, this resource is lacking in our case at this stage. In eBay[3] for instance, misspelled search terms play an important role. Some users cannot spell words correctly, as a result, they mistype their own eBay entries (e.g., *"Loubutin"* instead of *"Louboutin"*). Such a problem makes their items hard to find when potential buyers search using the correctly spelled terms. With this in mind, a few specialized Web search tools (spelling mistake spotters such as Fatfingers, Baycrazy, Goofbid and BargainChecker) took advantage of this situation. By *trawling* eBay for all possible orthographical combinations of search terms, they offer a service for their users to find some "hidden bargain gems" on eBay.

In either case, the presence of an inexact term in the user's query can impose a negative effect on retrieval results and thus be very taxing on the final effectiveness. To account for this issue, queries can be *enriched* or *expanded* by adding

[1] http://en.wikipedia.org/wiki/High_German_consonant_shift
[2] http://en.wikipedia.org/wiki/Grimm%27s_law
[3] http://www.ebay.com/

to the query one or more potentially useful terms so as to minimize the damage of possible term confusion. Such damage is especially visible in the performance degradation with very short (single-termed) queries. If the only query term is confused with a different word or have been misspelled, the target text-line might not be found as a possible semantic shift may have occurred. With longer queries however, the damage may be less pronounced since other query terms may contribute to finding the target piece of information.

The rest of this chapter is organized as follows: Sections 7.1 and 7.2 provide a more detailed view of the confusion problem we encounter and present our proposed methods to handle query confusion. Section 7.3 presents our experiments and evaluates the results obtained with the Middle High German corpus. Finally, Section 7.4 samples a few queries and their results for analysis.

7.1 Methodology

Earlier in this dissertation, we presented the traditional query expansion and relevance feedback methods in Section 2.1.4. We also mentioned that query expansion methods can be split into two main classes [36], namely global and local methods. Global methods are independent of the search results and utilize either an external thesaurus or a manual one. Local methods attempt to reformulate the query by taking into account the search results. Both methods can be performed either blindly (i.e., without consulting the user) or alternatively by involving her/him in the expansion process.

Since the individual search items with which we are dealing are relatively short (one text-line, each representing one verse), local methods do not constitute an adequate solution since the expansion may lead to a topical shift that diverges from the user intent. For the same reason, we did not consider the Pseudo-Relevance Feedback scheme presented in Section 4.3. Further, according to Abdou *et al.* [1], [2], neither Rocchio nor their proposed domain specific query expansion techniques resultd in higher retrieval perfomance than with the lack thereof. Global methods, on the other hand, require the use of a thesaurus. Integrating an external thesaurus is out of the question in our case for the following reasons:

(a) Using an external resource might be costly

(b) Such thesauri are rarely available for older languages

(c) The compatibility of external thesauri with our text cannot be guaranteed due to possible orthographical normalization or the lack/presence of some vocabulary

7.1.1 Upcycling: Automatic Thesaurus Generation

Traditionally, and from a text recognition perspective, only the top recognition result per word matters (the most-likely word), the rest is viewed as a by-product and is usually discarded (Section 3.1.2, Table 3.1). But since *one man's trash is another man's treasure* and with the need for a thesaurus and the insufficiency of resources, we re-used the HMM-based recognition results (7-r sets), their frequencies and some of their respective occurrence patterns as raw material to automatically generate a thesaurus for the use of query expansion.

Every word in the underlying language's vocabulary set possesses one entry in our thesaurus. All these entries were generated automatically based on the HMM word recognition results and the occurrences of each word accross the transcription. To create entries to our thesaurus the following procedure is performed: For each word type in the transcription, the respective recognition results (hypotheses) for all of the occurrences of that word type across the entire poem (transcription) are combined according to our upcycling algorithm (an illustrative, toy example is given in Figure 7.1). This word is then logged as an entry into the thesaurus alongside with its variants (analogous to synonyms in conventional dictionaries and thesauri).

It is important to highlight the fact that the absolute log-likelihood values are of less use at this stage. This is due to their being related to the graphical aspect of the handwriting and the shape of the letters, whether the different occurrences were produced by different writers, or the quality of the medium at each occurrence and the possibility of having undergone different sources of noise (hole, stitching, stains, etc.). The words' log-likelihood may retrospectively reflect these factors. However, the ranking within the 7-r sets (which is dependent on the log-likelihood) is of essential importance in the construction of the thesaurus.

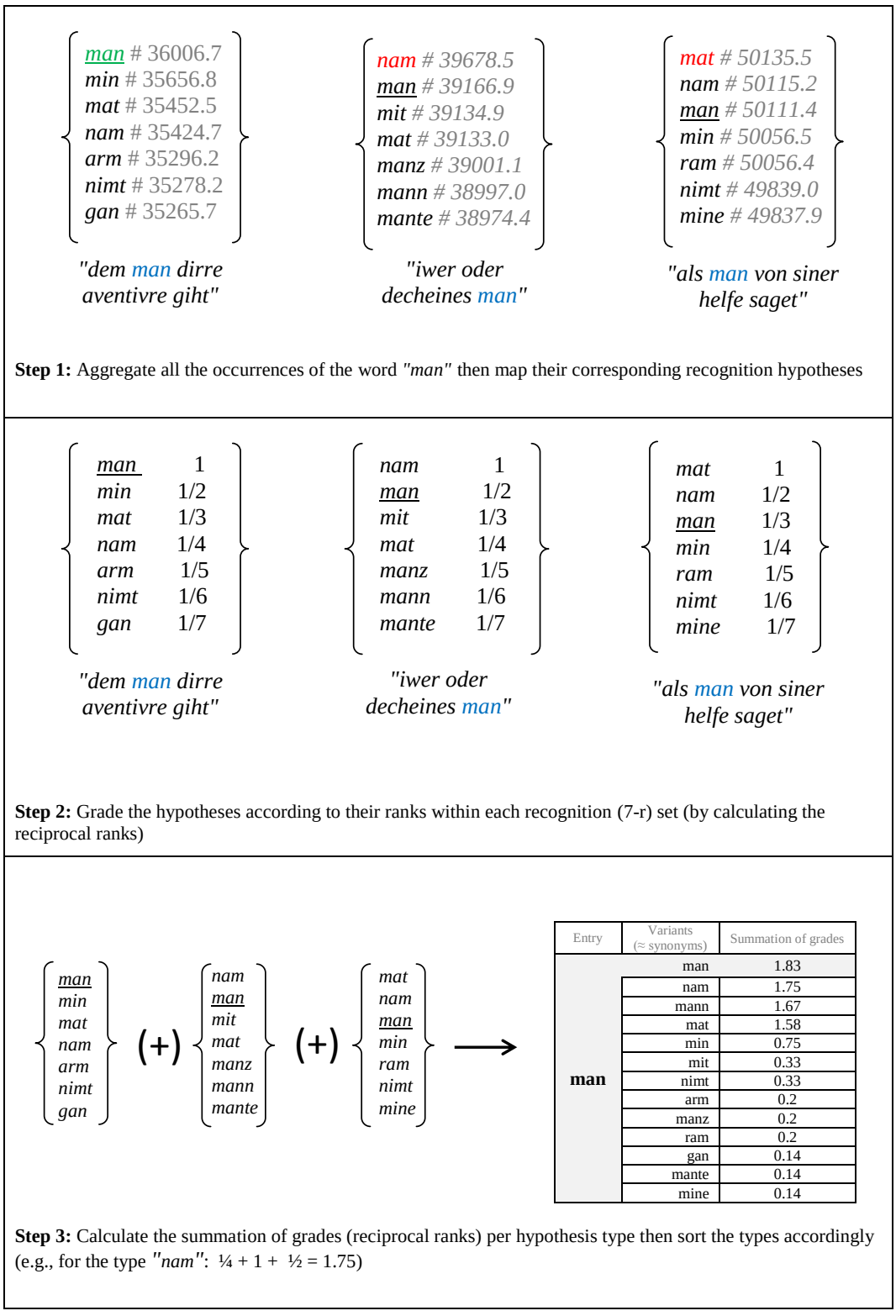

Figure 7.1: Toy example of generating the thesaurus entry for *"man"* based on three of its occurrences only

7.1.2 Thesaurus-based Blind Query Expansion

In order to generate a corrupted version of our original query sets QT1, QT2 and QT3, we simulated the confusion by substituting one query term with its subsequent most likely variant based on the HMM recognition system. We will refer to the corrupted versions as $\overline{QT1}$, $\overline{QT2}$ and $\overline{QT3}$ and the combined set as $\overline{QTAll}$.

Searching the GT, BW1 or $BW1_{NN}$ using these corrupted single-term queries led to a complete failure since the target text-lines (known-items) were not retrieved (RR=zero for all queries). With the multi-hypotheses representation, very few documents were found but in very late rankings. This demonstrates the difficulty yet the importance of this task.

We then passed these corrupted query sets ($\overline{QT1}$, $\overline{QT2}$, $\overline{QT3}$ and their combination $\overline{QTAll}$) to our expansion function in order to produce their expanded versions. We refer to these as $\overline{QT1}+$, $\overline{QT2}+$, $\overline{QT3}+$, and $\overline{QTAll}+$.

Our expansion function is illustrated by an example in Figure 7.2. As can be seen, this query expansion (QE) method succeeded in involving the original clean query term *"ranch"* which will contribute in finding the verse(s) of interest.

We used our upcycled thesaurus to perform query expansion, which clearly falls into the blind global expansion category. Our proposed function constitutes a low-cost, local (massive) query expansion method that approaches the first type of term confusion (valid term confusion).

It is important, in our opinion, to assess this method and how beneficial it is to the final retrieval effectiveness. Therefore, we performed the expansion on the corrupted term only. Hence, the unaltered query terms in $\overline{QT2}$ and $\overline{QT3}$ were not expanded (although we do realize that in practice the system is not supposed to be aware of the corrupted term).

It is noteworthy that the parameters reported in Figure 7.2 are set based on an extended series of experiments that we have conducted (i.e., the depth of reverse lookup variants, the corresponding weighting factor, and the edit distance).

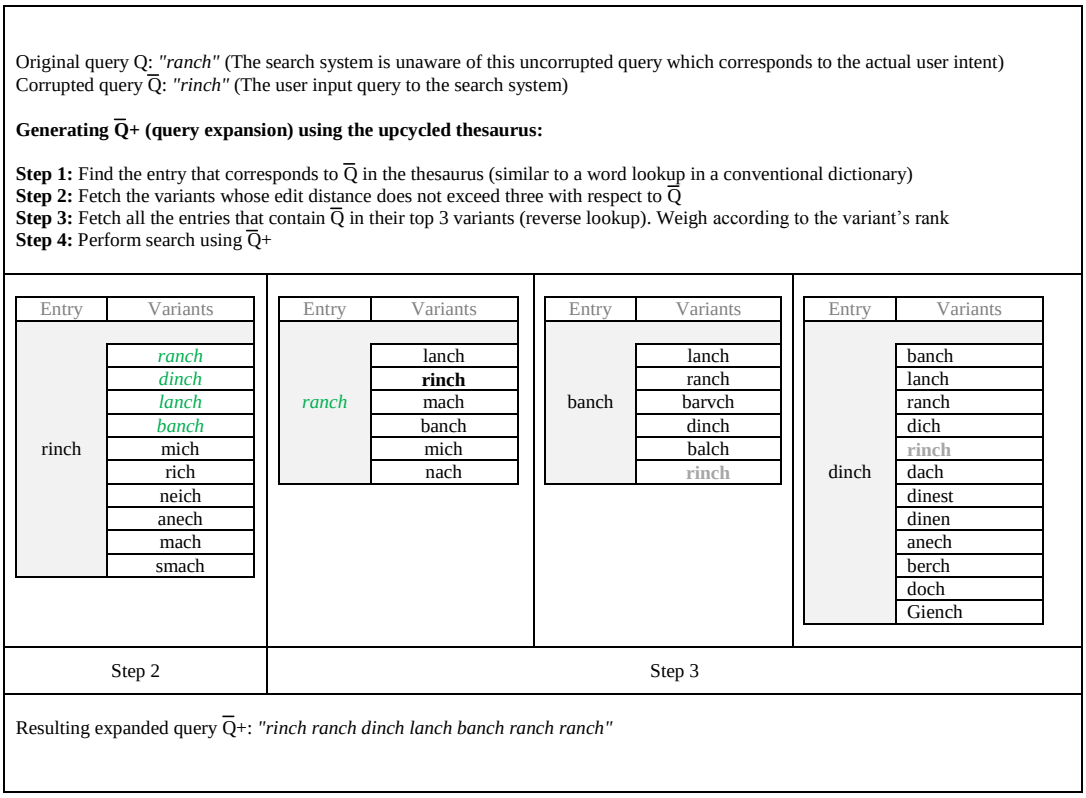

Figure 7.2: An example illustrating our proposed blind query expansion method using the upcycled thesaurus

7.2 Experimental Setup

Single-term queries constitute the most difficult task due to having their only term confused with another and, as a result, are completely corrupted. Long queries, particularly 3-term ones, still retain two guiding terms that have not been altered by confusion. Finally, the case of 2-term queries is quite tricky.

In 2-term queries, confusion may hit one or the other term, it is thus not uncommon that the term originating from the known-item is affected by confusion and the one belonging to the adjacent verse of the known-item remains unaltered. This particular case conveys more challenge than its opposite or than that of 3-term queries.

We used all these queries to search the multiple versions of our corpus. First of all we performed this using the GT (error-free) corpus. We also used the BW1, BW1$_{NN}$, BWδ and BWδ_{NN} corpora.

In terms of IR models and indexing strategies, we continued our tradition of using six models (Okapi, DFR-I(n_e)B2, DFR-PL2, Lnu-ltu, LM and *tf·idf*) *vs.* whole-word (no stemming) and our decompounding techniques.

7.3 Evaluation

The system performance is measured using Graded Reciprocal Rank (GRR). Over the whole set of queries, the mean over all queries is calculated (GMRR). It is known by now that this measure is high (i.e., close to one) if the respective known-item appears in the top ranks. For a given query, if the known-item was not retrieved in the top 1000 answers (retrieved documents), the GRR is equal to zero for this query. This case was experienced with single-term queries suffering from confusion ($\overline{QT1}$) resulting in an overall GMRR equal to zero with the unnormalized GT, BW1 and BW1$_{NN}$, very close to zero with BWδ_{NN} and less severely with decompounding or BWδ (see Tables 7.1, 7.2 and Tables 7 through 16 in Appendix A).

When using BW1$_{NN}$ with no stemming, the confusion resulted in very poor results; six queries did not match with any document (one of which is a 2-term query and the rest are single-term ones). Moreover, none of the other single-term queries and three 2-term queries retrieved their corresponding known-item at any

rank earlier than 1000. The corrupted 3-term query set, however, suffered less than the two other sets.

Using the decompounder instead of the no-stemming representation introduced some enhancement. First of all, all the queries found matching documents. Secondly, twenty-six single-term queries retrieved their known-item at ranks ranging between 4 and 363 as opposed to later than 1000 (GRR=0) in the former case. Longer queries returned their known-items in mostly earlier ranks than with the no-stemming setting.

With the similar setup above (confusion queries, no stemming), the BWδ_{NN} results outdo those of BW1$_{NN}$ having only two single-term queries without any matches. Four single-term queries retrieved their documents at ranks as early as Rank 2.

With the decompounder, we observed a more pronounced enhancement having all queries of all lengths found their known-items in reasonably early ranks. This is another proof of the benefit of the multi-hypotheses representation especially when the decompounder is integrated.

The reported GMRR results show that the query expansion method clearly improved the performance levels for the single-term queries for both GT (the error-free version), BWδ and BWδ_{NN} corpora. However, performing the expansion for longer queries may sometimes lower the overall performance (Tables 12 and 16, Appendix A) and can thus be viewed as redundant or unnecessary. We mentioned earlier that in these queries, the expansion was purposely performed for the corrupted terms only, which, nonetheless was too overwhelming for the retrieval. We expect a more degraded performance if the expansion were to be implemented over all the query terms (i.e., including the non-confused terms, too). We, therefore, do not recommend expanding queries that are longer than one term.

Table 7.1: GMRR for the error-free corpus GT with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted query set $\overline{QTAll}$ (180 queries) and its expanded version $\overline{QTAll}+$ (180 queries)

	GT, None		GT, Dec	
IR Model	$\overline{QTAll}$	$\overline{QTAll}+$	$\overline{QTAll}$	$\overline{QTAll}+$
Okapi	0.3153	0.5442	0.3366	0.5033
DFR-PL2	0.3159	0.5448	0.3379	0.5031
DFR-I(n_e)B2	0.3177	0.5209	0.3341	0.5033
LM ($\lambda = 0.35$)	0.3195	0.5492	0.3417	0.4985
LNU-ltu	0.3192	**0.5499**	0.3414	0.5046
tf·idf	0.3189	0.5321	0.3299	**0.5147**
Difference %		70%	6.04%	58.81%

Table 7.2: GMRR for the error-free corpus GT with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted single-term query set $\overline{QT1}$ (60 queries) and its expanded version $\overline{QT1}+$ (60 queries)

	GT, None		GT, Dec	
IR Model	$\overline{QT1}$	$\overline{QT1}+$	$\overline{QT1}$	$\overline{QT1}+$
Okapi	0.0000	0.6171	0.0324	0.5376
DFR-PL2	0.0000	0.6185	0.0332	0.5447
DFR-I(n_e)B2	0.0000	0.5897	0.0333	**0.5620**
LNU-ltu	0.0000	0.5814	0.0340	0.4926
LM ($\lambda = 0.35$)	0.0000	**0.6199**	0.0330	0.5272
tf·idf	0.0000	0.5937	0.0303	0.5419
Average	0.0000	0.6034	0.0327	0.5343

7.4 Query-by-Query Analysis

Single-term queries are the most negatively affected by term confusion as these queries failed to retrieve their respective known-item or, sometimes, even match any other document at all. Performing query expansion is thus mandatory in this case. Along expansion, we can perform decompounding if BW1 or BW1$_{NN}$ are used, or otherwise stick to the no-stemming representation when using BWδ or BWδ_{NN}. The combination of the multi-hypotheses representation with both decompounding and expansion can overwhelm the retrieval process and result in poorer results instead of enhanced ones.

The longer the query, the less necessary the expansion is. With 3-term queries espacially, expansion becomes redundant and less beneficial. This is also true in the case of the various noisy corpora (BWx) as well as the clean corpus (GT).

When looking at the combined query set QTAll and particularly its corrupted version, we can conclude that in general any slight processing contributes to a performance improvement, whether it is expansion, decompounding, multi-hypotheses or some of their combinations. By examining the respective tables in Appendix A, we can conclude that vector space models (*tf·idf* and Lnu-ltu) tend to provide the best effectiveness levels almost all the time.

A successful example of our expansion function is given in Figure 7.22 *"rinch"* wherein the expansion method succeeded in retrieving the known-item at early ranks for different corpora.

We think that it is interesting to inspect Query #6 as a somewhat non-mainstream query. The only query term *"liehte"* was substituted with the term *"brehte"* due to (simulated) term confusion. Clearly, the confusion term does not relate morphologically to the original term. Traditionally, this confusion is resolved with a simple Levenshtein distance function. However, we think it is quite interesting to examine the behaviour of our expansion function on this query.

The corrupted version of this query failed to retrieve the known-item when the clean corpus GT was used regardless of the indexing scheme. This is also true for the BWδ_{NN} and BW1$_{NN}$ corpora. The query was then automatically expanded. During this process, the original term *"liehte"* was incorporated back into the query (in addition to other terms such as *"bliche"*, *"drvchte"*, and *"brahte"*) and consequently the known-item was retrieved at Rank 1 (without stemming) and

at Rank 2 (with decompounding). This shows that despite the beneficial effect of decomponding on retrieval effectiveness in general, it can sometimes introduce unwanted effect on the query that can cause a relatively late arrival of the target document.

7.5 Summary

In this chapter we introduced our methods that (help to) solve the problem of user confusion while querying due to her/his false memories or unfamiliarity with the underlying language.

Traditional expansion approaches are not always affordable/applicable. Further, text normalization methods such as stemming and decompounding may contribute to resolving confusion in the case of grammatical variants but are not sufficient for more complicated cases of confusion.

Single-term queries are especially critical. We showed that expanding such queries beforehand is necessary to avoid possible poor retrieval performance in the case of term confusion. Longer queries, on the other hand, have more redundancy and can thus be relatively more robust, and hence should only expanded when needed.

To perform the query expansion, we proposed an algorithm to create a local thesaurus which we later incorporated in our propsed expansion method. The results show an improvement of up to 70% compared to the unexpanded confused queries. This can be interpreted as that the user is finding the target item in average at a rank as high as somewhere between the first and the second ranks.

8

Conclusions

8.1 Contributions

(a) We developed a low-cost, highly-effective, special multi-hypotheses representation of text recognition results. This method provides more robustness in retrieval compared the traditional single-hypothesis output of a recognition system in terms of coping with the recognition errors.

(b) We developed text analysis tools for the Middle High German Language, namely, a decompounder and an aggressive stemmer. However, the decompounder has shown to tremendously outperform its text analysis counterparts especially the aggressive stemmer. Nonetheless, these analysis tools can assist the decompounder in achieving better performance results.

(c) We upcycled the recognition results (that are usually discarded) into a rich resource, precisely a thesaurus, that we used for query expansion to resolve the problem of query term confusion. This thesaurus can further be utilized in practice for query suggestion/autocompletion and correction.

(d) We developed a query expansion method that is capable of achieving high effectiveness levels especially with very short (single-term) queries. These queries are particularly problematic in case of confusion at which time this

expansion methods proved to be essential for resolving the problem.

(e) We extended the classical Mean Reciprocal Rank (MRR) measure and its *rigid* and *relaxed* variants into one that can handle graded relevance. We refer to our measure as Graded Mean Reciprocal Rank (GMRR) which in turn has a variation that we call Best-Rank-Oriented GMRR (broGMRR). We proved that the classical, *rigid* and *relaxed* MRR are special cases of both of our extensions.

8.2 Challenges and Limitations

Dealing with the lesser known older languages encompasses several difficulties that particularly burden information retrieval, these are presented briefly in the following list:

(a) These old languages and scripts are only known to (understandable by) the experts of medieval languages and the related fields who represent a small group of people. Therefore, manually transcribing the ground-truth is very time-consuming and resource-demanding. At the same time, the ground-truth is a pre-requisite for the learning phase of the automatic recognition process. Furthermore, the ground-truth constitutes the evaluation baseline in information retrieval that is necessary to evaluate the various schemes until solid conclusions can be drawn regarding which schemes are best for which scenarios in practice.

(b) There is very few literature originating from this era of time; newspapers did not exit yet, as a result, all we have on hand are small sets of pages to build our test-collections whose size falls far below the (relatively massive) modern standard test-collections.

(c) The Internet is of very limited use when it comes to resources related to old languages. Modern online resources such as thesauri are thus less useful if of any use at all. Google books, for instance, are only based on scanned, OCR'ed printed books that originally date to quite recent eras of time (no

earler than AD 1500).

(d) There is a lack of real user queries and logs. Such material is extremely useful for improving effectiveness and for reflecting real-life experiences which in turn contributes to building more robust retrieval tools.

Empirical test-collections usually provide a set of users' topical information needs and their assessments with respect to the results returned. When building our own test-collection, we had to automatically create these query sets and their corresponding assessments since acquiring them from real users was not feasible due to the high costs associated to this process. However, we aimed at finding and adopting the proper algorithm and adapting it to our needs. We also attempted to simulate real user needs and assessments with the highest achievable fidelity in order to obtain results that are reliable and realistic enough to draw conclusions that can be applied to further applications. In the subject of simulating user interaction-related data, simulating the user behavior in certain situations was also taken into consideration such as providing different query structures and simulating user false memories when querying (Chapter 7) as well as when assessing results (Section 3.2.2).

8.3 Conclusions and Take Away Lessons

Correcting recognition errors and normalizing unstandardized orthography are very costly processes. We believe that the associated expenses can be limited or even eliminated and what is needed instead are better practices and representations (which are far less expensive) in order to survive an error rate of 6% for word recognition and 5% for OCR in IR.

We analyzed retrieval effectiveness degradation when facing with two noisy versions of a corpus (having error rates of around 5% and 20%) originating from an OCR device and printed Modern English text. Based on six different retrieval models, three indexing strategies and three stemming approaches, there is a relative decrease of more than 20% when having an error rate of around 5%. When indexing documents and queries using n-gram, we can recover some loss in retrieval quality. From these experiments, we found that our best-performing IR system (LM model combined with the 4-gram representation) tends to perform

around 20% better than the best TREC-5 results when processing a corpus whose recognition error rate is 5% (0.6631 *vs.* 0.5737).

When the error rate increases to 20%, this degradation ranges in average between −47% and −65%. From a statistical point of view, these performance differences are always significant. During these experiments, the Divergence from Randomness (DFR) probabilistic family and Language Model (LM) resulted in the best performance levels.

To reduce the degradation, we decided to skip the removal of certain suffixes. This solution has shown to provide similar results as those obtained after the removal of the plural suffix "-s", and the use of a more aggressive stemmer does not show any significant performance differences.

As an attempt to enhance the effectiveness, we applied an automatic query expansion using Rocchio's pseuro-relevance feedback, but based on three versions of our corpus, we conclude that this technique does not provide any significant improvement. The results of this analysis show that with an error rate of around 5%, the standard IR models may still result in good overall retrieval effectiveness (around −20% of relative quality loss). When the recognition error rate is limited to around 5%, applying an n-gram representation seems to provide better performance than does a word-based indexing approach. This latter scheme is, however, still the best indexing scheme when facing with a clean corpus. When the error rate increases to 20%, the usual techniques (light stemmer, aggressive stemmer, no-stemmer, blind query expansion) are not enough to compensate for the decreased retrieval effectiveness.

As for the historical transcription in Middle High German, our special multi-hypotheses representation has proven to provide superior results to those obtained using the tradiditonal output of a text-recognition system, that is, the best word hypothesis. Both the HMM-based and the NN-based systems supported this finding. We also found that including too many recognition hypotheses per word is not the remedy for surviving recognition error. This is true for BW3 and BW7 which obviously do not constitute good surrogates especially with their lack of the discriminative nature of BWδ between likely- and less likely-correct word recognitions. Since the threshold at which representative word hypotheses are kept is fixed for both BW3 and BW7 alike, terms with relatively low log-likelihood val-

ues were nonetheless included, thus forcing likely incorrect word recognitions into the text representation. Eventually, this resulted in more harm (noise) than improvement (enrichment) in terms of IR performance. With the δ representation, this noise is quite limited (but not completely eliminated though) by discarding word recognitions having relatively low log-likelihood (or probability for the case of NN) values of being correct.

In terms of text representations, our decompounder wich also performs ultralight stemming and variable n-gram-like functionalities has shown to be one of the best representations followed by whole-word representation without any stemming and then by an n-gram representation with n=4.

We also simulated the real-life user behaviour of having false memories in the form of corrupted queries. In order to overcome this query term confusion problem, we developed a query expansion function which makes use of the recognition results. We believe that our query expansion method is essential for single-term queries as they constitute a challenging task and having them confused is crucial or rather detrimental to retrieval effectiveness. This is illustrated by the average effectiveness results being very close or even equal to zero in the case of corrupted single-term queries. However, with longer, partially corrupted queries, we do not encourage using the expansion methods as it tends to add more noise than does it contribute to enriching the queries.

We cannot clearly determine the best retrieval model since they tended to produce similar results most of the time. But generally speaking, we detected a tendancy of Okapi and the Divergence from Randomness (DFR) probabilistic family members to perform quite well. The vector space models *tf·idf* and Lnu-ltu, tended to perform best when query exansion was used or when decompounding and BWδ_{NN} were paired together.

In terms of representations and indexing strategies, we recommend the following practices:

(a) Using the multi-hypotheses (δ) representation instead of the conventional output of a recognition system whenever possible.

(b) Expanding single-term queries beforehand for safer and potentially better results especially in the absence of normalization or decompounding.

(c) Avoiding the implicit expansion of longer queries and only performing expansion if the initial query fails or in the case of a positive opt-in by the user (in interactive systems).

(d) Avoiding overdone implementations. That is, to avoid integrating too many possible enhancement all at the same time. For instance, when already using BWδ with expansion, adding in decompounding to this formula can be too overwhelming and quite harmful to the final performance. On the other hand, when BW1 is the only option, implementing both decompounding and expansion can be very fruitful.

(e) Although not practically attempted, but based on our experiments we think that the 4-gram strategy might be useful in the case of heterogenous corpora to be searched via one IR system, i.e., corpora generated from using different recognition techniques (OCR, word-based), and/or ones written in a certain language each (e.g., English and German). An additional aspect is the rather cheap implementation of 4-gram since no external resources (e.g., thesaurus) are needed.

8.4 Future Work

The main motivation of this dissertation is to contribute to cultural heritage preservation and enable easy access and effective search tools to users. Our indexing and retrieval system is capable of handling huge corpora in reasonably short time and can thus handle these tasks.

We developed several methods that are capable of effectively limiting the negative impact of the various degradation factors, for instance, by overcoming or at least coping with the recognition error and language non-standardization. The question that remains unanswered is that will these methods and practices perform as well or at least degrade slightly and gracefully when faced with higher error rates? For the moment, we know that the n-gram scheme on its own is not powerful enough to survive 20% character error-rate. But we believe that it is

curious to see how well our multi-hypotheses representation or query expansion methods will perform when facing with recognition error rates higher than the experienced 6%.

To answer this quesion, further experiments on further corpora should be performed using the different recognition systems. This, however, is not an easy task due to the aforementioned lack of historical test-collections. We have already conducted a series of experiments with a small Medieval Latin test collection (both word-based HMM and Line-based NN transcriptions, and an error-free version). Preliminary results show that the multi-hypotheses approach performs as well as it did with the Parzival corpora. However, it takes a few trial and error attempts to select the best δ but this can be automatically resolved in no time.

In addition to historical documents, other noisy corpora (with short documents, in particular) can also relate to this dissertation and its contributions, such as microblog retrieval, image and table captions, short text messages (SMS) and sponsored retrieval among others. In such cases, however, document and/or query expansion using thesauri and other external resources usually available on the World Wide Web (WWW) are proven to be effective approaches.

We would also like to see the multi-hypotheses representation utilized with further applications such as speech recognition. Also, seeing how fruitful these methods will be with different language families is another future goal.

Appendix A: Additional Retrieval Results

Table 1: GMRR for the BW1 corpus with six IR models and six indexing strategies, with no-stemming as a baseline "None", single-term queries (QT1, 60 queries)

IR Model	Word-based			4-gram	Dec	Trunc-4
	None	Light	Agg			
Okapi	**0.6971**	**0.6251**†	0.5839†	0.6598	0.6891	0.5591†
DFR-I(n_e)B2	**0.6971**	**0.6251**†	0.5839†	0.6598	0.6849	0.5554†
DFR-PL2	**0.6971**	**0.6251**†	0.5839†	0.6598	0.6896	0.5554†
Lnu-ltu	**0.6971**	0.6249†	0.5837†	0.6509	**0.6947**	**0.5885**†
LM ($\lambda = 0.35$)	**0.6971**	**0.6251**†	0.5839†	0.6515	0.6882	0.5362†
tf·idf	0.6727	0.6150†	**0.5890**†	**0.6766**	0.6666	0.5480†
Difference %		−10.05%	−15.63%	−4.80%	−1.08%	−19.62%

Table 2: GMRR for the BW1 corpus with six IR models and six indexing strategies, with no-stemming as a baseline "None", 2-term queries (QT2, 60 queries)

IR Model	Word-based			4-gram	Dec	Trunc-4
	None	Light	Agg			
Okapi	0.6465	0.6297	0.6105	0.6074	0.6549	0.5424†
DFR-I(n_e)B2	**0.6558**	**0.6398**	**0.6178**	0.6333	0.6382	0.5689†
DFR-PL2	0.6549	0.6352	0.6133	0.6076	**0.6604**	0.5549†
Lnu-ltu	0.6454	0.6295	0.6073	0.6122	0.6329	0.5682†
LM ($\lambda = 0.35$)	**0.6558**	0.6395	0.6175	0.6158	0.6590	0.5689†
tf·idf	0.6451	0.6367	0.6036	**0.6385**	0.6575	**0.5751**
Difference %		−2.39%	−5.98%	−4.84%	−0.02%	−13.46%

Table 3: GMRR for the BW1 corpus with six IR models and six indexing strategies, with no-stemming as a baseline "None", 3-term queries (QT3, 60 queries)

	Word-based					
IR Model	None	Light	Agg	4-gram	Dec	Trunc-4
Okapi	0.6859	0.6492	0.6555	0.6686	0.6924	**0.6442**
DFR-I(n_e)B2	0.6903	0.6562	0.6528†	0.6708	**0.7056**	0.6235†
DFR-PL2	**0.6915**	0.6454†	0.6516†	0.6690	0.6965	0.6428†
Lnu-ltu	0.6903	0.6537	**0.6597**†	0.6708	0.6870	0.6379†
LM ($\lambda = 0.35$)	0.6903	0.6563†	0.6596	**0.6722**	0.6965	0.6404†
tf·idf	0.6771	**0.6583**	0.6495	0.6625	0.6674	0.6164†
Difference %		−5.00%	−4.76%	−2.70%	0.49%	−7.76%

Table 4: GMRR for the BWδ corpus with six IR models and six indexing strategies, single-term queries (QT1, 60 queries)

	Word-based					
IR Model	None	Light	Agg	4-gram	Dec	Trunc-4
Okapi	0.7166	0.6568†	0.6030†	0.7251	0.7342	0.5591†
DFR-I(n_e)B2	0.7166	0.6518†	0.5950†	0.7238	0.7321	0.5554†
DFR-PL2	0.7166	0.6518†	0.5950†	0.7233	0.7348	0.5554†
Lnu-ltu	0.7275	0.6647†	0.6210†	0.7011	0.7089	**0.5885**†
LM ($\lambda = 0.35$)	0.7166	0.6449†	0.5951†	0.7114	0.7103	0.5362†
tf·idf	**0.7447**	**0.6790**†	**0.6365**†	**0.7343**	**0.7397**	0.5480†
Difference %		−8.98%	−15.97%	−0.46%	0.49%	−22.96%

Table 5: GMRR for the BWδ corpus with six IR models and six indexing strategies, 2-term queries (QT2, 60 queries)

	Word-based					
IR Model	None	Light	Agg	4-gram	Dec	Trunc-4
Okapi	**0.6346**	**0.6364**	**0.6145**	**0.6475**	**0.6688**	**0.5745**
DFR-I(n_e)B2	0.6113*	0.6105	0.5820	0.6028*	0.6135*	0.5488†
DFR-PL2	0.6054	0.6006*	0.5781*	**0.6475**	0.6590†	0.5594
Lnu-ltu	0.6290	0.6291	0.6020	0.6238*	0.6389	0.5645†
LM ($\lambda = 0.35$)	0.6182	0.6207	0.5960	0.6284	0.6375	0.5701
tf·idf	0.6054	0.6015	0.6071	0.6277	0.6240	0.5401
Difference %		−0.14%	−3.36%	1.99%	3.72%	−9.35%

Table 6: GMRR for the BWδ corpus with six IR models and six indexing strategies, 3-term queries (QT3, 60 queries)

	Word-based					
IR Model	None	Light	Agg	4-gram	Dec	Trunc-4
Okapi	0.6706	**0.6586**	0.6528	0.6471	**0.6785**	0.6503
DFR-I(n_e)B2	0.6161*	0.6145	0.6183	0.6473	0.6399	0.6336
DFR-PL2	**0.6718**	0.6567	**0.6554**	**0.6542**	0.6778	**0.6528**
Lnu-ltu	0.6544	0.6315	0.6379	0.6393	0.6515	0.6240†
LM ($\lambda = 0.35$)	0.6647	0.6485	0.6466	0.6504	0.6711	0.6479
tf·idf	0.6313	0.6146*	0.6193*	0.6176*	0.6267*	0.6057*
Difference %		−2.16%	−2.01%	−1.36%	0.93%	−2.42%

Table 7: GMRR for the the error-free corpus GT with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted single-term query set $\overline{QT2}$ (60 queries) and its expanded version $\overline{QT2}+$ (60 queries)

	GT, None		GT, Dec	
IR Model	$\overline{QT2}$	$\overline{QT2}+$	$\overline{QT2}$	$\overline{QT2}+$
Okapi	0.4385	0.4673	0.4635	0.4538
DFR-PL2	0.4441	0.4677	0.4605	0.4487
DFR-I(n_e)B2	0.4448	0.4762	0.4805	0.4608
Lnu-ltu	0.4463	0.4960	0.4593	0.4845
LM ($\lambda = 0.35$)	0.4448	0.4712	0.4633	0.4504
tf·idf	0.4418	0.4638	0.4678	0.4814
Average	0.4434	0.4737	0.4658	0.4632
Difference %		6.84%	5.06%	4.48%

Table 8: GMRR for the the error-free corpus GT with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted single-term query set $\overline{QT3}$ (60 queries) and its expanded version $\overline{QT3}+$ (60 queries)

	GT, None		GT, Dec	
IR Model	$\overline{QT3}$	$\overline{QT3}+$	$\overline{QT3}$	$\overline{QT3}+$
Okapi	0.5074	0.5481	0.5140	0.5186
DFR-PL2	0.5035	0.5481	0.5198	0.5160
DFR-I(n_e)B2	0.5082	0.4967	0.4884	0.4869
Lnu-ltu	0.5112	0.5723	0.5310	0.5367
LM ($\lambda = 0.35$)	0.5138	0.5565	0.5288	0.5180
tf·idf	0.5149	0.5388	0.4918	0.5210
Average	0.5098	0.5434	0.5123	0.5162
Difference %		6.59%	0.48%	1.25%

Table 9: GMRR for BW1 and BWδ with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted query set $\overline{QTAll}$ (180 queries) and its expanded version $\overline{QTAll}+$ (180 queries)

	BW1, None		BW1, Dec		BWδ, None		BWδ, Dec	
IR Model	$\overline{QTAll}$	$\overline{QTAll}+$	$\overline{QTAll}$	$\overline{QTAll}+$	$\overline{QTAll}$	$\overline{QTAll}+$	$\overline{QTAll}$	$\overline{QTAll}+$
Okapi	0.2994	0.5261	0.3136	0.4702	0.4118	0.4322	0.4054	0.4122
DFR-PL2	0.2973	0.5268	0.3164	0.4723	0.4093	0.4186	0.3918	0.4060
DFR-I(n_e)B2	0.2976	0.5000	0.3121	**0.4777**	0.3935	0.4042	0.3774	0.3911
LM ($\lambda = 0.35$)	0.3027	0.5314	0.3200	0.4689	0.4066	0.4021	0.3959	0.3854
Lnu-ltu	0.3032	**0.5375**	0.3116	0.4751	0.3857	0.4431	0.4006	0.4017
tf·idf	0.2959	0.5094	0.3111	0.4754	0.3685	**0.4788**	0.3675	**0.4527**
Difference %		74.35%	4.96%	58.11%	32.27%	43.60%	30.22%	36.36%

Table 10: GMRR for BW1 and BWδ with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted single-term query set $\overline{QT1}$ (60 queries) and its expanded version $\overline{QT1}+$ (60 queries)

	BW1, None		BW1, Dec		BWδ, None		BWδ, Dec	
IR Model	$\overline{QT1}$	$\overline{QT1}+$	$\overline{QT1}$	$\overline{QT1}+$	$\overline{QT1}$	$\overline{QT1}+$	$\overline{QT1}$	$\overline{QT1}+$
Okapi	0.0000	0.5903	0.0310	0.5037	0.1667	0.4721	0.1661	0.4517
DFR-PL2	0.0000	0.5931	0.0318	0.5080	0.1667	0.4664	0.1642	0.4481
DFR-I(n_e)B2	0.0000	0.5827	0.0319	0.5329	0.1667	0.4587	0.1671	0.4319
LM ($\lambda = 0.35$)	0.0000	0.5986	0.0316	0.4933	0.1667	0.4545	0.1742	0.4343
Lnu-ltu	0.0000	0.5809	0.0328	0.4611	0.1651	0.4752	0.1623	0.4078
tf·idf	0.0000	0.5596	0.0298	0.4981	0.1571	0.5428	0.1488	0.4915
Average	0.0000	0.5842	0.0315	0.4995	0.1648	0.4783	0.1638	0.4442

Table 11: GMRR for BW1 and BWδ with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted 2-term query set $\overline{QT2}$ (60 queries) and its expanded version $\overline{QT2}+$ (60 queries)

	BW1, None		BW1, Dec		BWδ, None		BWδ, Dec	
IR Model	$\overline{QT2}$	$\overline{QT2}+$	$\overline{QT2}$	$\overline{QT2}+$	$\overline{QT2}$	$\overline{QT2}+$	$\overline{QT2}$	$\overline{QT2}+$
Okapi	0.3955	0.4574	0.4128	0.4080	0.4902	0.3964	0.4723	0.3763
DFR-PL2	0.3955	0.4574	0.4159	0.4081	0.4686	0.3831	0.4418	0.3626
DFR-I(n_e)B2	0.4017	0.4387	0.4391	0.4172	0.4861	0.3402	0.4438	0.3445
LM ($\lambda = 0.35$)	0.4017	0.4569	0.4253	0.4068	0.4875	0.3668	0.4529	0.3466
Lnu-ltu	0.4033	0.4735	0.4107	0.4295	0.4568	0.3974	0.4670	0.3645
tf·idf	0.3939	0.4370	0.4178	0.4185	0.4293	0.4097	0.4422	0.3949
Average	0.3986	0.4535	0.4203	0.4147	0.4697	0.3823	0.4533	0.3649
Difference %		13.77%	5.43%	4.03%	17.84%	−4.10%	13.73%	−8.46%

Table 12: GMRR for BW1 and BWδ with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted 3-term query set $\overline{QT3}$ (60 queries) and its expanded version $\overline{QT3}+$ (60 queries)

	BW1, None		BW1, Dec		BWδ, None		BWδ, Dec	
IR Model	$\overline{QT3}$	$\overline{QT3}+$	$\overline{QT3}$	$\overline{QT3}+$	$\overline{QT3}$	*QT3+*	$\overline{QT3}$	$\overline{QT3}+$
Okapi	0.5026	0.5352	0.5001	0.4994	0.5799	0.4326	0.5707	0.4126
DFR-PL2	0.4963	0.5347	0.5048	0.5013	0.5943	0.4103	0.5621	0.4113
DFR-I(n_e)B2	0.4909	0.4838	0.4680	0.4830	0.5281	0.4186	0.5217	0.4014
LM ($\lambda = 0.35$)	0.5062	0.5437	0.5062	0.5073	0.5668	0.3894	0.5617	0.3788
Lnu-ltu	0.5063	0.5633	0.4945	0.5357	0.5359	0.4624	0.5654	0.4373
tf·idf	0.4937	0.5322	0.4887	0.5102	0.5196	0.4878	0.5117	0.4712
Average	0.4994	0.5322	0.4937	0.5062	0.5541	0.4335	0.5489	0.4188
Difference %		6.57%	−1.13%	1.36%	10.96%	−13.18%	9.92%	−16.14%

Table 13: GMRR for BW1$_{NN}$ and BWδ_{NN} with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted query set $\overline{QTAll}$ (180 queries) and its expanded version $\overline{QTAll}+$ (180 queries)

	BW1$_{NN}$, None		BW1$_{NN}$, Dec		BWδ_{NN}, None		BWδ_{NN}, Dec	
IR Model	$\overline{QTAll}$	$\overline{QTAll}+$	$\overline{QTAll}$	$\overline{QTAll}+$	$\overline{QTAll}$	$QTAll+$	$\overline{QTAll}$	$\overline{QTAll}+$
Okapi	0.2914	0.5073	0.3176	0.4724	0.3103	0.5200	0.3399	0.4774
DFR-PL2	0.2910	0.5078	0.3269	0.4695	0.2988	0.5051	0.3326	0.4774
DFR-I(n_e)B2	0.2932	0.4800	0.3250	0.4687	0.3078	0.4996	0.3324	0.4903
LM ($\lambda = 0.35$)	0.2946	0.5115	0.3307	0.4727	0.3109	0.5057	0.3406	0.4686
Lnu-ltu	0.2953	0.5168	0.3281	0.4755	0.3217	0.5346	0.3365	0.4833
tf·idf	0.2942	0.4989	0.3087	0.4852	0.3061	0.5139	0.3092	0.4979
Difference %		71.38%	9.64%	61.85%	5.44%	75.63%	12.66%	65.60%

Table 14: GMRR for BW1$_{NN}$ and BWδ_{NN} with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted single-term query set $\overline{QT1}$ (60 queries) and its expanded version $\overline{QT1}+$ (60 queries)

	BW1$_{NN}$, None		BW1$_{NN}$, Dec		BWδ_{NN}, None		BWδ_{NN}, Dec	
IR Model	$\overline{QT1}$	$\overline{QT1}+$	$\overline{QT1}$	$\overline{QT1}+$	$\overline{QT1}$	$\overline{QT1}+$	$\overline{QT1}$	$\overline{QT1}+$
Okapi	0.0000	0.5806	0.0348	0.5239	0.0131	0.5900	0.0503	0.5259
DFR-PL2	0.0000	0.5820	0.0347	0.5302	0.0131	0.5621	0.0505	0.5217
DFR-I(n_e)B2	0.0000	0.5444	0.0348	0.5289	0.0131	0.5687	0.0506	0.5659
LM ($\lambda = 0.35$)	0.0000	0.5811	0.0344	0.5246	0.0131	0.5581	0.0506	0.5208
Lnu-ltu	0.0000	0.5471	0.0362	0.4832	0.0139	0.5658	0.0464	0.5089
tf·idf	0.0000	0.5516	0.0302	0.5248	0.0139	0.5864	0.0351	0.5590
Average	0.0000	0.5645	0.3420	0.5193	0.0133	0.5719	0.4720	0.5337

Table 15: GMRR for BW1$_{NN}$ and BWδ_{NN} with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted 2-term query set $\overline{QT2}$ (60 queries) and its expanded version $\overline{QT2}+$ (60 queries)

	BW1$_{NN}$, None		BW1$_{NN}$, Dec		BWδ_{NN}, None		BWδ_{NN}, Dec	
IR Model	$\overline{QT2}$	$\overline{QT2}+$	$\overline{QT2}$	$\overline{QT2}+$	$\overline{QT2}$	$QT2+$	$\overline{QT2}$	$\overline{QT2}+$
Okapi	0.4023	0.4292	0.4308	0.4086	0.4189	0.4408	0.4298	0.4317
DFR-PL2	0.4023	0.4293	0.4420	0.4021	0.4105	0.4359	0.4285	0.4175
DFR-I(n_e)B2	0.4085	0.4335	0.4635	0.4168	0.4216	0.4320	0.4454	0.4276
LM ($\lambda = 0.35$)	0.4085	0.4341	0.4442	0.4116	0.4272	0.4346	0.4452	0.4144
Lnu-ltu	0.4103	0.4585	0.4328	0.4370	0.4373	0.4847	0.4422	0.4393
tf·idf	0.4016	0.4243	0.4270	0.4402	0.4085	0.4396	0.4224	0.4489
Average	0.4056	0.4348	0.4401	0.4194	0.4207	0.4446	0.4356	0.4299
Difference %		7.20%	8.50%	3.40%	3.71%	9.61%	7.39%	5.99%

Table 16: GMRR for BW1$_{NN}$ and BWδ_{NN} with six IR models, two indexing strategies (no stemming and decompounding), and the corrupted 3-term query set $\overline{QT3}$ (60 queries) and its expanded version $\overline{QT3}+$ (60 queries)

	BW1$_{NN}$, None		BW1$_{NN}$, Dec		BWδ_{NN}, None		BWδ_{NN}, Dec	
IR Model	$\overline{QT3}$	$\overline{QT3}+$	$\overline{QT3}$	$\overline{QT3}+$	$\overline{QT3}$	$\overline{QT3}+$	$\overline{QT3}$	$\overline{QT3}+$
Okapi	0.4719	0.5123	0.4900	0.4848	0.4990	0.5293	0.5396	0.4745
DFR-PL2	0.4708	0.5123	0.5069	0.4764	0.4728	0.5173	0.5189	0.4929
DFR-I(n_e)B2	0.4711	0.4622	0.4794	0.4603	0.4888	0.4980	0.5011	0.4775
LM ($\lambda = 0.35$)	0.4752	0.5192	0.5166	0.4820	0.4923	0.5243	0.5259	0.4706
Lnu-ltu	0.4755	0.5448	0.5184	0.5068	0.5138	0.5532	0.5209	0.5016
tf·idf	0.4810	0.5210	0.4716	0.4906	0.4960	0.5157	0.4701	0.4856
Average	0.4742	0.5120	0.4971	0.4835	0.4938	0.5229	0.5127	0.4838
Difference %		7.95%	4.83%	1.95%	4.13%	10.27%	8.12%	2.01%

Appendix B: Algorithms

```
If the word ends with '-ies' but not with '-eies' nor '-aies'
   then replace the  '-ies' with a  '-y',
end;
If the word ends with '-es' but not with '-aes', '-ees' nor '-oes'
   then replace the  '-es' with an '-e', end;
If the word ends with an '-s' but not with '-us' nor with '-ss'
   then delete the  '-s';
end.
```

Figure 1: Harman's Light Stemming Algorithm [25]

Appendix B

```
Perform multi-tiered aggressive stemming in the following order (skip step
whenever inapplicable, apply stemming incrementally using the form
resulting from the previous applicable step if any)

1- STEP ne#3#
2- STEP ge#3#(en/t)
3- STEP 1
4- STEP ge#3#(en/t)
5- STEP 2
6- STEP ge#3#(en/t)
7- STEP 3
8- STEP 1
9- STEP ge#3#(en/t)
10- Return resulting stem

Where:

STEP ne#3#: If the word starts with 'ne-' AND the stem is 3 characters or
longer

     then remove the 'ne-'

  end;

STEP ge#3#(en/t): If the word starts with 'ge-' AND ends with '-en' OR '-t'
AND the stem is 3 characters or longer

     then remove the 'ge-' AND remove the '-en' or '-t'

  end;

STEP 1: (If the word ends with one of the following suffixes '-ern', '-em',
'-er', '-es', '-ez'*, '-iu', '-iv'*, '-e', or '-s' OR ends with '-en' but
does not start with 'ge-') AND the stem is 3 characters or longer

     then remove the suffix

  end;

STEP 2: (If the word ends with one of the following suffixes '-er', '-st',
or '-et' OR ends with '-en' but does not start with 'ge-') AND the stem is
3 characters or longer

     then remove the suffix

  end;

STEP 3: (If the word ends with one of the following suffixes '-lichkeit',
'-igkeit', '-erlich', '-erheit', '-igend', '-igung', '-isch', '-lich', '-
heit', '-keit', '-ung', '-end', '-ik', '-ig', '-ic', or '-ec' OR ends with
'-enlich' or '-enheit' but does not start with 'ge-') AND the stem is 3
characters or longer

     then remove the suffix

  end;


Suffixes denoted with (*) are particular to the Middle High German Language
```

Figure 2: Our Middle High German n-tiered Aggressive Stemming

Bibliography

[1] S. Abdou and P. Ruck and J. Savoy. Evaluation of Stemming, Query Expansion and Manual Indexing Approaches for the Genomic Task. The Fourtheenth Text REtrieval Conference Proceedings (TREC 2005), 863–871. 81

[2] S. Abdou and J. Savoy. Searching in Medline: Query expansion and manual indexing evaluation. Information Processing and Management, 44(2):781–789, 2008. 81

[3] M. Akasereh, N. Naji, and J. Savoy. UniNE at CLEF 2012: CHiC (Cultural Heritage in CLEF). 2012. 24

[4] G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems, 20(4):357–389, 2002. 14

[5] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: Ad hoc retrieval results since 1998. In proceedings of ACM-CIKM, Hong Kong, pages 601–609, 2009. 47

[6] L. Azzopardi and M. de Rijke. Automatic Construction of Known-Item Finding Test Beds. In proceedings of ACM SIGIR, The ACM Press, New York, pages 603–604, 2006. xv, 33, 34

[7] J. P. Ballerini, M. Bchel, R. Domering, D. Knaus, B. Mateev, E. Mittendorf, P. Schuble, P. Sheridan, and M. Wechsler. SPIDER retrieval system at TREC-5. In proceedings of TREC-5, NIST Publication, Gaithersburg (MD), 500-238, 1997. 19, 20, 44

[8] P. Borlund. The Concept of Relevance in IR. Journal of the American Society for Information Science and Technology, 54(10):913–925, 2003. 15

[9] C. Buckley, A. Singhal, M. Mitra, and G. Salton. New retrieval approaches using SMART. In proceedings of TREC-4, NIST Publication, Gaithersburg (MD), (500-236):25–48, 1996. 11, 50

[10] C. Buckley and E. Voorhees. Retrieval system evaluation. In E. M. Voorhees, D. K. Harman, (eds) TREC, experiment and evaluation in information retrieval, The MIT Press, Cambridge (MA), pages 53–75, 2005. 35

[11] J. Callan and M. Connell. Query-Based Sampling of Text Databases. Information Systems, 19(2):97–130, 2001. 33

[12] J. Callan, P. Kantor, and D. Grossman. Information retrieval and OCR: From converting content to grasping meaning. SIGIR forum, 36:58–61, 2002. 19

[13] O. Chapelle and Y. Zhang. Expected Reciprocal Rank for Graded Relevance. In proceedings of CIKM-09. The ACM Press, New York, 36:621–630, 2009. 37

[14] K.W. Church and R.L. Mercer. Introduction to the Special Issue on Computational Linguistics Using Large Corpora. Computational Linguistics, 19(1):1–24, 1993. 45

[15] W. S. Cooper. A denition of relevance for information retrieval. Information Storage and Retrieval, 7:19–37, 1971. 15

[16] H. Craig and R. Whipp. Old Spellings, New Methods: Automated Procedures for Indeterminate Linguistic Data. Literary & Linguistic Computing, 25(1):37–52, 2010. 20

[17] S. Cucerzan and E. Brill. Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of the Web Users. In proceedings of EMNLP., pages 293–300, 2004. 20

[18] K. Eguchi, K. Oyama, E. Ishida, N. Kando, and K. Kuriyama. Overview of the Web Retrieval Task at the Third NTCIR Workshop. NII Publication, Tokyo, 2003. 37, 40

[19] A. Ernst-Gerlach and N. Fuhr. Generating search term variants for text collections with historic spellings. In 28th European Conference on Information Retrieval Research (ECIR), 2006. 17, 20, 21

[20] C. Fautsch, J. Savoy, and L. Dolamic. UniNE at Domain-Specific IR CLEF 2008. Scientific Data Retrieval: Various Query Expansion Approaches. 2008. 55, 56, 58, 69

[21] A. Fischer. Handwriting Recognition in Historical Documents. Ph.D. dissertation, University of Bern, 2012. 71, 72

[22] A. Fischer, M. Wthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehhauser, and M. Stolz. Automatic Transcription of Handwritten Medieval Documents. 15th International Conference on Virtual Systems and Multimedia, 2007. 29, 31

[23] A. Gardt, U. Hauss-Zumkehr, and T. Roelcke. Sprachgeschichte als Kulturgeschichte. Walter de Gruyter, Berlin, 1999. 18, 56

[24] S. M. Harding, W. B. Croft, and C. Weir. Probabilistic retrieval of OCR degraded text using n-grams. In proceedings of the ECDL97, Springer, Heidelberg, pages 345–359, 1997. 20

[25] D. Harman. How effective is suffixing. Journal of the American Society for Information Science, 42:7–15, 1991. xv, 19, 44, 46, 55, 111

[26] A. Hauser, M. Heller, E. Leiss, K. U. Schulz, and C. Wanzeck. Information Access to Historical Documents from the Early New High German Period. In Digital Historical Corpora- Architecture, Annotation, and Retrieval, 2007. 17, 20, 21

[27] D. Hiemstra. Using Language Models for Information Retrieval. CTIT Ph.D. Thesis, 2000. 15

[28] C. Jordan, C. Watters, and Q. Gao. Using Controlled Query Generation to Evaluate Blind Relevance Feedback Algorithms. In the sixth ACM/IEEE-CS Joint Conference on Digital Libraries, The ACM Press, New York, pages 286–295, 2006. 33

[29] P. Kantor and E. Voorhees. Report on the TREC-5 Confusion Track. In TREC, 1996. 25

[30] P. Kantor and E. Voorhees. The TREC-5 Confusion Track: Comparing retrieval methods for scanned text. IR Journal, 2:165–176, 2000. 25

[31] J. Kekäläinen. Binary & Graded Relevance in IR Evaluations - Comparison of the Effects on Ranking of IR Systems. Information Processing and Management, 41(5):1019–1033, 2005. 37

[32] R. Keller. Die Deutsche Sprache und ihre historische Entwicklung. Helmut Buske Verlage, Hamburg, 1995. 17

[33] K. Kukich. Techniques for automatically correcting words in texts. ACM Computing Surveys, pages 377–439, 1992. 20

[34] A. Lam-Adesina and G. Jones. Using String Comparison in Context for Improved Relevance Feedback in Different Text Media. In SPIRE, pages 229–241, 2006. 21

[35] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. Computer Speech and Language, 14(4):373–400, 2000. 72

[36] C. D. Manning, P. Raghavan, and H. Schtze. Introduction to Information Retrieval. Cambridge University Press, Cambridge (UK), 2008. 15, 17, 44, 81

[37] P. McNamee and J. Mayfield. Character n-gram tokenization for European language text retrieval. IR Journal, 7:73–97, 2004. 20, 47, 59

[38] A. Miller, G. The Magical Number Seven, Plus or Minus Two Some Limits on Our Capacity for Processing Information. Psychological Review, 101(3):343–352, 1956. 40

[39] M. Mitra and B.B. Chaudhuri. Information retrieval from documents: A survey. IR Journal, 2:141–163, 2000. 19

[40] E. Mittendorf and P. Schuble. Information retrieval can cope with many errors. IR Journal, 3:189–216, 2000. 19

[41] H. Paul. Mittelhochdeutsche Grammatik, 23rd edition, edited by Wiehl, P and Grosse, S. Niemeye, 1989. 17

[42] C. Peters, J. Gonzalo, M. Braschler, and M. Kluck. CLEF 2003. LNCS, Springer, Heidelberg, 3237(2):179–186, 2004. 20

[43] T. Pilz. Unscharfe Suche in Textdatenbanken mit nichtstandardisierter Rechtschreibung am Beispiel von Frakturtexten zur Nietzsche-Rezeption. Staatsexamensarbeit, Universitat Duisburg-Essen, 2003. 17

[44] T. Pilz, W. Luther, N. Fuhr, and U. Ammon. Rule-Based Search in Text Databases with Nonstandard Orthography. Literacy & Linguistic Computing, 21(2):179–186, 2006. 20

[45] M. F. Porter. An algorithm for suffix stripping. Program, 14:130–137, 1980. 19, 44, 45, 46, 55

[46] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton (ed.), The SMART Retrieval System, Englewood Cliffs (NJ): Prentice-Hall Inc., 3:313–323, 1971. 50

[47] S. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at TREC. Information Processing and Management, 36(1):95–108, 2000. 13

[48] S. Robertson and H. Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends®in Information Retrieval, 3(4):333–389, 2009. 13

[49] S. E. Robertson. The probability ranking principle in IR. Journal of Documentation, 33(4):295–304, 1977. 12

[50] T. Sakai. Ranking the NTCIR Systems Based on Multi-grade Relevance. In proceedings of: AIRS. LNCS #3411, Springer-Verlag, Berlin, pages 251–262, 2004. 37

[51] T. Sakai. Bootstrap-Based Comparison of IR Metrics for Finding One Retrieval Document. In proceedings of: AIRS. LNCS #4182, Springer-Verlag, Berlin, 1(2):374–389, 2006. 37

[52] T. Sakai. On the Reliability of Information Retrieval Metrics Based on Graded Relevance. Information Processing and Management, 43(2):531–548, 2007. 37

[53] G. Salton. The SMART retrieval system. Experiments in automatic document processing. Prentice-Hall Inc., Englewood Cliffs (NJ), 1971. 9

[54] J. Savoy. Statistical Inference in Retrieval Effectiveness Evaluation. Information Processing & Management, 33(4):495–512, 1997. 46, 58

[55] A. Singhal, D. D. Choi, J.and Lewis, and F. Pereira. AT&T at TREC-7. In Proceedings of: TREC -7, Gaithersburg (MA), NIST., pages 239–251, 1999. 11

[56] C. Strohmaier, C. Ringlstetter, K. U. Schulz, and S. Mihov. A visual and interactive tool for optimizing lexical postcorrection of OCR results. In proceedings of: the IEEE Workshop on Document Image Analysis and Recognition, DIAR03, 2003. 20

[57] J. Strunk. Information retrieval for languages that lack a fixed orthography. Technical report, Linguistics Department, Stanford University, 2003. 21

[58] K. Taghva, J. Borsack, and A. Condit. Results of applying probabilistic IR to OCR text. In proceedings of the ACM-SIGIR, the ACM Press, New York, pages 202–211, 1994. 19, 44

[59] K. Taghva, J. Borsack, and A. Condit. Evaluation of model-based retrieval effectiveness with OCR text. ACM Transactions on Information Systems, 14:64–93, 1996. 19, 44

[60] K. Taghva, T. Nartker, and J. Borsack. Information access in presence of OCR errors. In Proceedings of the ACM Workshop on Hardcopy Document Processing, the ACM Press, New York, pages 1–8, 2004. 19

[61] K. Taghva and E. Stofsky. OCRSpell: an interactive spelling correction system for OCR errors in text. International Journal of Document Analysis and Recognition, 3:125–137, 2001. 20

[62] E. Voorhees and J. Garfolo. Retrieving noisy text. In E.M. Voorhees, D. K. Harman, (eds) TREC, experiment and evaluation in information retrieval, The MIT Press, Cambridge (MA), 2:183–197, 2005. 25

[63] E. M. Voorhees. Evaluation by Highly Relevant Documents. Proceedings ACM SIGIR, The ACM Press, New York, pages 74–82, 2001. 37

[64] P. Wilson. Situational relevance. Information Storage and Retrieval, 9:457–469, 1973. 15

[65] M. Wthrich, M. Liwicki, A. Fischer, E. Indermhle, H. Bunke, G. Viehhauser, and M. Stolz. Language model integration for the recognition of handwritten medieval documents. In proceedings of the 10th Int. Conf. on Document Analysis and Recognition, 1:211–215, 2009. 72

[66] J. Zobel and P. Dart. Finding approximate matches in large lexicons. SoftwarePractice and Experience, 25(3):331–345, 1995. 21