

# The Anatomy of a Search and Mining System for Digital Humanities

Search And Mining Tools for Language Archives  
(SAMTLA)

**Martyn Harris**

A dissertation submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy



Computer Science and Information Systems

Birkbeck, University of London

United Kingdom

August 2016

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Digital archives and the Humanities . . . . .	15
1.2	Problem definition . . . . .	17
1.3	Contributions . . . . .	22
1.4	Thesis Structure . . . . .	27
<b>2</b>	<b>Critical Review</b>	<b>28</b>
2.1	Overview . . . . .	29
2.2	Research in an “age of abundance” . . . . .	30
2.3	Barriers to tool adoption in the Humanities . . . . .	35
2.4	Information seeking in the Humanities . . . . .	42
2.5	Exploring digital archives . . . . .	46
2.6	Discussion . . . . .	53
<b>3</b>	<b>Architecture</b>	<b>57</b>
3.1	Overview . . . . .	57
3.2	The framework . . . . .	58
3.3	The case studies . . . . .	64
3.4	Discussion . . . . .	74
<b>4</b>	<b>Search</b>	<b>76</b>
4.1	Overview . . . . .	77
4.2	Statistical Language Models (SLM) . . . . .	78
4.3	Document index . . . . .	80
4.4	Query results ranking . . . . .	86
4.5	Smoothing . . . . .	91
4.6	Metadata search . . . . .	97
4.7	Case studies . . . . .	100

---

4.8	Discussion . . . . .	105
<b>5</b>	<b>Text Mining</b>	<b>106</b>
5.1	Overview . . . . .	107
5.2	Metadata . . . . .	109
5.3	Recommendation tools . . . . .	129
5.4	Document comparison tool . . . . .	143
5.5	Named entity tool . . . . .	150
5.6	Discussion . . . . .	156
<b>6</b>	<b>User Interface</b>	<b>158</b>
6.1	Overview . . . . .	159
6.2	The interface structure . . . . .	161
6.3	Search . . . . .	164
6.4	Browsing . . . . .	170
6.5	Document view . . . . .	173
6.6	Document comparison . . . . .	176
6.7	Discussion . . . . .	178
<b>7</b>	<b>Evaluation</b>	<b>180</b>
7.1	Overview . . . . .	181
7.2	Crowdsourcing . . . . .	182
7.3	Methodology . . . . .	184
7.4	Evaluation Results . . . . .	197
7.5	Discussion . . . . .	207
<b>8</b>	<b>Conclusions and Future Work</b>	<b>212</b>
8.1	Summary of the Thesis . . . . .	212
8.2	Summary of the Contributions . . . . .	217
8.3	Future work . . . . .	218
<b>Appendix A</b>	<b>Architecture</b>	<b>221</b>
A.1	The Model . . . . .	221
A.2	The controller . . . . .	221
A.3	The view . . . . .	222

---

<b>Appendix B Evaluation</b>	<b>223</b>
B.1 The test queries . . . . .	223
B.2 Evaluation queries . . . . .	226

# LIST OF FIGURES

3.1	The <i>Samtla</i> architecture. . . . .	60
3.2	An Aramaic Magic Bowl for protection against demons, 6th century AD, ©The Trustees of the British Museum [21]. . . . .	66
3.3	“The House that Jack Built”, Moon, M. Harris. Published 1821, London. Printed for Harris and Son, corner of St. Paul’s Church-yard. . . . .	69
3.4	A single page from the Financial Times newspaper archive for Friday 1st September 1939. . . . .	70
3.5	The inside front cover of the 1611 edition of the King James Bible. . . . .	72
3.6	A portrait of Giorgio Vasari. . . . .	73
4.1	A suffix tree constructed from the string “bananas”. . . . .	82
4.2	Average word length across several language groups represented by Faroese, English, Latin, Hindi, and Hebrew. . . . .	84
4.3	A compressed version of the suffix tree where the super-nodes are rendered as ellipses. . . . .	85
4.4	Calculating the <i>MLE</i> for each order of <i>n</i> -gram stored in the collection model <i>C</i> represented by the suffix tree data structure. . . . .	90
4.5	The search results for a query representing a “parallel passage”, submitted to the <i>Aramaic Magic Bowl archive</i> . . . . .	100
4.6	The search results for the query “American air strikes over North Vietnam”, submitted to the <i>FT newspaper archive</i> . . . . .	101
4.7	The search results for the query “In the beginning was the word”, submitted to the <i>King James Bible</i> . . . . .	102

4.8	The search results for the query “the journal of the royal geographical society”, submitted to the <i>British Library Microsoft archive</i> . . . . .	103
4.9	The search results for the query “Madonna and Child 1380”, submitted to the <i>Giorgio Vasari archive</i> . . . . .	104
5.1	The search filter for a query representing a “parallel passage”, submitted to the <i>Aramaic Magic Bowl archive</i> . . . . .	112
5.2	The search filter for the query “American air strikes over North Vietnam”, submitted to the <i>FT newspaper archive</i> . . . . .	113
5.3	The search filter for the query “New Testament”, submitted to the <i>King James Bible</i> . . . . .	114
5.4	The search filter for the query “war office”, submitted to the <i>British Library Microsoft archive</i> . . . . .	115
5.5	The search filter for the query “the last supper”, submitted to the <i>Giorgio Vasari archive</i> . . . . .	116
5.6	A portion of the browsing tree structure for the <i>King James Bible</i> . . . . .	120
5.7	The layout process with 4 child nodes. Bold arrows indicate the flow of the construction, which preserves the best aspect ratio. . . . .	121
5.8	The vertical list view representing the landing page of the <i>Aramaic Magic Bowl archive</i> . . . . .	123
5.9	The vertical list view representing the landing page of the <i>FT newspaper archive</i> . . . . .	125
5.10	The vertical list view representing the landing page of the <i>King James Bible</i> . . . . .	126
5.11	The vertical list view representing the landing page of the <i>British Library Microsoft archive</i> . . . . .	127
5.12	The vertical list view representing the landing page of the <i>Giorgio Vasari archive</i> . . . . .	128
5.13	A document comparison between two “parallel passages” in the <i>Aramaic Magic Bowl archive</i> . . . . .	147
5.14	A document comparison between two “parallel passages” in the <i>King James Bible</i> . . . . .	148

5.15	A document comparison between two “parallel passages” in the <i>Giorgio Vasari archive</i> . . . . .	149
5.16	The <i>FT newspaper archive</i> named entity view for the original image of the document. . . . .	152
5.17	The named entity view for the <i>Financial Times newspaper archive</i> , with a Google map of the locations mentioned in the document. . . . .	153
5.18	The named entity view for the <i>King James Bible</i> . . . . .	154
5.19	The named entity view for the <i>Giorgio Vasari archive</i> . . . . .	155
6.1	The basic structure of the user interface is composed of a header and main body divided in to left, middle, and right columns. . . . .	162
6.2	The header provides access or settings for the most often used tools represented by search, recommendations, and browsing preferences. . . . .	163
6.3	An example breadcrumb describing the path taken from the root level to the document level of the Bible version of Samtla. . . . .	163
6.4	An example of the related queries for the query “Nebuchadrezzar”, a name of a King mentioned in the King James Bible. . . . .	168
6.5	The left sidebar showing the user query and document history, and the most popular queries and documents in the whole community of users. . . . .	169
6.6	Browsing the Bible corpus using the list view. . . . .	170
6.7	Browsing the <i>King James Bible</i> through the treemap view generated from the document metadata. . . . .	171
6.8	Browsing the photograph category of the <i>Aramaic Magic Bowl archive</i> . . . . .	172
6.9	Browsing the images cited in the text of the <i>Giorgio Vasari archive</i> . . . . .	172
6.10	The document level for the <i>Aramaic Magic Bowls archive</i> , which shows the default metadata view for the document. . . . .	173
6.11	The document level for the <i>FT newspaper archive</i> , which displays the original scanned image of the document. . . . .	174
6.12	The document-level tools. . . . .	175

---

6.13	An example of Samtla document comparison. The document comparison interface shows a pairwise comparison of the target document (left) and a document selected from the list of related documents (right). Sequences highlighted in yellow reflect the currently selected sequence, and blue represents all sequences shared between the two documents. . . . .	179
7.1	Users are assigned to one of five bins, each with their own distribution of queries from <i>Set 1</i> , and <i>Set 2</i> . . . . .	186
7.2	Summary statistics on the successful participants by demographic.	197
7.3	Average relevance score by rank position divided by query set.	203
7.4	Distribution of relevance grades used in the evaluation. . . . .	205
7.5	Average correlation by query partitioned in to <i>Footrule</i> and <i>M-measure</i> . . . . .	210
7.6	Average <i>NDCG</i> by query divided according to discounting function. . . . .	211



# LIST OF TABLES

2.1	An example “parallel passage”, adapted from de Jong (2007). . .	30
2.2	A summary of the research and commercial digital tools discussed in this section. . . . .	52
4.1	The conversion of the string “banana” in to character $n$ -grams.	81
4.2	The lambda weights defining the contribution of the probability for each order of $n$ -gram, to the final approximation of the 5-gram “abcde”. . . . .	94
5.1	A minimal metadata record for the <i>King James Bible</i> . . . . .	118
5.2	Related queries generated for the character-sequence “lord”. . .	134
5.3	<i>Type II related queries</i> : a small set of character rules for the <i>King James Bible</i> that associate characters of the query with old spelling variants extracted from the <i>Tyndale</i> and <i>Wycliffe</i> Bibles. . . . .	137
5.4	The top-ten documents ranked by the value of their prior. . . .	142
5.5	An example shared-sequence between the <i>Douay-Rheims</i> and <i>King James</i> bible, which were written in different forms of the English language. . . . .	144
5.6	An example “parallel passage” shared between two chapters of the <i>King James Bible</i> , which demonstrates the flexibility of the approach to word choice made by two authors. . . . .	145
5.7	The longest sequence shared between two chapters of the Bible, with the differences between the two sequences underlined. . .	148
5.8	The longest sequence shared between two chapters from the work of <i>Giorgio Vasari</i> , with the differences between the two sequences underlined. . . . .	149

6.1	The ranked search snippets generated for a single document matching the query “Hast thou appealed unto Caesar?”, submitted to the <i>King James Bible</i> . . . . .	167
7.1	Calculating the DCG@5 for each rank. . . . .	191
7.2	Calculating the IDCG@5 for each rank. . . . .	192
7.3	Two ranked lists $r_1$ and $r_2$ , with a document $D_4$ at rank 1 of $r_1$ , and rank 2 of $r_2$ . . . . .	195
7.4	Baseline NDCG . . . . .	198
7.5	Average query <i>NDCG</i> for each set of queries according to the <i>SLM</i> rank order of the document with the 95% confidence intervals reported in square brackets. . . . .	198
7.6	Average <i>NDCG</i> for the <i>Set 2</i> queries according to the <i>display</i> rank order of the documents, with the 95% confidence intervals reported in square brackets. . . . .	199
7.7	Average query baseline correlation for each measure, which compares the <i>SLM</i> rank order of the documents to the <i>display</i> rank order for <i>Set 2</i> queries only. . . . .	200
7.8	Average query and user consensus correlation scores for the <i>SLM</i> rank order of the documents, divided by query set. . . . .	200
7.9	Average query and user agreement correlation scores for the <i>Set 2</i> queries in <i>display</i> rank order. . . . .	201
B.1	The fifty queries selected for the formal evaluation. . . . .	227

## ABSTRACT

Humanities researchers are faced with an overwhelming volume of digitised primary source material, and “born digital” information, of relevance to their research as a result of large-scale digitisation projects. The current digital tools do not provide consistent support for analysing the content of digital archives that are potentially large in scale, multilingual, and come in a range of data formats. The current language-dependent, or project specific, approach to tool development often puts the tools out of reach for many research disciplines in the humanities. In addition, the tools can be incompatible with the way researchers locate and compare the relevant sources. For instance, researchers are interested in shared structural text patterns, known as “parallel passages” that describe a specific cultural, social, or historical context relevant to their research topic. Identifying these shared structural text patterns is challenging due to their repeated yet highly variable nature, as a result of differences in the domain, author, language, time period, and orthography.

The contribution of the thesis is a novel infrastructure that directly addresses the need for generic, flexible, extendable, and sustainable digital tools that are applicable to a wide range of digital archives and research in the humanities. The infrastructure adopts a character-level  $n$ -gram Statistical Language Model (*SLM*), stored in a space-optimised  $k$ -truncated suffix tree data structure as its underlying data model. A character-level  $n$ -gram model is a relatively new approach that is competitive with word-level  $n$ -gram models, but has the added advantage that it is domain and language-independent, requiring little or no preprocessing of the document text unlike word-level models that require some form of language-dependent tokenisation and stemming. Character-level  $n$ -grams capture word internal features that are ignored by word-level  $n$ -gram models, which provides greater flexibility in addressing the information need of the user through tolerant search, and compensation for erroneous query specification or spelling errors in the document text. Furthermore, the *SLM* provides a unified approach to information retrieval and text mining, where traditional approaches have tended to adopt separate data models that are often ad-hoc or based on heuristic assumptions. In addition, the performance of the character-level  $n$ -gram *SLM* was formally evaluated through crowdsourcing, which demonstrates that the retrieval performance of

the *SLM* is close to that of the human level performance.

The proposed infrastructure, supports the development of the *Samtla* (Search And Mining Tools for Language Archives), which provides humanities researchers digital tools for search, browsing, and text mining of digital archives in any domain or language, within a single system. *Samtla* supersedes many of the existing tools for humanities researchers, by supporting the same or similar functionality of the systems, but with a domain-independent and language-independent approach. The functionality includes a browsing tool constructed from the metadata and named entities extracted from the document text, a hybrid-recommendation system for recommending related queries and documents. However, some tools are novel tools and developed in response to the specific needs of the researchers, such as the document comparison tool for visualising shared sequences between groups of related documents. Furthermore, *Samtla* is the first practical example of a system with a *SLM* as its primary data model that supports the real research needs of several case studies covering different areas of research in the humanities.

# DEDICATION

This thesis is dedicated to the memory of Nigel Harris, who saw the potential of the ZX Spectrum to inspire, and educate.

# DECLARATION

This thesis is the result of my own work, except where explicitly acknowledged in the text. \_\_\_\_\_ [Martyn Harris].

# ACKNOWLEDGEMENTS

This thesis has involved a lot of hard work on the part of many individuals aside from myself. I would like to acknowledge the support of my wife Ilaria Rovera, who has been extremely patient, supportive, and understanding of the work during the course of this PhD. I would also like to acknowledge Angela, Iain, and Philip Harris, as well as my extended family, for their patience due to my recent absence from family life.

The thesis would not have seen the light of day if it had not been for the role played by my supervisors Prof. Mark Levene, and Dr Dell Zhang who played an important part in getting me to the end of the PhD. I would not have made the transition from the humanities to the computer sciences without your time, dedication, and patience. To Dr Dan Levene whose perspective on life, frankness, and sense of humour during the development of *Samtla* provided much needed light relief. I will always remember our Aramaic lessons with fondness.

Lastly, I have much respect and owe a great deal to Phil Gregg, who provided seemingly instantaneous and extensive technical support. As well as to Adam Towner and Tara Orlanes-Angelopoulou for their very efficient administrative and departmental support. I did not always make it easy for you.

# CHAPTER 1

## INTRODUCTION

This chapter presents a discussion of the need by humanities researchers for flexible, extensible, and sustainable generic digital tools that support search, browsing, and mining of digital content stored in a digital archive. The background, scope, and motivation for the research is discussed in more detail in Section 1.1. The problem domain and contributions of the thesis are introduced in Section 1.2 and Section 1.3, respectively, where a novel approach to digital tool provision that can support the analysis of documents in any domain, language, data-format, is presented. In Section 1.4, the chapter concludes with a summary of the thesis structure.

### 1.1 DIGITAL ARCHIVES AND THE HUMANITIES

Digital representations of original historic objects are being made available as a result of the work undertaken by digital archiving projects [164]. A wealth of digitised objects, represented by text and scanned images, have been published online by a range of international institutions. A recent example is the *Hebrew Manuscripts Digitisation Project* conducted by the British Library [30], which released 1,300 digital scans of Hebrew manuscripts to the public, under a creative commons license [23]. With the increasing number of large-scale digital archives available, humanities researchers now have unprecedented access to electronic editions of primary source material, which offer an opportunity for raising new questions and revisiting old ones [187], as a result of the increased breadth and depth of the topics represented by the documents. Digital tools



were developed in response to provide access and computer-assisted forms of analysis, many of which were in response to a need by specific research projects. Despite the increased availability of tools and their potential for performing new forms of analysis, current research on tool adoption in the humanities reveals that the discipline has not been able to move forward in the way envisioned by the increased availability and range of digital tools [185]. It appears that the tools are not widely adopted by humanities researchers, and the literature suggests that the root cause is attributed to the approach underlying tool development. The existing approaches cause the tools to be tied to specific archives or language corpora, such as the Bible and the Works of Shakespeare. Furthermore, the tools mainly support the dominant and commonly studied languages, such as English. In other words, the approach does not provide the appropriate level of flexibility with respect to the languages, domains, and consequently, digital archives with which the tools can operate.

Many digital tools adopt ad-hoc and language-dependent approaches to the representation and scoring of the  $n$ -grams of the document, which provides very little support for digital archives represented by the little studied and morphologically complex languages like Hopi, Turkish, and Cuneiform, due to a lack of natural language processing tools and resources. Furthermore, the output of digitisation projects, can also present a challenge to information retrieval and mining due to a number of issues, including the following:

- Digital archives are becoming much larger in scale and scope, and may contain multilingual documents, or encompass different literary text genres.
- Natural language is complex with variations in the orthography associated with language-specific syntax, language change over time [158], and differences in dialect.
- The documents may be provided in formats including raw text files, EXtensible Markup Language (*XML*) [45], and *TEI* (TExt Encoding Initiative) [39], which may not be supported by the approach.
- The quality of the digital object is largely determined by the state-of-the-art in scanning and recognition technology at the time they were used, which may not be fully optimised for some document collections, such

as those representing historical documents, resulting in poor character-recognition rates.

The lack of a generic set of flexible tools for search and mining, means that both the current and future demands of various disciplines in the humanities are not being met [98]. A large number of the tools have become “abandonware” [174], which has been linked to the unsustainability of the approaches, and duplicated efforts resulting in a lack of breadth in the type of analysis that can be performed [185]. Furthermore, the tools developed may be considered as “black boxes”, since the underlying implementation details are not documented, or they require some prior knowledge of the literature in a domain outside of the humanities. The thesis presents a novel approach that directly addresses the need for a generalised platform that can support the development of flexible and sustainable digital tools for a wider range of humanities researchers than previously achieved. This is demonstrated by the *Samtla* (Search And Mining Tools for Language Archives) system, and achieved through a language-independent data model that supports the search, and mining of small and large-scale digital archives based on a character-level  $n$ -gram Statistical Language Model (*SLM*) stored in a space-optimised  $k$ -truncated suffix tree data structure.

## 1.2 PROBLEM DEFINITION

Researchers in the humanities are not adopting tools more widely, as a result of several issues identified in the literature (see Chapter 2). One issue is attributed to a lack of awareness, or interest in the potential of computer-assisted forms of analysis. However, many more of the barriers appear to be associated with the approach underlying the way in which tools are developed. These include, the development of project specific tools, usability issues relating to the user interface, and the way the tool interacts with the data, the lack of flexibility to other digital archives, or data formats for which they were not specifically designed, and the incompatibility of the tool to how researchers wish to interrogate the sources.

Many of the barriers can be attributed to the domain and language-dependent nature of the approaches, which tend to rely on normalised versions of the document text, requiring a preprocessing step. One example is

the application of stemming algorithms, which normalise the text by reducing words to a common root form, devoid of any grammatical features. This enables a retrieval model to capture all instances of the same word to produce a more statistically stable model of the terms in the documents. The most widely adopted stemmer, is the *Porter* stemmer [160], which operates over the English language, and requires at least five phases and sixty different rules to identify commonly occurring affixes. In addition, statistical approaches to stemming often require large amounts of training data, which may not always be available for some languages [155]. Preprocessing the documents is becoming unsustainable with the increased availability of multilingual and large-scale digital archives. Furthermore, rule-based stemming approaches may become unreliable when applied to large scale digital archives, due to the difficulty in comprehensively describing a set of rules that can accurately extract the relevant patterns from the document content. Lastly, the documents may contain erroneous strings resulting from *OCR* errors, which can be a problem for word-based approaches to search, which is commonly adopted by the digital tools. When considering how to develop tools that directly address the different needs of researchers in the humanities, Cohen et al.(2009) raised the following questions [80]:

1. Do you try to build a comprehensive tool or one that does something very narrow?
2. A “killer app” to be adopted broadly or a tool to solve a particular problem faced by a specific scholarly community?
3. Is there such a thing as a “killer app” in the humanities, or are tools necessarily discipline-specific?

A system for search and mining of digital archives requires several components, each of which determine the flexibility and performance of the system. The main components include an index for storing the  $n$ -grams of the documents, along with their associated weights. In addition, a retrieval model is required to measure the relevance of each document given  $n$ -grams of the query. Researchers in the humanities require tolerant search tools in order to identify specific contexts, or events through phrase-like search queries. Mining tools, should help to support the comparison of variable length structural text

patterns to facilitate the discovery of related source material discussing the same context or event. However, these basic forms of analysis are not often supported by the tools, which tend to operate on individual documents, or a small subset. Any system for the humanities also needs to be extensible to enable the creation of new tools and features to address the individual needs of specific disciplines in the humanities, including linguistics, history, sociology, literary criticism, and art history. Each of the main components of a system has its own set of considerations and challenges, which are summarised as follows:

### **Indexing**

An index provides a record of all instances of a word or character-sequence for each document stored in a collection. One important issue to consider is the level of representation for the documents, which determines the amount of information we record about the language contained in the documents. The choice of representation is important as it determines how flexible the system will be in identifying full and approximate text patterns in a document. The majority of systems adopt a word-level  $n$ -gram representation, where words are identified by segmenting the text according to a delimiter, such as the whitespace character. However, this approach is language-specific, since languages such as Chinese, have no whitespace equivalent, which makes the task of identifying the morphemes of the language difficult [192]. Furthermore, some languages such as Turkish, attach affixes to a root word, which means that segmenting the text according to the whitespace character will result in many words being *Out Of Vocabulary* (*OOV*) [55, 190], resulting in an inaccurate model of the language. Another approach is to use a character-level model, but this approach has largely been ignored by the research community (see Chapter 2). This is because the word-level representation has performed well for many well-known language corpora, for instance, English, where there are a large number of Natural Language Processing (*NLP*) resources available for preprocessing the text [113]. Character-level approaches have been viewed as unnecessarily complex due to the storage requirements, compared to those of the word-level approach [111]. The appropriate choice of representation, is determined by a number of important considerations:

- Which of the representations provides the most flexibility with respect to the domain and language of the digital archive, which could also be multilingual?
- Does the chosen representation provide a single unified approach to the development of both search and mining tools?
- Is there any flexibility provided by the choice of representation with respect to how the users' information needs are expressed and answered? For example, how does the representation address erroneous query specification by the user, and spelling errors in the document text.
- How do we store the documents efficiently, given the selected representation, so as to provide fast and efficient retrieval of the documents, when querying the index for matching documents.

## Search

A user defines an information need through a query, represented by one or more “words” in the language. Based on the query, a search engine should locate and rank the documents of the collection, in such a way as to provide the most relevant documents at the top of the search results. Relevance describes how well a search engine ranks documents addressing a particular topic, described by the query (see Mizzaro (1997) [148]). The index is the primary data structure underlying search, and is used for retrieving a subset of the documents matching all, or part, of query. A survey of the information-seeking behaviors of historians [183], reveals that the most important resources in archives, are literary works, correspondence, pamphlets, diaries, journals, reports, and government papers, which are all examples of domain-specific document collections, containing their own specialised vocabulary.

This means that search evaluated on one domain, is not necessarily transferrable to another, as the language can be quite different. Users require a flexible querying language that will compensate for erroneous query specification, spelling errors in the document content, and linguistic differences arising from the morphology and syntax of the language and domain, however, these are not often supported by the majority of search tools. The choice of retrieval model is therefore determined by a number of related factors, including:

- Which retrieval model provides the most flexibility with respect to the domain and language of the digital archive?
- How does the retrieval model cope with missing information, such as query over-specification, or spelling errors?
- How should the documents be ranked so that researchers will find the approach intuitive and easy to understand?
- Can the retrieval model be extended to non-traditional tasks other than search?
- To what degree does the retrieval model require ad-hoc or heuristic assumptions when ranking the documents?

### **Mining**

The close-reading and comparison of document content is a fundamental task conducted by researchers, which enables them to summarise the similarities and differences between documents. Typically, the collection is a representative sample of the research topic, and the researcher adopts comparative methods to identify and summarise the content in relation to their research objectives by selecting representative examples of the research topic. Under this context the following questions are raised:

- Is there a generic set of typical mining tools that are widely adopted by researchers of digital archives due to their compatibility with their research methods?
- Where is the gap in terms of the scope of the provided tools? In other words, what types of tools would enable researchers to address new questions that have not been possible with the current set of tools?
- Should mining tools be developed as standalone applications as per the current approach, or as components of a much larger system?
- Can the mining tools be generalised to permit their application to any language or domain, in order to provide a consistent set of features and functionality for any research group or digital archive?

## 1.3 CONTRIBUTIONS

The main contribution of the thesis is a general purpose infrastructure represented by a character-level  $n$ -gram *Statistical Language Model (SLM)* [159, 194], stored in a space-optimised  $k$ -truncated suffix tree data structure [173].

- The infrastructure uses very little information about what natural language is, which makes it very flexible compared to word-level approaches.
- The *SLM* approach has been well studied and shown to perform well for speech recognition and information retrieval tasks [139, 159].
- A principled approach to document representation and term weighting, which is sometimes an ad-hoc or heuristic design decision in other approaches, such as the boolean retrieval model commonly adopted by digital archive providers. The term weights are calculated on the basis of a good statistical foundation, which utilises many well-established statistical, such as the common *Maximum Likelihood Estimator (MLE)* [194].
- The *SLM* is flexible to different smoothing strategies, which are an important component of language models. Smoothing plays two key roles in the retrieval model [197]. The first role is to reduce the influence of terms representing the syntax of the language, which are not good descriptors of the topic defined by the users query. The second role compensates for terms that are missing in the documents, which can occur when the text collection does not cover the topic defined by the query [195], or when users are unfamiliar with the archival contents and unable to specify an appropriate query.
- A *SLM* adapts to the domain and language of the documents, which is often a problem for ad-hoc retrieval.
- Some of the most popular retrieval models such as the vector space model and BM25 are based on a heuristic design approach for the retrieval model. *SLMs*, on the other hand, do not require much in the way of heuristic design due to the probabilistic framework adopted [195].
- The *SLM* is well-suited to modeling non-traditional retrieval problems [195], including machine translation, part-of-speech tagging, syntax parsing, mining, and handwriting recognition. The underlying probabilistic

approach is easily adaptable to special or complex information retrieval tasks compared to other approaches [194], which tend to adopt an ad-hoc or heuristic approach that is often independent of the data model used for search.

- The approach facilitates the identification of textual patterns that are relevant to the discovery and comparison of the document content. These textual patterns representing “parallel passages”, can be quite varied as a result of differences in the language, author, or literary style, but are easily captured through the generation of partial matches to the textual pattern.
- The infrastructure is fully extensible with regards to the range of digital tools that can be developed from the *SLM* data model. The approach presented in the thesis would enable any digital archive to be made accessible relatively quickly online, complete with a set of generic tools that meet the basic needs of humanities researchers for the analysis of unstructured text including the document content, metadata, and the browsing of accompanying image data. Furthermore, the current implementation of the infrastructure provides a solid foundation for developing semantic search and mining tools as part of future work.
- *Space-optimised.* Storing character-level  $n$ -grams is more efficient than word-level models, due to the finite set of possible character combinations in the language, whereas the set of words in a language is potentially infinite as new words are always being introduced to the language, or borrowed from other languages.

The thesis presents the proposed approach in more detail in the coming chapters, where the following novel contributions are described:

- **A unified approach to domain and language-independent search and text-mining of a digital archives.** The character-level representation for the  $n$ -grams of the query and documents has several advantages over the word-level representation including, capturing the sub-word level features to estimate the statistics of the language more accurately. The resulting data model represents a semantic model of the domain and language recorded by the document content. The *SLM*



is well-motivated and supported by a large body of research in speech recognition. Their recent adoption in information retrieval has shown that their performance is on par with more traditional approaches such as the popular Vector Space Model (*VSM*) [142, 144, 194]. Furthermore, the *SLM* can be extended in many novel ways to applications beyond traditional information retrieval, such as text mining, named entity recognition, authorship attribution, and recommendation, which is often supported by ad-hoc or heuristic approaches in more conventional models adopted by the current digital tools.

- **The first practical implementation of a character-level space-optimised *SLM*, stored in a suffix tree data structure, as the underlying data model.** There has been a lot of research, and tool kits developed for the purpose of exploiting *SLMs*, such as the *Lemur Project* [31], however, to the best of our knowledge, there has not been much adoption of a *SLM* as an integral part of a digital tool or system. As far as the author is aware, the proposed infrastructure is the first practical implementation of a character-level *SLM* stored in a *k*-truncated suffix tree structure that directly supports the development of flexible tools to address the needs of real users reflected by several research groups in the humanities, introduced in Chapter 3).
- **A set of innovative mining algorithms.** The algorithms developed for mining are novel as they support the language and domain-independent mining and comparison of variable length text patterns, through adoption of many components of the same data model as that used for information retrieval.

*Samtla* was developed in response to a specific need for digital tools by historians researching digital archives containing historic documents. Through this collaboration a set of key search and mining tools were identified and developed from the infrastructure to provide a set of generic and flexible digital tools to support a number of important tasks that are integral to researchers in the humanities, including the following:

- **Search.**

Fast and tolerant full-text and metadata search using an *SLM* combined

with a character-level index stored as a compressed suffix tree data structure. The character-level representation for the documents supports full and partial query matching through keyword and phrase-like queries representing textual patterns of importance to researchers.

- **Browsing.**

Hierarchically clustered views of the archive constructed from the document metadata and named entities provides researchers with novel ways to browse and explore a range of information across different media formats and stored in digital archives, allowing greater flexibility in how researchers can locate the documents stored in digital archives.

- **Recommendation.**

A hybrid-recommender system constructed from the user activity log data, and the statistics of the language stored in the *SLM*, provides recommended queries and documents to researchers based on the search and browsing behaviour of the whole community of researchers; enabling researchers to identify the interesting parts of the archive. A further component generates recommendations on the basis of the properties of the language such as alternative spellings for the query, and semantically related documents according to the  $n$ -gram probability distribution stored in the *SLM* for the complete archive, and each individual document.

- **Comparison.**

Researchers are able to explore both global and local similarities between the documents through comparison and visual mining of shared-sequences present in semantically similar documents, where semantic similarity is defined by the set of matching  $n$ -grams shared between groups of documents. The tool provides a flexible approach to locating identical, and near-identical patterns, using an adapted version of the *Basic Local Alignment Search Tool (BLAST)*, adopted in bioinformatics.

The proposed infrastructure fulfills the need for a more generic, flexible, and extensible approach to digital tool provision that will support the basic information needs of many disciplines across the humanities. The proposed approach represents the development of an “infrastructure for digital scholarship in the humanities” [65], which supports the current and future research of

digital archives through cross-domain and language-independent digital tools.

## 1.4 THESIS STRUCTURE

In Chapter 2, an analysis of the current approach to tool development is presented, together with a review of the barriers to tool adoption faced by researchers in the humanities. The chapter discusses commonly adopted, or long-standing tools, with a summary of the main findings surrounding their scope, functionality, and limitations inherent in the approaches adopted. In Chapter 3, a description of the infrastructure and architecture supporting the *Samtla* system is presented, which describes the storage of the data model, the communication between the tools and the data model, and an introduction to the research groups that will provide the basis for the case studies. Chapter 4 presents the data model component of the system, including the document representation selected for the index, the retrieval model based on *SLMs*, and several tools developed for the purpose of mining the content of digital archives. The domain and language-independent design of the search tools are demonstrated through a number of case studies that vary with respect to the language, domain, size, format, and quality of the documents. In Chapter 5, the mining tools developed from the underlying data model are presented. The tools were designed to support the research needs of specific research groups who required tools to support the search and exploration of a digital archive through search, browsing of metadata, images, and named entities, recommendation tools, and the comparison of “parallel passages”. The *Samtla* system user interface (*UI*) is presented in Chapter 6. User interface design is an important aspect of any system, and poor design can often result in poor usability, and consequently a lack of adoption or abandonment. The user interface was developed through a collaborative effort involving feedback provided by the research groups. A formal evaluation of the *SLM* data model, is presented in Chapter 7, which involves a system-based evaluation through a crowdsourcing platform. The results are evaluated through a set of well-known non-parametric measures and significance tests. Lastly, Chapter 8 summarises the main contributions of the thesis, and describes the prospects for future research and development of the infrastructure, and *Samtla* system tools.

## CHAPTER 2

# CRITICAL REVIEW

The chapter presents a critical review of the research and literature related to the current provision and adoption of tools for the analysis of digital archives by humanities researchers. Digital archives are highly variable with textual content across many different domains e.g. poetry, ethnographic reports, newspaper articles, and languages, with some being multilingual in nature. In addition, some digital archives such as those held by the National Gallery [33] are largely image-based and the textual content comes as captions and metadata. A digital tool is defined as any software application, which has been developed for the creation, interpretation, or communication of digital resources [198], through the access, search, and mining of electronic media formats. Section 2.1 discusses the recent increase in digital representations of primary source material published online, and their implications for research in the humanities. In Section 2.2, it is argued that the large scale and complexity of digital archives pose a problem for researchers, who lack the necessary tools to access and interrogate the sources. The literature reveals several barriers to a wider adoption of tools in the humanities, which appear to result from the development of highly specialised tools designed for specific forms of analysis, domains, and languages (see Section 2.3). Many of the current approaches produce tools that are not compatible with the way researchers “do research”. Section 2.4 discusses how humanities researchers locate and analyse the documents, with a view to developing tools that model the research approach. Several tools and systems are briefly described in Section 2.5, in order to identify a set of key tools that are currently being used by researchers. Lastly, Section 2.6 discusses the observations drawn from the literature.

## 2.1 OVERVIEW

Humanities research covers a broad range of subdisciplines including, anthropology, literary criticism, cultural history, history of art, philosophy, political science, and gender studies [187]. Research in the humanities involves the interpretation of documentary sources including text, images, illustrations, and audio and video witness accounts. These documentary sources, known as “primary source material”, record cultural contexts that are of value to humanities researchers [187].

Accessing the source material traditionally required a visit to the physical library or archive, and analysis of the sources was mainly performed manually through close-reading of the handwritten, or printed text. Many institutions are now making the content of their archives available online through digital archiving projects. Some of these institutions have partnered with large technology companies such as Microsoft and Google to enable the scanning of thousands of books a day [135]. Increasingly, government bodies, companies, and institutions are promoting the online access of their content and services, and so researchers are finding more of the primary source material relevant to their research being “born digital”, and only available in electronic format, for example, raw text, Portable Document Format (*PDF*), web pages, scanned images, audio, video, 3D models, maps and geo-location data. Furthermore, there is an abundance of derived data generated by online summarisation tools, and user generated tags associated with the material which can be of equal importance to researchers. Section 2.2 describes the emergence of a relatively new discipline, the “Digital Humanities” [163].

The increasing volume of digital source material is now becoming difficult for humanities researchers to manage, and digital tools are now required to support fast and flexible access to information relevant to a variety of information needs. However, Humanities researchers have very specific needs that are not adequately being met by the existing set of digital tools [98]. Researchers analyse the documents by identifying possible interpretations or contexts represented by recurring text patterns found in the archival content [187], such as the distribution of words, and set phrases, known as “parallel passages”. These parallel passages may be duplicated in whole, or in part, across a large number of documents, making their identification challenging as a result of

the domain, authorship, and spelling differences that can exist between two similar texts. The example below, from the *King James Bible*, illustrates two “parallel passages” that would be regarded as highly similar by researchers of the Bible [88], as they discuss the same event. It is generally agreed that *Isaiah, Chapter 7* was derived from *2 Kings, Chapter 16* [88]. However, the similarity between these two texts is not easily identifiable with the current tools developed for search and mining of digital archives, due to the variability in the choice of language.

<i>2 Kings, Chapter 16</i>	<i>Isaiah, Chapter 7</i>
Then came up King Rezin of Aram and King Pekah son of Remaliah of Israel to wage war on Jerusalem; they besieged Ahaz but could not prevail over him.	In the days of Ahaz son of Jotham son of Uzziah, king of Judah came up King Rezin of Aram and King Pekah son of Remaliah of Israel to Jerusalem to attack it,  but could not mount an attack against it.

Table 2.1: An example “parallel passage”, adapted from de Jong (2007).

Several studies in the digital humanities indicate that despite the potential of digital archives as a basis for widening the scope of research in the humanities to enable “new intellectual strategies”, which were previously impossible due to a lack of access to primary source material [187], the development of an appropriate set of digital tools is still very much behind that of the natural sciences [98]. Despite the perceived value of digital source material and optimism surrounding their potential use for research, there is a lack of support with respect to a set of generic tools that operate across domains, languages, and media formats.

## 2.2 RESEARCH IN AN “AGE OF ABUNDANCE”

Digital Humanities is not so much a single discipline, but a group of convergent disciplines represented by historians, linguists, and computer scientists who explore a domain where print is no longer the main medium with which

knowledge is produced and disseminated [74]. The Digital Humanities therefore represents a new set of practices, where the adoption of technology to address research problems is common place [65].

Humanities researchers are facing an increase in the availability of digitised collections of documents important to their research, which is mainly attributable to the efforts of large digitisation projects [164]. The abundance of digital versions of primary source material has promoted the concept of a “library without walls” or “digital library” [101], where the digitised archives are accessible online at any time of day for the purpose of research and public interest. One example is Microsoft’s *Book Search Project* [2], which digitised 68,000 books (25 million pages) covering a range of languages including English, French, Spanish, German, Hungarian, Italian, Russian, and Malagasy; and domains represented by poetry, correspondence, news, technical reports, and media including text, and images of photographs, illustrations, and maps.

Previously the document collections were small in scale, or only available at specific institutions in physical form [187]. Recently, researchers have begun to appreciate the value of digital representations of primary source materials as a means to facilitate faster access and improve the breadth of their analysis. However, the increase of primary source material, in electronic format, means that the humanities is now facing an age of information overload, where it is becoming increasingly difficult for researchers to feel that they have gained a comprehensive overview of the sources relevant to their research (see Section 2.4). Digital archives, therefore, represent an opportunity for researchers to operate with a much larger volume of sources. Looking to the future, there is an even more pressing need for digital tools to support a new generation of researcher who actively seek to adopt computer-assisted analysis tools as an integral part of their research methodology. This is not only motivated by the desire to speed up the analysis [98], but also because an increasing amount of primary source material is now only accessible and understood through digital media [187], for example, the large volume of “born digital” source material [94], reflected by online material hosted on websites, including: images, government papers, newspaper articles, and blogs and twitter data. The volume of primary sources material is now on a much larger scale than the volume of material available to researchers of the past. As noted by Cohen et



al. (2009), tools are not necessarily the solution to humanities research per se, but, when it comes to large-scale archives and electronic formats, digital tools are critical [80], and will become “integral to doing research” in the humanities over the coming decades [81].

Digital humanities researchers recognise the need for a new approach to tool development, as the current language-dependent and domain-specific development of tools has not helped to move the discipline forward in response to the increased availability of primary sources relevant to research. However, the current approach to tool provision has resulted in a wide range of highly specialised tools that are difficult to adapt for the research of digital objects across different domains, languages, media formats, and size of digital archives.

To address these issues, researchers in the humanities have looked to develop a general-purpose “cyberinfrastructure” that would offer support for a variety of digital corpora, and researcher needs by acting as a single point of access to digital tools. The proposed infrastructures increase awareness and accessibility of digital tools, but have, so far, not addressed the immediate problem. That is, the majority of the digital tools are still not generalisable to other digital collections outside of their specified requirements, and consequently require considerable effort on the part of the researcher to adapt them to the needs of their research or discipline.

One example of a cyberinfrastructure is the *TAPoR* project [38], where researchers can access and experiment with a large collection of text mining tools. The majority of the tools are optimised for the English language, often due to the reliance on language-specific preprocessing, or data used for training models. For example, the popular visualisation tool *DocuBurst* [4, 83], uses *WordNet* to produce visual summaries of the semantics of a documents. There are two issues, the first is that the interface only allows a single document to be uploaded at a time for analysis, which makes the analysis a cumbersome and time-intensive process if the collection of documents is large. Furthermore, the reliance on language-dependent technologies that are usually optimised for well known languages like English, means that unless *WordNet* is extended to more languages, there is no provision for other researchers who may wish to take advantage of the tool.

With the increase in large-scale archives, the language and domain-specific approach to tool development is quickly becoming unsustainable. This is because digital archives such as the *British Library Microsoft Book Search Project* corpus, present a number of challenges for search and text mining tasks, which include the following:

- How should the data be structured in order to facilitate search, browsing, and comparison of the content across-domain, language, which is tolerant to the way in which researchers wish to discover and interrogate the content of the archive?
- How can the approach operate consistently across the eight different languages available (English, French, Spanish, German, Hungarian, Italian, Russian, and Malagasy), which are varied in terms of their morphological complexity, time period, author, and domain? Many approaches adopted by existing tools require language-dependent preprocessing of the documents before they can be made available for search or mining.
- How do we compensate for issues associated with the quality of the resulting digital representation? The documents in the British Library Microsoft corpus reflect the quality of the *OCR* technology at the time. Furthermore, historical documents represent a particular challenge for *OCR* technology as a result of non-standard typesetting, and the quality of the original image. This makes some digital objects potentially undiscoverable despite being relevant to the researcher.
- Can the approach meet the same needs of researchers who are interested in a different archive such as the British Library *Million Book Project* [32], which will likely differ in terms of language, domain, and available data, media, and quality of the digital objects, compared to that used when initial developing the approach. Digital content providers may adopt different data-standards and formats for storing the digital objects – often providing only the raw text for download. Furthermore, not all archives will provide consistent data, for example, there may be no metadata provided for the documents, or conversely, there is no text content for the documents e.g. images, but extensive and potentially useful metadata that can be used to search and browse the archive.

For researchers in the humanities, the interrogation of primary source material remains at the heart of their discipline, but researchers are fast moving from a “culture of scarcity”, to a “culture of abundance” [166]. However, when it comes to systems and tools for the purpose of research, these have generally been the product of specific research projects for specific collections, resulting in a number of disparate tools and systems that have not been widely adopted by researchers in the humanities despite their perceived value. With respect to the commercial software available, they are often relevant only to a small niche of researchers, or where there is a market for tools for particular text collection e.g. Bible reading software (some examples are presented in Section 2.5). Furthermore, the tools may only partially support the needs of the researcher, and can be difficult to adapt or extend to a particular research problem [98]. In addition, there are costs involved in purchasing the software, or subscribing to the service [187].

## 2.3 BARRIERS TO TOOL ADOPTION IN THE HUMANITIES

This section discusses some of the common barriers faced by researchers who wish to adopt digital tools to complement their existing research strategies. There have been several studies that have surveyed the current situation with respect to tool provision and uptake by humanities researchers. There is a general consensus that access to digital tools helps to increase the researchers' capacity to conduct research, resulting from the accelerated and much broader access to primary source material stored than ever before [72, 98, 120, 141].

Qualitative studies involving interviews with humanities researchers, have attempted to identify a set of key barriers to tool adoption. These studies show that in reality, not all researchers actually require tools, because their research revolves around a very small and specific collection of well-known documents that can be easily analysed manually. However, for researchers whose research topic covers a much larger volume of primary sources, there is a justifiable need for tools, but researchers are faced with a number of barriers that are currently preventing them from adopting tools more widely [98]. Due to the broad nature of humanities research, it is difficult to describe all the barriers faced by humanities researchers, but there are a number of reasons that appear to be common to the studies.

One of the main barriers faced by researchers is the recent increase in availability of large volumes of electronic data provided by digitisation projects. The current set of tools do not support the analysis of these collections due to a lack of flexibility and generalisability to other domains and languages for which they were not designed [185].

Digital archives are also often provided “as is”, where the researcher can only download the raw files and associated metadata. Whereas the tools are often maintained separately from the archival content [65], which does not facilitate research “on the spot” [94]. Furthermore, even when digital archiving projects are willing to develop tools that operate with the collections, there are often technical challenges that prevent them from “graft[ing] a particular tool onto their collection” [80]. This situation is not helped either, by the fact that humanities researchers do not generally see it as their responsibility to develop the required tools themselves. The assumption has been

that the responsibility lies with the providers of digital archives, or through collaborative efforts involving computer scientists [166]. This creates another issue, however, as there can be a gap in communication between humanities researchers and tool developers who have their own specialised vocabulary to describe the details of their tools. As Gibbs (2012) revealed in their study users wanted the theoretical benefits spelled out in plain language irrespective of [the] discipline [98].

### **Generality of the tools**

Many digital tools have been developed by large digitisation projects as a means to publish the digital editions, meaning they were not actually designed for more general use, but as a necessary means to make available the output of a specific project [163]. These issues are beginning to be resolved, particularly in the digital humanities, where computer programming is more widely-adopted, and researchers are beginning to supplement their existing methods with purpose built tools to handle the digital sources [172].

The advice to tool developers is that they need to understand what users actually want in terms of tool provision, and then develop a tool with the “correct functionality and user interface to meet those needs” [80]. However, although an important consideration, this approach tends to result in tools with limited scope with respect to their functionality, or the corpora with which they can operate [98, 163], because they have been addressed to meet the specific needs of a specific group of researchers studying a specific archive or language. As noted by Unsworth’s review of the past ten years of tool development, “it is not at all uncommon for researchers to develop tailor-made systems that replicate much of the functionality of other systems” [185]. This results in tools that cannot be adopted easily by other researchers, and the end result is an “endless software waste cycle” [185]. Reusability is key in avoiding developers from “re-inventing of the wheel” each time a new tool is developed, which has been the currently adopted strategy for the past three decades [80, 185].

To illustrate the problem, many of the text analysis tools listed on *TAPoR* only support the analysis of individual texts, which makes it difficult to apply the tools to a large volume of documents stored in a digital archive. Further-

more, the tools are often language-dependent, for instance, tools like *Voyant tools* (see Section 2.5) provides researchers with an environment for close-reading and analysis of the documents. However, because the tool adopts a word-level representation for the documents, where the documents are reduced to morphemes, by delimiting the text according to the whitespace character, it does not operate well with languages that do not have an explicit delimiter such as Chinese, and languages that are morphologically complex such as Turkish, Russian, Hebrew, and Arabic (an example of this issue appears in [41]).

Digital tools should therefore, be envisioned as more than just a one-off project, but as a means for supporting “rigorous and long-term scholarship”. If this is to be achieved then infrastructure is key, but there is some evidence suggesting that the accessibility, and quality of a tools is very much correlated with the quality of the supporting infrastructure [174].

### Usability of the tools

Usability of the tools, related to accessibility and user interface design issues, is a further factor that can cause many researchers to abandon their initial attempts at incorporating tools [80, 98, 163]. When tools are invisible, inaccessible, or difficult to understand, researchers are less inclined to adopt them more broadly, less able to extend them, and this affects how well the tool is able to support and extend the research for which they were designed [174].

One example of this issue is the way in which some tools impose a particular structure in terms of how the data is loaded in to the tool, or how a user interacts with the data through the systems interface. The structure imposed often resembles that of the data, such as the tabular format represented by spreadsheets, or tree structures adopted by the HTML and XML formats. These are issues that can be easily resolved, as illustrated by the *Blake* project [46], which made a simple revision to the user interface to allow users to make side-by-side comparisons of source material. Previously users were required to open two web browsers, side-by-side, and navigate a hierarchical file structure in order to select the relevant document for comparison [184]. Consequently, a lack of understanding of the types of tools and interfaces that would be useful to researchers, can have a big impact on wider-adoption [98].

Furthermore, many of these digital archives are of interest to more than one discipline in the humanities, and so many researchers will want to access the same source material, but to address very different research topics [187]. Consequently, tools that provide specific types of output, such as word-frequency statistics, may not be as intuitive to art historians, as they are to corpus linguists or literary critics.

### **Compatibility with research needs**

Where collaboration has taken place, between humanities researchers and tool developers, the results have been mixed. This is attributed to the fact that tool developers do not necessarily appreciate the concerns and goals of their colleagues in the humanities. This is sometimes caused by tool development being “tech-centric” as opposed to “scholar-centric”, where tool developers are more focused on addressing the technical challenges of interest in their field, rather than addressing the specific research needs that could lead to new discoveries in the field of humanities research [80]. Furthermore, researchers are sometimes unable to accurately articulate what tools they actually need in advance, which means the approach tends to fallback to tools that are biased towards “repeating modes and interfaces”, that are not effective for doing digital scholarship, than more “innovative software” [80]. In this respect a better approach would involve users right from the start of the development as part of an iterative and user centered design process.

There is also a tendency for developers to focus on the creation of sophisticated tools that are “in fashion” e.g. social network graphs, and word clouds, but these do not address the basic needs of the researchers [98]. Furthermore, the consensus is that there is little evidence to suggest that humanities researchers actually “take full advantage of the possibilities of more advanced tools” [72,98], especially when their basic needs for a generic set of tools are not being met.

Making use of source material stored in large digital archives, is becoming increasingly difficult when addressed with traditional approaches of the past, where close-reading of the texts was feasible. However, developing tools that simply provide a faster alternative to manual approaches that researchers have been adopting for years, for example, word-frequency counts and concordances,

are useful, but more is needed to help move the discipline forward in a way that enables researchers to pose new questions about history, language, and society that have not been possible until the recent arrival of these vast digital archives [82].

In addition, the results of the survey conducted by Gibbs and Owen (2012), suggest that sophisticated tools are not currently necessary, as the basic needs of researchers have not yet been met. More often than not, researchers are skeptical of “sophisticated” tools, as they tend to restrict the possibility of a more modular approach to tool development, and the adoption of simple, and intuitive tools. Humanities researchers are more in favour of tools that are easy to use, and transparent with respect to how the tool interfaces with the data [98], particularly where the interpretation of the semantics of texts is a key component of the research. Conversely, disciplines focused on the processing of image data, such as the datasets produced by NASA, may not be concerned with the underlying algorithms powering the tools.

### **Awareness of the potential**

So far, the number of researchers in the humanities, who actively adopt tools, is quite small compared to their counterparts in the natural sciences [98]. This is due to the fact that many researchers are not aware of the tools existence or are unable to find a tool that is suited to their research needs. This tends to be the result of a lack of “community building and marketing functions”, which are required if a tool is to experience wide-spread adoption [80].

Humanities researchers sometimes find it difficult to identify the actual benefits of adopting digital tools in the first place, or they have difficulty interpreting the results generated by algorithms that are unfamiliar to them [62], which can be attributed to the way in which the tools are marketed to researchers, where it is often implied that the tool is “doing history”, with respect to interpreting the “meaning” underlying the data. Furthermore, there is often a lack of real-world examples that demonstrate how a given tool interfaces with the data. When researchers are unable to understand the approach, the tools tend to be perceived as “little more than a black box”, especially in the case of “advanced visualisation” tools, which do not explain how the output was generated, or how the resulting visualisation could be useful for



addressing the research topic [98].

### Defining the way forward

Developing tools for specific document collections to satisfy the needs of a niche group of users, is no longer sustainable given the scale and breadth of digital archives. Furthermore, if researchers are to adopt digital tools more widely in order to support the future development of their respective disciplines, in an age where digital archives are now more accessible than before, then both researchers and tool developers need to understand better what types of tools are really needed [80]. According to a report by the *American Council of Learned Societies Commission on Cyberinfrastructure for the Humanities and Social Sciences* (2006), addressing the current barriers to tool adoption requires that tool developers adhere to a set of criteria that can be used to evaluate the future adoption of digital tools [187]. The criteria are defined as follows.

1. Accessibility. Easy and seamless access to the material by researchers and the public.
2. Sustainability. The appropriate level of investment of both financial and human resources to ensure the long term future of a system.
3. Interoperability. An open, modular, and easily adaptable approach to address different data formats and standards, pre-existing repositories of information, and new technologies.
4. Facilitates collaboration. Researchers are able to share, collaborate, recommend, and comment on the content of the archives.
5. Support for experimentation. The approach is extensible and offers potential for future experimentation, which will encourage more “risk-taking” and “ambitious research programs” [187].

These criteria are designed to guide the development of large humanities cyberinfrastructure, standalone tools, so as to promote the longevity of a new generation of tools and systems through the principles of modularity, experimentation, and extensibility. When tools and systems do not commit to these principles, they become part of an increasing volume of “abandonware” [174],

represented by the tools and system that never received wider-adoption by humanities researchers. Therefore, in order to support current and future needs of researchers in the humanities, digital tools should aim to be flexible to the domain and language to allow them to be sustainable and applicable to a wide range of research questions in the humanities.

## 2.4 INFORMATION SEEKING IN THE HUMANITIES

In the humanities there are a number of core approaches adopted by researchers for interpreting the cultural context recorded by primary source material that may be of significance to the research topic. Researchers wish to discover patterns of significance, often known as “parallel passages” or “parallel sequences” [154], which exist in potentially large-scale digital archives. An example is the King James Bible, where many of the chapters make reference to the same event, but were written from the perspectives of different authors.

Looking across the various disciplines of the Humanities it is clear that the definition of what constitutes “the researcher” can be quite problematic. Early career researchers or those approaching a new topic, will tend to adopt a directed search strategy in order to identify the most highly relevant primary source material in their chosen domain. Established humanities researchers, on the otherhand, tend to focus on acquiring an understanding of the primary sources through close-reading [181,188], where the researcher reads the text several times in order to obtain a comprehensive overview of the relevant textual content. This is then followed by a deeper analysis, which attempts to identify the relationship between patterns in the textual content reproduced over several documents (known as hermeneutics) [187], which can also lead to the discovery of new documents, known as linking [181]. The approach allows researchers to evaluate any correlation or connection between entities and events, which define the context and scope of the research topic [141]. Coverage of the topic is therefore an important aspect of the research, as the results are largely evaluated on the basis of the extent and depth of the identified patterns, or relationships, between the sources [141].

Unsworth (2000) [184], proposed a set of “scholarly primitives” to describe the activities conducted by researchers of textual primary source material. The motivation behind these primitives was to define a set of basic functions commonly performed by researchers that could provide an appropriate framework for tool developers to adopt in order to directly address the needs of researchers. Three of the primitives relevant to the thesis are as follows:

- *Discovery*: The researcher identifies a collection of primary source ma-

terials that describe a specific research topic or purpose.

- *Selecting*: The researcher filters the collection for a subset of the most relevant sources for close-reading, and further analysis.
- *Comparison*: The researcher gains a greater understanding of the importance of each source according to the wider cultural context recorded by the collection, through comparison of the relevant sources.

Digital libraries and digitisation projects have begun to provide tools for searching the content of their digital archives, but the resulting approaches tend to be limited, or ad-hoc with respect to the retrieval model, and the way in which the tools interface with the data is often incompatible with how researchers wish to locate information. When humanities researchers search digital collections, they submit very specific query types, reflecting the names of people, geographical locations, chronological terms, and events [71]. Search engines provided by digital archives are not very flexible to query formulation, and do not provide much support for erroneous query specification, where there is a mismatch for one or more of the researchers query terms. For example, the search tool supporting *Shakespeare's Globe* [36] is limited to keyword search over an index of the metadata record for the documents. The approach models the traditional card index adopted by the library catalogues, which means that search is limited to keyword matching in the the values of the metadata record, for instance the title or subject of the document, which tend to be very short and do not necessarily contain much useful information about the actual topic expressed by the document. The provision of full-text search would enable researchers to locate relevant sources more readily, by providing search over the full content of the documents, but the most common approach adopted for full-text search still tends to be keyword-based search, which adopts a word-level representation for the document and query. The retrieval task involves identifying exact matches for the terms in the documents. This is often incompatible with how researchers define their information need. As mentioned, their queries are highly specific, and often describe named entities [63], reflecting the names of individuals and locations, which tend to produce large lists of imprecise results, unless included as part of a phrase [71]. Consequently, humanities researchers often submit phrase-like queries consisting of two to three words to filter the search results to a specific set of relevant entries. However,

many of the metadata search engines provided by digital archive projects, are not optimised for phrase search, which means that researchers find little of relevance in the top ranks of the search results when specifying these types of query [71]. The issues with the keyword approach are best illustrated by an example:

The *Blake* project [46], provides full-text search over the literary works and paintings of William Blake. The archive adopts a boolean retrieval model [142], combined with a word-level representation for the documents as its model for search, which is an approach often adopted by digital archives. The limitations of the retrieval model become obvious when we consider a user who is not completely familiar with the collection, or who is unable to formulate the exact query for the specific document they wish to retrieve. Consider, for instance, a search for the known-item, Blake’s “Visions of the Daughters of Albion”, using several search strategies, which might represent a user who has difficulty formulating an appropriate query:

- Searching with an approximate query “Vision of the Daughter” as an exact phrase, returns no results.
- Reformulating the query to a Boolean OR query-type retrieves 67 results, but still with no known-item in the list of results.
- Reformulating the query to “Visions of the Daughter” or “Visions of the Daughters”, and relaxing the match to any terms, ranks the known-item as third in the list after “The Book of Thel” and “America a Prophecy”. This is due to the fact that the retrieval model ranks documents according to the order of the terms in the query, and so documents with the word “vision” are ranked higher than documents with the word “daughters”, which is quite an ad-hoc approach.
- Reformulating the query to match the exact terms contained in the title returns the known-item at the top-rank position.

The example above, illustrates some of the difficulty experienced by researchers, who are forced to adapt to the tool in order to obtain the desired results. Aside from the issues inherent in the ranking of results by the boolean retrieval model, if a character-level representation for the documents was adopted, the retrieval model would be able to compensate for these examples of erroneous

query specification on the part of the user. This is achieved by returning partial character matches in response to the query. The disparity between the way researchers search for primary source material, and the ad-hoc or heuristic approaches adopted by the underlying retrieval models adopted by the tools, has resulted in a call for more flexible “search capabilities” [141], than those currently being supported, which tend to be designed to address the information seeking needs of the general public [108]. More advanced approaches to retrieval offered by large-scale search engines like *Google*, *Yahoo*, and *Bing*, allow researchers to locate primary sources across a range of media including text, images, and video. However, they are still not suitable to “do research” [71], as they tend to hide the existence and extent of the sources relevant to researchers, due to “high recall” [141], which causes many of the relevant sources to become obscured by less relevant ones. Furthermore, there is generally no further provision for analysing the actual content of the source. As stated by Rockwell (2003), “original research consists of asking new and unanticipated questions”, and this requires tools to search and access a much larger body of supporting source material. For example, whilst past studies have provided insightful and rich descriptions of the experiences of a handful of holocaust survivors, the ability to perform the same analysis over thousands of eye witness accounts offers a new depth and breadth of analysis that was never before possible [187]. However, when it comes to analysing the source material, the “breadth of tools with which to study the evidence” is not currently available [163].

## 2.5 EXPLORING DIGITAL ARCHIVES

In attempting to understand where the gap is in terms of tool provision, and why certain tools do not lend themselves to wide adoption, this section presents an overview of some of the most popular, or often cited, tools adopted by researchers of the humanities. The tools and systems can be divided into two categories, the first category represents tools and systems developed for research. Many research tools and systems are developed to meet the needs of a specific research project, and later released to the research community to support similar analyses or corpora. The second category reflects tools developed as commercial products to address a gap in the market represented by a group of niche users who have an interest in particular text collections represented by popular, well-known, or historic texts, such as the Bible.

### 2.5.1 Research tools and systems

Systems and tools developed for the purpose of research are often the result of funded research projects that focus on a specific archive that is domain or language-specific, e.g. the 1641 depositions project and the Blake Project, mentioned in the previous section. Some of the resulting research systems and tools have attempted to support the needs of different groups of researchers, however, the approaches are often optimised for the same or similar domain and language, for instance, *Wordseer* and *Voyant Tools*, and the *Responsa* project. These tools are not usually generalisable to applications outside of the project requirements, due to restrictions or limitations in their design. For example, standalone tools such as *DocuBurst* do not support the upload of multiple documents for analysis. Furthermore, certain tools only operate with specific data-formats, requiring some effort on the part of the researcher in order to make the data conform to the required input. Despite some of the limitations inherent in the approaches, as identified in Section 2.3, some tools have experienced wider-adoption, and for a longer period of time, exceeding the extent of the project in which they were developed. One such example is the *The Software Environment for the Advancement of Scholarly Research (SEASR)*, which has been sustained through the release of developer tools that can be used to extend the platform for specific research and digital archives. Several of these systems are summarised below, with emphasis on

those that operate over similar corpora, or provide a similar set of features as those supported by the approach presented in the thesis:

- *Wordseer* [48] is a text analysis web application for English texts that provides search in the form of “grammatical search”, natural language processing tools for extracting words from the document text to create word frequency statistics, and visualisation tools for analysing the relationship between word sequences. The tool requires users to upload their documents in *XML* files, which are then preprocessed, and stored in a *MySQL* database, which acts as the main index for the document terms.
- *Voyant Tools* [44] is a browser-based suite of text-analysis tools that enable researchers to browse and analyse a corpus of documents. The system provides tools for generating concordances through a Key Word In Context (KWIC) tool, word clouds for identifying the most frequent terms in a document and the corpus more generally. In addition, the tool provides data-analytics in the form of statistics constructed from the word frequency distributions, according to specific sections of the document.
- The Bar-Ilan *Responsa* project was established in Bar-Ilan university in 1963 [17]. The *Responsa* system operates with a corpus of Hebrew texts spanning approximately three thousand years, and contains prominent religious and legal texts represented by the Mishnah, Talmud, Torah, and the Bible in Aramaic [79]. The *Responsa* system is still one of the most popular systems for the study of Hebrew and Aramaic historic texts today, and the research that evolved from the project has made a considerable contribution to computational linguistics and information retrieval for Hebrew texts. The system supports word, and phrase search, which is flexible to the variety of variant forms represented by affixes in the language, and the comparison of “parallel passages” between Talmudic and other documents. The system is limited to the Microsoft Windows operating system, making it an example of a platform specific system, which restricts its accessibility for some researchers.
- *CULTURA* [9] was adopted as part of the *1641 depositions project* at



Trinity College Library (Dublin) [19], for the analysis of 31 volumes of books containing 19,010 pages of witness accounts reporting theft, vandalism, murder, and land taking during the conflicts between Catholics and Protestants in 17th century Ireland. It has also been applied to the *Imaginum Patavinae Scientiae Archivum (IPSA)* digital archive of illustrated manuscripts representing depictions of herbs and plants, which dates from around the 14th century. The archive is largely image-based with some descriptive text composed in Latin. *IBM LanguageWare* [1] provides text analysis tools for mining facts from large repositories of unstructured text. The main features include lexical analysis, language identification, spelling correction, part-of-speech disambiguation, syntactic parsing, semantic analysis, and entity and relationship extraction. *LanguageWare* was selected partly due to the complexity of the language contained in the documents, which have many spelling mistakes making analysis a complex task. *CULTivating Understanding Through Research and Adaptivity (CULTURA)* is a project related to the 1641 depositions, launched in 2011, and supports researchers through tools for normalising texts containing inconsistent spelling, entity and relationship extraction from unstructured text, and social network analysis tools for displaying the entities and relationships stored in the documents and metadata.

- *Texcavator* is a research tool developed as part of a project under the *Translantis* research program at Utrecht University (Netherlands) [18], which was invited to be part of a pilot project, coordinated by the British Library, for the Financial Times newspaper archive. The tool provides search over a newspaper archive, with partial query matching supported through wild-card characters. The search tool is constructed from the Elasticsearch framework, which supports query recommendation, and search result snippet generation [11]. *Texcavator* provides data mining and visual summaries of the document collection based on user generated time-lines, word-clouds, and further visualisations generated from user supplied tags, such as named entities, using the included annotation tools.
- *The Blake project* [46] provides search and comparison tools for exploring the works of William Blake. The retrieval model is based on a boolean

retrieval model that adopts a word-level representation for the documents. The archive contains many literary works and paintings created by the author and artist, which can be compared side-by-side using a text and image comparison tool.

- *Text Analysis Portal for Research (TAPoR)* [38, 163] represents a portal for humanities researchers to try out different text analysis tools for their research, aggregated under a single site. In some respects it was designed to meet the need for a “cyberinfrastructure”, that was designed to support common research needs, through a network of universities hosting servers and electronic labs where text analysis tools could be made available.
- *The Software Environment for the Advancement of Scholarly Research (SEASR)* [37], provides a range of text analysis tools as part of a virtual environment. Programmers develop tools using Java combined with *RDF (Resource Description Framework)* [5] to generate a series of reusable software components that can be coupled together and executed as part of a work flow of individual processes for data-analytics, text mining, and resource sharing. The system provides more flexibility than many of the systems and tools presented in the section. This is achieved through the “workflows” defined by the developer, for example, a preprocessing step, could be included before a call to generate word-frequency statistics, which means that developers can create tools for specific research groups and archives.

### 2.5.2 Commercial tools and systems

Commercial software is largely developed for a specific market or “niche” community of users. The majority of commercial systems incur a cost to download, or subscribe to the service, which is often at a level affordable only by institutions, causing them to be out of reach to many researchers [187]. Commercial software is also often targeted to specific platforms, limiting their availability. Furthermore, the tools may be “dumbed-down” for use by a general user, which often makes them incompatible with how researchers wish to analyse and compare the content of digital archives [187]. Some of the issues are attributed to how the tools are marketed to potential users. For example, it is

unclear what “Everything Search”, and “Smart Search” provided by the *Logos* Bible reading software, actually gives users in terms of functionality, as there are rarely any details on how the system interfaces with the data.

- *Logos Bible Software* [15], released in 1992, and developed by *Faithlife Corporation*, was designed for the study of the King James Bible, New Living Translation (*NLT*), and the Revised Standard Version (*RSV*) in English, Greek, and Hebrew original texts. It was founded by two former Microsoft employees at *Logos Research Systems*, a digital publisher and software company. The last release of the software was version six, released in 2014. The software provides tools for linking to external sources of information, creating annotations, and analysing the documents through “parallel passages” represented by interlinear translations of the text in English, Greek and Hebrew. The software also provides search in two forms “Everything Search”, and “Smart Search”, but there is no further details provided about the particulars of each of the two forms of search. *Logos* ships with data sets of named entities, important events, a range of dictionaries, lexicons, and encyclopedias. Visual media is also included with the documents in the form of illustrations, time-lines generated from important events, and maps of locations mentioned in the texts.
- *Accordance Bible study software* [8], released in 1994, and developed by *OakTree Software, Inc.*, was an early example of a more sophisticated approach to tool development for the analysis of the English, Greek, and Hebrew translations of the Bible. The system features a search tool providing exact and partial query matching, a browsing tool, and visualisation tools for creating a time-line representing when people lived and died, important events, and an atlas view for exploring famous journeys and battles mentioned in the texts.
- *Bibleworks* [20], is a desktop application, represented by a suite of software tools that provide search and text mining of the Bible. The software is aimed at both the individual who wishes to study the Bible, and researchers. The software provides a search tool with exact and partial matches to the query. The search tool is also supported by a morphological filter that allows users to filter the results according to a

part-of-speech, or the tense and aspect of the terms in the query. These tools are, by their necessity, language-dependent, since they deal with language-specific features, which operate at the morpheme-level and requiring a language-specific and word-level approach for identifying the morphemes encoding tense and aspect in the language.

- Text Metadata Services (TMS) [40], is a suite of NLP tools developed by ClearForest – a company later acquired by Thomas Reuters in 2007. The TMS platform is now deployed across the complete collection of Thomson Reuters metadata associated with their online digital content. The platform provides tools for detecting events in narrative text, extracting and identifying relationships between named entities, and topic classification. The output of the tools are represented by derived datasets for supplementing the textual content, which are supplied at a cost to customers in the commercial sector.
- Leipzig Corpus Miner [?] is an analysis tool developed for the retrieval, annotation, and mining of textual data through machine learning. The system allows users to combine the tools in a module way to produce various forms of analysis from quantitative corpus linguistics to qualitative approaches, for instance hermeneutics involving the interpretation of written texts. The tools generate frequency and co-occurrence statistics from the document collection, topic models for classification, and supervised learning methods based on annotated text to automatically annotate sections of the documents.

A summary of the tools introduced in this section, is presented in Table 2.2, which references the corpora, languages, and domains that are supported by the tools.

Digital tool	Corpus	Domain	Language(s)
Wordseer	Any	Any	English, Arabic, Chinese, German, Italian, Bulgarian, Portuguese.
Voyant Tools	Any	Any	English
Responsa	Bible, Mishnah, and Talmud	Religion, Law	Hebrew, Aramaic
CULTURA	1641 depositions project, Imaginum Patavinae Scientiae Archivum ( <i>IPSA</i> )	History	English, Latin
IBM LanguageWare	Any	Any	Multilingual
Texcavator	Any	Any	Multilingual
The Blake project	Works of William Blake	Prose	English
Text Analysis Portal for Research	Any	Any	Multilingual
SEASR	Any	Any	Multilingual
Logos Bible Software	The Bible	Religion	English
Accordance	The Bible	Religion	English
Bibleworks	The Bible	Religion	English
Text Metadata Services (TMS)	Thomas Reuters	News	English
Leipzig Corpus Miner	Any	Any	English

Table 2.2: A summary of the research and commercial digital tools discussed in this section.

## 2.6 DISCUSSION

The chapter described the current situation faced by the humanities researchers, who wish to use digital archives as a central resources of primary source material important to their research. There has been a large effort on the part of digitisation projects to make digital representations of archival content accessible for research through online access. This is an important first step in providing unprecedented access to primary source material, which was previously only accessible by visiting the physical archive or library. However, many of these digital archives provide little if any tools to actually interrogate the sources, which makes them largely redundant to researchers, since without the necessary tools for search and browsing, it challenging and time-consuming for the researcher to analyse thousands of potentially relevant documents, compared to the smaller collections of the past.

Where tools are provided, they may require adaption by the researcher, in order to obtain the required output, which is usually the result of the tool being incompatible with the analysis or format of the data. Researchers may also find themselves grappling with a number of disparate systems developed as individual tools that address specific research projects, and provide little consistency in the analysis across different digital archives. The commonly adopted word-level representation for the terms of the documents and query, often restricts the developed tools to specific domains and languages. Without generalised systems and tools in place, researchers are currently unable to fully analyse all the documents in a consistent way that would permit the discovery of “parallels passages”, representing textual patterns that encode important linguistic, literary, social, and cultural information to research. The main observation from the literature is that tools developed for the digital humanities need to increase their scope in terms of the provided functionality, data that the tool can operate with, and the intended audience [98]. Furthermore, the solution should be modular, extensible, and provide support for the types of analysis that researchers actually wish to perform, which will in turn enable them to ask “new questions”, and revisit “old ones” [185].

The approach outlined in this section supports the development of a general purpose digital infrastructure, and supporting architecture (see Chapter 3), which provides cross-domain and language-independent support for search and

text mining of digital archives, both large and small. The infrastructure and accompanying architecture, presented in the subsequent chapters of the thesis, represents a new approach that directly addresses the need for quantitative tools that can support the traditional qualitative methods adopted by researchers in the humanities. The approach allows any digital archive to be made accessible very quickly with a consistent set of generic tools. Digital tools developed from the infrastructure inherit the following properties from the data model and architecture:

- *Flexible to the digital archive.*

Addresses the need for a “cyberinfrastructure” that can provide search and text mining tools straight-forwardly for other research groups and digital archives.

- *Language and domain-independent.*

Any arbitrary set of symbols can form the basis of the analysis [165], making the proposed infrastructure flexible to the domain and language. The character-based *SLM* supports the indexing and search of an archive with very little preprocessing of the text. Furthermore, the character-level *SLM* captures more information at the sub-word level, which is generally ignored by word-based approaches, enabling the infrastructure to model some level of semantics encoded in the documents.

- *Unified approach*

The tools are developed from a unified approach to the storage, search, and text mining (see Chapter 5) of both structured data represented by metadata, unstructured data represented by the document text, and supplementary data reflected by images of the original source material.

- *Tolerant to the language of the documents and the query.*

The character-level representation for the documents and terms of the query provides built-in compensation for erroneous query specification on the part of the user, or data-integrity issues resulting from the poor quality of the digital representation. Phrase search is not often supported by digital archives, or many of the tools identified in Section 2.5. However, facilitating phrase-search enables researchers to narrow the search results to a specific set of relevant sources.

- *Transparent.*

The importance or relevance of character-sequences recorded in the documents and metadata is modelled using a probabilistic approach, where the assumption is that the more probable a document is given a sequence of characters, the more likely it will be relevant to the researchers information need. This means that researchers are no longer faced by a “black-box”. The *SLM* underlying the infrastructure enables researchers to evaluate the tool output under the context of a probabilistic framework, which is intuitive and easy to understand.

- *Accessible.*

The tools are provided online, and the interface has been developed according to common design practices adopted by popular tools, in order to promote accessibility and centralise the content of the archives, as presented in Chapter 6.

The details of the architecture supporting the development of the *Samtla* system, constructed from the infrastructure, are discussed in more detail in Chapter 3. In Chapter 4, the data model based on *SLMs*, is presented. The *SLM* assigns a weight to the terms of the documents to produce a language model of each document, and one for the whole collection, which acts as the main index for search. The chapter discusses how the resulting *SLM* is stored as a space-optimised  $k$ -truncated suffix tree data structure, which is then queried to retrieve a set of documents that are sorted according to the probability of matching the query inferred from the language model of each document to produce a probabilistic ranking of the documents that most likely meet the information need of the researcher. The resulting data model supports tolerant search by retrieving both full and partial matches to the users query as a result of the character-level representation selected for representing the terms of the documents.

The application of *SLMs* to tasks other than information retrieval is introduced in Chapter 5, which describes several text mining tools developed from the infrastructure and supporting architecture, in response to the need for digital tools to support specific types of analysis from the research groups discussed in the case studies. These include digital tools for search result filtering, browsing tools generated from the metadata and named entities, image view-



ing and browsing, query and document recommendation, and the comparison of variable length character-sequences shared between groups of documents discussing the same or similar content.

The user interface is an important aspect of system design and development, and *Samtla*'s user interface has been the result of consultation and feedback received from our research groups, as part of an iterative design process. The motivation for the current implementation are presented in Chapter 6, which describes the minimal, flexible, and context-dependent approach adopted for presenting the tools to the researchers in order to dedicate more space to the document content and digital tool output.

The underlying data model has been formally evaluated by a group of general users enlisted through a crowdsourcing platform, the details of which are discussed in Chapter 7. The performance of the *Samtla* search tool is assessed by comparing the ranking of the documents inferred from the *SLM* for a set of predefined queries, against a ranking generated from the relevance judgements submitted by the users. The graded relevance judgements provided by the users were evaluated using a novel approach comprising a set of non-parametric measures combined with the bootstrap method as a measure of statistical significance.

The thesis concludes with a summary of the main contributions and limitations of the research. A discussion of the potential avenues for future development of the infrastructure and *Samtla* system is also presented, which describes additional digital tools that are not currently deployed in the current version of *Samtla*, as well as extensions to the data model supporting the infrastructure.

The proposed infrastructure, and supporting architecture was developed in collaboration with real users at the beginning of the development process, for which no appropriate tools existed, or were inflexible to the type of analysis performed by the researchers. To summarise, the motivation for the infrastructure proposed in the thesis is to provide humanities researchers with tools that will enable them to pose new questions, through a comprehensive search, and comparison of key patterns significant to discovering documents that are relevant to their research topic.

## CHAPTER 3

# ARCHITECTURE

This chapter describes one of the main contributions of the thesis – a generalised infrastructure that supports domain and language-independent search, browsing, and comparison of documents stored in small and large-scale digital archives. A system developed from the proposed infrastructure and supporting architecture is introduced in Section 3.1. Section 3.2, provides an overview of the main components of the architecture supporting the *Samtla*'s search and mining tools. The *Samtla* system has been empirically assessed through collaboration with a number of research groups in the digital humanities, who provided the digital archives, see Section 3.3, as well as continued feedback on the design, and accessibility of the tools during the development process. The chapter concludes, in Section 3.4, with a discussion of the main advantages of the proposed infrastructure over current approaches to tool provision for the humanities.

### 3.1 OVERVIEW

The *Samtla* system is a web application built on a client-server architecture, the client is run in the users web browser, and the server is normally hosted externally, enabling many distributed users to interact with the application at the same time. A web application is represented by a web page with some client-side code written in Javascript to translate the users interaction in to requests for information from the server. Web pages are usually represents by a series of static *HTML* documents, for example, a home page, about page, contact us page. Links marked up in the text of the document provide the main

means for interacting and navigating the website. However, each interaction with the links in the web page triggers a round trip of communication between the web browser client and the external server. When the data for the new page is received by the client there is a noticeable refresh as the browser clears the old content and loads the new page data. A web application functions in more or less the same way, except that only select parts of the page are updated, which provides a seamless user experience similar to that of a desktop application.

Web applications have become popular as they can be deployed without the need to directly access the users device for the installation or updating of the application. Web applications therefore provide inherent cross-platform support, as a result of the ubiquity of web browsers across a multitude of devices. Some examples include, Google's webmail client *Gmail* [28], and *Google Photos* [29], an image editing application. These applications provide the same consistent functionality as their desktop software equivalents, but the application can be accessed by the user across different devices independently.

Web applications are more complex than their static counterparts, and require an approach that facilitates the handling of communication between potentially thousands of clients and the data stored on the server. Furthermore, web applications are software applications in their own right and can therefore be much larger in scale than a website. Developing highly maintainable and reusable code can be achieved by adopting an appropriate design pattern suited to the task of separating the application in to more manageable components, and for decoupling the components according to different levels of responsibility [96]. Some important design considerations include how the interface of the web application should be structured to facilitate accessibility (see Chapter 6), how users expect to interact with the application, and how the input provided by the user affects the data stored on the server. In the next section, a discussion of the web application implementation is presented, where the framework supporting the *Samtla* system is introduced.

## 3.2 THE FRAMEWORK

The *Samtla* system operates with a single code-base making the whole system data-driven. An advantage of this is that upgrades or changes to the function-

ality of *Samtla* can be rolled-out simultaneously to each user group. The data for each *Samtla* comes from corpora, where each corpus or subcorpus consists of a collection of text documents, which may be grouped according to a specific topic, genre, demographic, or origin (e.g institution storing original versions of the digital texts). A common design pattern adopted by web application developers is the *a Model-View-Controller (MVC)* design pattern [73, 130]. The *MVC* pattern is composed of three main components, which are listed as follows:

1. *Model*: located on the server and responsible for retrieving and updating aspects of the data model used by the system.
2. *View*: the client, which in our case is the user interface loaded in to the user's web browser.
3. *Controller*: a communication layer between the client and the server, where the user input is translated in to events that trigger an action in the *model*, the controller then updates the *view* component with the appropriate data.

An overview of the *Samtla* architecture is illustrated in Figure 3.1, where the components of the system are grouped according to the separation of concerns described by the *MVC* design pattern. Arrows in the diagram represent the flow of communication between the various components of the system. Each of the components of the *MVC* are described in more detail in the subsequent subsections to follow.

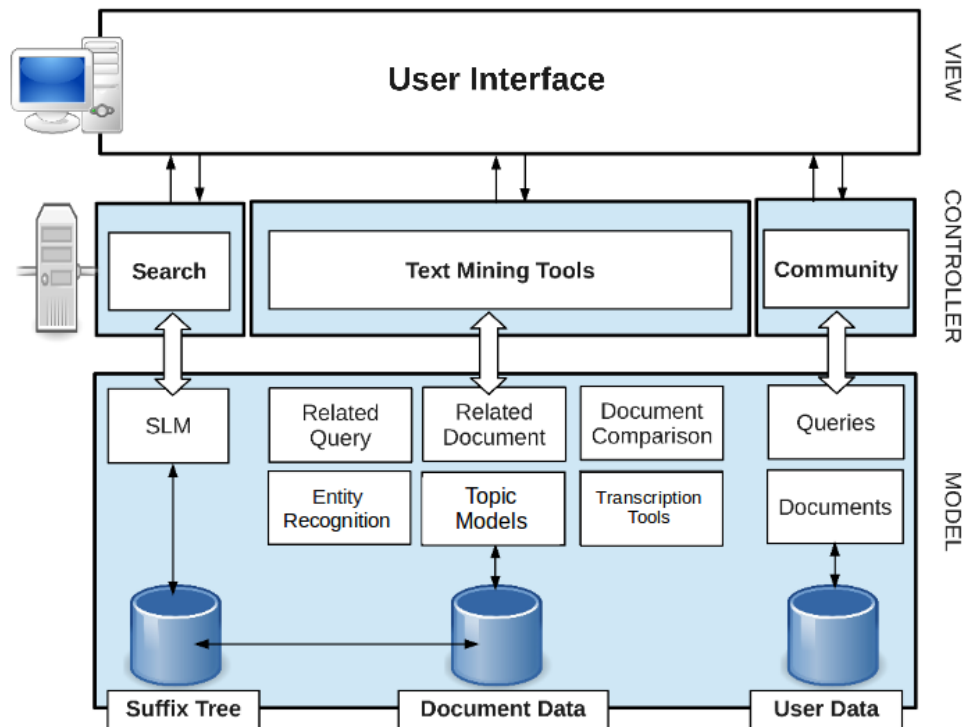


Figure 3.1: The *Samtla* architecture.

### 3.2.1 The Model

The *model* component encapsulates the application logic, which interacts with the data stored on the server. Any changes made to the data in the *model*, are communicated to the *controller* (see Section 3.2.3), which triggers an appropriate event in the *view* to update the user interface with the new data. Furthermore, the *model* may receive periodic requests from the *view* component for state updates, whereupon it sends the most recent snapshot of the data.

A typical example in *Samtla*, is the search and browsing histories for each individual user and the community. When a user submits a query or views a document, the *view* communicates this change to the *controller*, which then triggers an update to the log data for that user in the *model*. Next, the *model* communicates with the *controller*, which passes a ranked list of the most recent user activity to the *view* component of all the connected clients, and the section of the page representing the output of the recommendation tool is reloaded with the new content. All data passed between the *controller* and the *model* is serialised in *JavaScript Object Notation (JSON)* [10] format, which enables the data objects to be further processed by the browser for dynamic rendering

of *HTML* snippets for the search results.

In *Samtla*, the *model* is composed of a library of Python functions that interact with the system data. The system data is divided according to functionality and so the data represented by the documents, metadata, and images are stored in different databases. In addition, the search tools use an index constructed from the  $n$ -grams of the documents and metadata records, written to *JSON* format, and serialised to disk; see Chapter 4 for more detail on the suffix tree component. The *model* component adopts the following technologies:

- Python programming language: some adoption of the *Numpy* and *Scipy* statistical libraries for computing the prior in Chapter 5, and the correlation measures in Chapter 7.
- SQL database: all data is stored in a SQL database according to function e.g. the *SLM* document model, metadata, pair-wise JSD scores for the related documents, and user activity log data.
- The character-level  $n$ -gram *SLM* for the collection model is stored in *JSON* format to promote portability. The loading function used by the standard *JSON* library is overridden with a custom loader, which casts strings representing numbers to integer format, which considerably reduces the memory requirements of the data structure when loaded in to memory after construction.

Where possible, native tools and standard-libraries have been preferred due to issues arising from third-party API updates, which can sometimes break a system due to the depreciation of certain features. The advantage is that the system requires little maintenance, aside from development revolving around the system itself e.g. new features and text mining tool development.

The application logic is represented by a library of tools. These include the *search* component is responsible for answering user queries, and uses a *Statistical Language Model (SLM)* [194]; see Section 4.4 for more detail on how we make use of *SLM* in *Samtla*. The *SLM* communicates with one or more suffix tree data structures (see Section 4.3), representing the index of the documents and metadata for search. The suffix tree is loaded into memory at runtime to ensure a fast response to user queries. The suffix trees also

support a number of mining tools, which include a *related query* feature for recommending queries to the user based on permutations of the original query, a *related document* tool, which presents users with a list of similar documents to the one they are viewing, and a *document comparison* tool that facilitates the comparison of shared-sequences between documents. As mentioned, the *community* component is responsible for logging user data, such as query submissions and document views, usage statistics reflecting the user's navigation histories through the system, and for returning recommended queries and documents based on their popularity in the user community; see Chapter 5 for more detail. The only archive-dependent component of the *Samtla* system, resides in the *model* and is represented by a small wrapper function responsible for parsing the raw content of the documents, or metadata, according to a number of common electronic formats e.g. *ASCII*, *HTML*, *XML*, *TEI*, *PDF*.

### 3.2.2 The View

The *view* component represents the client, or user interface (*UI*) (see Chapter 6), which displays the results of the user's interaction with the web application through the browser. The *view* is a component of the digital infrastructure with respect to the front-end application logic, whereas the design of the user interface is considered to be a customisable component that can be tailored to the needs of each user group.

The *view* is composed of a series of *HTML* document fragments, stylesheets for formatting the look of the page elements, and Javascript code libraries for handling the users interaction with the user interface, and adopts the following technologies:

- Javascript programming language: used for developing the functionality of the tools in the user interface.
- jQuery: for addressing the disparity between web browsers to ensure the interactive elements of the interface are consistent cross-browser.
- HTML5: The development of the HTML5 standard provided support for a number of new features including Local Web storage for recording users preferences. The Canvas element of the page is used for rendering and interacting with the original image of the documents.

The aim of a web application is to provide a desktop-like experience through the web browser. When users interact with a traditional web page composed of static *HTML* documents, each call to the server requires a certain amount of processing time before a response is received. During this time the user is unable to interact with the interface as a result of the main thread of the application being blocked while the server is being polled for a response for data.

*Asynchronous Javascript and XML (AJAX)* [105] is a group of web-technologies and approaches that enable parts of a web page to be updated dynamically without the need to block the main thread, or reload the entire content of the web page after each interaction. The main component of the technology is the *XMLHttpRequest* protocol [7] that provides the main method of communication between a client and server. This makes a web application appear as responsive as a typical desktop application, by enabling the user to continue to interact with the *view*, while the response from their last interaction is being processed in the background.

When the user interacts with an element on the page, the *view* assesses whether more data is required. If a user collapses or hides an element in the interface, there may be not need to revisit the *model*, whereas a list of search results in response to a query does require additional data to be retrieved. If needed, an *AJAX* request is sent to the *controller* for a response from the *model*, which is then passed back to the *view* to update the appropriate section of the page. *AJAX* is one of the key web technologies that permitted the emergence of web applications capable of emulating the user experience of desktop applications.

### 3.2.3 The Controller

The *controller* defines how the web application behaves in response to the users interaction with the system. Consequently, the *controller* acts as the main communication bridge between the *model* stored on the server, and the client represented by the *view*. The *controller* achieves this through the adoption of the following technologies:

- Python 2.7 programming language: version 2.7 was selected due to its compatability with some of the more advanced Python packages includ-



ing *numpy* and *scipy*, which provide powerful matrix computation and implementation of common statistical measures. Python 3.2 may be more appropriate in future due to the unifying approach adopted for the encoding of strings e.g. unicode versus ASCII.

- Django web framework: provides the main mechanism for communication between the server and the front-end of the *Samtla* system. Also responsible for the storage, retrieval, and validation of user login credentials.

The *controller* maps a number of event listeners to the elements of the web page in the *view* layer using a unique *URL*, one for each action, which are then mapped to a corresponding function in the *model* which is mediated by the Django web framework.

The event listeners in the *controller* are partitioned according to functionality: search, mining, and the recommendation tools (see the *controller* layer in Figure 3.1). When the user interacts with the page elements, through selecting text or clicking a button widget, an event is triggered and picked up by the *controller*, which calls the appropriate function that instigates a request for information or an update to the data in the *model*. When data is received by the *controller* the appropriate section of the page is identified and the data is loaded in to the *view* for rendering on the page.

To illustrate with an example, clicking a button representing the query and document history for the user and community of users (see Chapter 6), triggers an event listener in the *view* component, which sends a request for an update on the most recent search and browsing activity in the system, from the *model* through the *controller* component, which passes the result of the request to the *view* for rendering in a side-bar.

### 3.3 THE CASE STUDIES

*Samtla* has been developed through continued collaboration with a range of user groups. Each case study provided a specific challenge to information retrieval, browsing, document comparison, and user interface design. One of the biggest challenges revolves around the nature of the corpora, which encompass historic documents that vary in terms of the language, due to

a lack of standardisation in spelling conventions between periods, and individual authors who tended to write as they spoke given the local dialect. Furthermore, historic texts contain archaic affixes, vocabulary, and grammar that have no modern day equivalent. Examples of such corpora include the *Aramaic Magic Texts from Late Antiquity*, and the *British Library Microsoft archive* of scanned books.

The quality of the digitised object may also present a challenge, due to poor *OCR* recognition rates, or damage to the original source. Quality issues have an impact on all the document collections introduced, to some degree, but the problem is particularly pronounced with the aforementioned *British Library Microsoft archive*, and early editions of the *Financial Times newspaper archive*, which was acquired through a pilot study in collaboration with the British Library.

Aside from common issues associated with language change over time and digitisation quality, some collections may contain documents and metadata in different languages within the collection itself. For example, the previously mentioned *Aramaic Magic Texts from Late Antiquity*, and *British Library Microsoft archive*. A further case-study is provided by the works of Giorgio Vasari, who was one of the world's first art historians. *Samtla* operates with his most famous and significant work, the *Lives of the Most Excellent Painters, Sculptors, and Architects*. This collection supports the research needs of the Art and History department at Birkbeck University by facilitating the exploration, search, and comparison of the English and Italian translations of the original, under a single system. A last case-study is represented by the *King James Bible* in English, which was developed for the purpose of demonstrating the capabilities of the framework, as many people are familiar with the content of the Bible, and the output of the search and mining tools are in English.

The philosophy has been to work alongside our users to understand the problem domain and then to develop the tools and features that will be of practical use to them. One of the advantages of the framework presented in the thesis, is that the tools are modular in design, and data-driven. This means that tools developed for one specific user group, can be released to all user groups, allowing the whole community of users to benefit from tools developed through the collaborative efforts of each separate research group.

This section describes each case study in more detail with a description of the users group, the problem domain, the provenance of the texts, and the tools that were designed to address the research needs of the group.

### Aramaic Magic Texts from Late Antiquity



Figure 3.2: An Aramaic Magic Bowl for protection against demons, 6th century AD, ©The Trustees of the British Museum [21].

The first research group is represented by a team of historians led by the University of Southampton [6], who are analysing a corpus of 650 Aramaic Magic Bowls and Amulets from Late Antiquity (6th to 8th CE). A large portion of the texts are written in ink on earthenware bowls (see Figure 3.2), and represent important primary sources describing the cultural and religious beliefs of Jewish, Christian, Mandaean, Manichaean, Zoroastrian, and Pagan communities living in the period just before the Islamic conquest of the Sasanian Empire [43]. The texts are written in a number of related dialects including Aramaic, Mandaic, and Syriac scripts, and some sections in Hebrew representing passages from the Bible [131]. Furthermore, the texts represent a variety of topics and subject matter, including magic incantation formulae, medicine,

law, and culture.

The research involves searching and comparing textual fragments that have significance for history, religious beliefs, and linguistic research. Specifically, there is the potential for discovering unattested vocabulary and grammatical structures that have not been recorded in other similar text collections. The approach involves identifying the main textual fragment, and then locating duplicated forms across the corpus, including approximate matches that represent slight variations of the sequence as a result of differences in authorship, dialect, script, and time period.

The analysis of the texts was largely performed through close-reading of the texts, and with some adoption of computer-assisted analysis through standard wordprocessing and spreadsheet software for search and highlighting text fragments for document comparison. However, the tools were not well-suited, or sufficient for text-analysis over such a varied corpus of documents. The main issue was that the native search tools were unable to identify approximate text fragments, resulting in the recall and precision of the search results being very low. Furthermore, the limited functionality of the tools meant that comparing the complex similarities and variations between the content of the documents was not a trivial task, even for a corpus of only 650 texts.

To illustrate the problem, if we were to segment the text into words according to white-space, it would ignore the fact that agglutinative languages adopt a root morpheme, which is then inflected through the addition of prefixes and suffixes describing number, gender, and syntax (i.e. prepositions) in relation to the subject of the verb. As previously stated, Aramaic words are composed of a tri-consonantal root. As an example, the verb *ktb* “to write”, takes the following suffixes (singular forms):

- *k'tab-it* “I wrote”.
- *k'tab-t(a)* “You wrote” (masculine).
- *k'tab-t* “You wrote” (feminine).
- *k'tab* “He (or it) wrote”.
- *k'tab-at* “She (or it) wrote”.

Nouns display similar behaviour in terms of number, with the addition of singular and plural marking, for instance: *yoma* “day” (singular), *yomayyaa*

“days” (plural). However, there are also irregular forms, resulting in slight phonological changes: *qarta* “city” (singular), *qirwayyaa* “cities” (plural). The noun, may also take a pronominal suffix, such as *yati* ‘me’, or *yat’hen* ‘them (feminine)’, which can be combined with a preposition, for example, *l* “to”, becomes: *li* “to me”, *l’hon* “to them” (feminine).

The above outline of the Aramaic language neglects one of the main features of the texts. The inscriptions were not designed to be read, and consequently there was no editing or “proof-reading” of the final inscription. Therefore, these texts represent how people spoke in late antiquity and include a wealth of information about the phonological properties of different Aramaic dialects. As a result, there is a great deal of variability, even those containing the same inscription or magic formula, which is caused by a number of factors including differences in vernacular, and the literary competence of the scribe. In fact, the bowls were commissioned for clients who were often illiterate, resulting in some bowls being “faked” due to the actions of some ‘unscrupulous’ scribes [167]. The *Samtla* system was first developed in response to the need for tools to cope with the diversity of the texts. The collaboration, feedback, and empirical assessments determined much of the design and implementation of the system’s key tools.

## British Library Microsoft archive

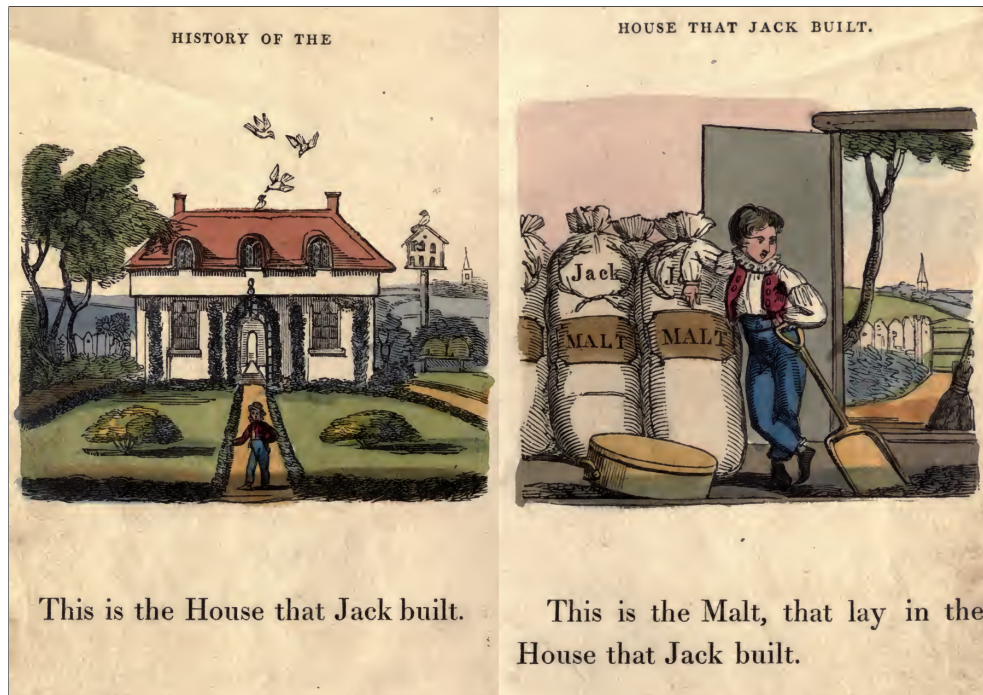


Figure 3.3: “The House that Jack Built”, Moon, M. Harris. Published 1821, London. Printed for Harris and Son, corner of St. Paul’s Church-yard.

Microsoft partnered with the British Library to digitise and make available via *MSN Book Search* [2,3], 25 million pages from the British Library collections. In total 68,000 volumes were scanned, resulting in 30TB of digitised material. The MSN Book Search service, was later abandoned potentially due to direct competition from its competitors like Google, who were providing similar services. The digital archive was bequeathed to the British Library and released to the public domain.

The collection contains out of print books that are little known or studied as a result of being first editions that were never reprinted. The texts are considered to be important to historians, linguists, and educators, but also provides data for the research of information retrieval applied to book corpora [191].

The documents cover a broad range of languages, including English, French, German, Spanish, Hungarian, Russian, and Malagase. Furthermore, the documents are diverse in terms of literary genre, e.g. poetry, maps, journals, reports. In addition, the retrieval task is not trivial as the language contained in the documents evolved over a time period of nearly four centuries (1500

- 1900). Lastly, the OCR quality of many of the documents is quite poor, rendering the documents unsearchable as a result.

Samtla operates with a 3TB sample of the collection, provided by the British Library as a test case. The underlying framework introduced in Chapter 4, solves many of the issues associated with information retrieval over low quality OCR as a result of the flexible search and mining tools. A metadata search tool was also developed for this archive to provide search over different media-types, including unstructured text, tables of figures, photographs, and maps, thus providing a unified approach to information retrieval over different media (detailed in Chapter 5).

Financial Times newspaper archive

THE FINANCIAL TIMES, Friday, 1st September, 1939

## BUILDING AND EQUIPPING SHELTERS

### MECHANICAL EQUIPMENT



**NORRIS A.R.P.**  
F. A. NORRIS & CO. LTD., NORRIS WORKS CO., LTD.  
RIPLEY HOUSE, THEOBALDS RD., LONDON, W.C.1.

### 232<sup>ND</sup> VIEW OF BUILDING

11<sup>TH</sup> A.R.P. FEATURE



**HOMERIC DRAWING OF AIR-RAID SHELTER IN WEST END BLOCK OF FLATS**  
Architect: Howard Leicester and Partners

## Johnson and Phillips Ltd

Charlton London S.E.7

*Good Cables Since 1875*

Foundation Members of the C.M.A.

### ACUTE STEEL SHORTAGE OVERCOME

"I would like to thank you for the prompt and careful manner in which the job was handled during a period of acute difficulty in maintaining supplies of constructional steelwork."

The original of this article may be inspected at our office.

**EDWARD WOOD & CO LTD**  
CONSTRUCTIONAL ENGINEERS

100, Tottenham Court Road, London, W.1.

### EVER READY PORTABLE LAMPS OF ALL TYPES



Because of their independence of interruptions in mains services, Ever Ready Torches, Pocket Lamps, and Handlamps are an essential part of Air Raid Precautions. There is a wide choice of models from 1/- upwards.

Made by the Ever Ready Co. (UK) Limited, Hove, Sussex, England. Sole agents in Great Britain and Ireland of all types, Radio Batteries and Rechargers, etc.

### A.R.P. CONTROL SYSTEMS

are installed at PORCHESTER GATE

THE ARDENTE TALKING SYREN SYSTEM gives the following facilities—

- Air Raid Warning and All-Clear Signal during War.
- Accident, Fire Alarm and Staff Location in Peacetime.
- Radio and Gramophone Music in Shelters.
- Shelter profile, instantly detectable in last alarm.

Suitable for PLATS, OFFICE BUILDINGS, FACTORIES, DOCKS, WAREHOUSES, QUARRIES, etc.

**ARDENTE ACOUSTIC LABORATORIES LTD.,**  
11-12, POLLEN STREET, LONDON, W.1.

### FIRST-AID ROOM IN A WEST END AIR-RAID SHELTER



Architect: Howard Leicester and Partners

### THE HANDY LIGHT for all emergencies

Handmade by the Ever Ready Co. (UK) Limited, Hove, Sussex, England. Sole agents in Great Britain and Ireland of all types, Radio Batteries and Rechargers, etc.

### STEEL DOORS FOR AIR RAID SHELTERS.

Standard designs or made to order as supplied to shelter reviewed on this page.

**GARTON & THORNE Ltd.**  
44, ST. PAUL'S CRESCENT, LONDON, N.W.1

### ARMED WARREN'S POST-SHOWER INTERCOMMUNICATION AND BROADCASTING EQUIPMENT



Figure 3.4: A single page from the Financial Times newspaper archive for Friday 1st September 1939.

The *Financial Times Historical Archive* [27] covers 122 years of the daily editions of the Financial Times newspaper from the period 1888 to 2010. The newspapers cover a broad range of literary domains including: business and finance, British and international politics, science and technology, and arts and leisure. The archive was released as an interdisciplinary resource for the research of history, business, management, finance, and politics over the past 122 years. The complete archive was published online by *Gale Cengage Learning* a publisher of e-research and education for libraries, universities, and businesses. Access to the archive is restricted to institutions and businesses, which puts it beyond the individual researcher, unless their institution has purchased a copy.

The main tools provided by the website include both full-text search, and an advanced search that allows users to specify boolean operators to narrow down the search results. Furthermore, the users can browse the collection for any particular date, or by selecting filters representing the metadata stored for the documents. A pilot study was conducted in 2015, between the British Library and the Financial Times, who were looking for new ways to search and browse their collection. *Samtla* was selected as a case study, together with the *Texcavator* tool (see Chapter 2). The *FT* were looking for novel ways to search and explore their content to help increase their readership to one million subscribers. The British Library released a small sample of the collection containing four years of daily newspapers. The collection is supplied in *XML* format with accompanying metadata embedded in each document, and a library of images representing the original scans of the newspapers.

A set of new features were developed which utilised the image data that comes as part of a document collection. Images include illustrations in the main text of the document, or the scanned original of the collection represents the output of an *OCR* project. Users can choose to navigate between the raw *OCR* text and the original source document, which enables them to view the original document when the *OCR* recognition rate is low.

The *Samtla* system provides the majority of the same functionality as that developed by *Gale Cengage Learning* for the archive. The difference between *Samtla* and the existing system, is that *Samtla* provides tolerant search at the character-level, whereas the *Gale Cengage Learning* system operates at the



word-level. Furthermore, the system lacks a recommender system, and the related document and document comparison tools native to *Samtla* for the *FT newspaper archive*.

### King James Bible in English

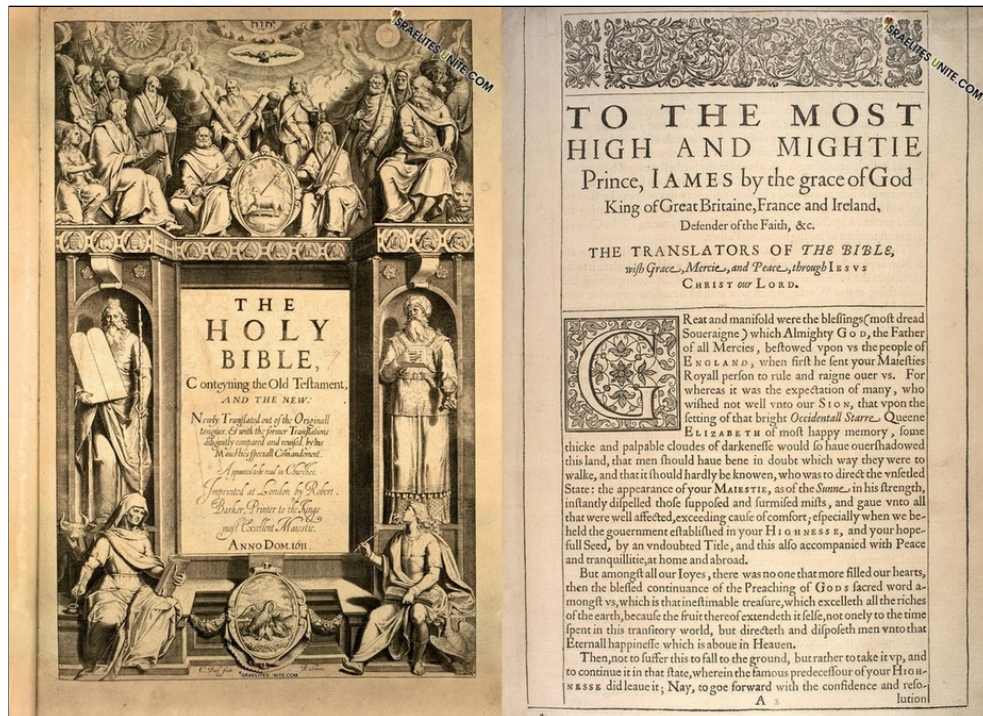


Figure 3.5: The inside front cover of the 1611 edition of the King James Bible.

As a proof-of-concept, a special edition of *Samtla* has been applied to the Old and New testaments of the *King James Bible* from the 1611 edition, in English. This corpus is used for demonstration purposes, and as the data for a formal evaluation, presented in Chapter 7.

### Giorgio Vasari archive



Figure 3.6: A portrait of Giorgio Vasari.

The fourth research group is the Vasari Research Centre. The documents represent chapters from the book *Lives of the Most Excellent Painters, Sculptors, and Architects* by Giorgio Vasari (1511 - 1574). Giorgio Vasari is considered to be the founding father of the Art History discipline [49].

The *Giorgio Vasari archive* consists of 314 documents in the original Italian and a corresponding English translation. The collection is used for research and as a teaching aid for students in class, and the search and mining tools facilitate the discovery of related material in both languages. A large number of images representing the paintings, sculptures, and architectural features also accompany the archive, which are important to the researchers of art history.

## 3.4 DISCUSSION

The framework presented in this chapter supports the search, browsing, comparison, and recommendation of archival content. The platform independent solution provided by the web browser promotes accessibility of the digital archive to a wider community of researchers than could be achieved with a desktop application. In addition, researchers may wish to access the system across a range of different devices, such as smart phones, and most of these devices come pre-installed with at least one web browser. A further advantage is that any system updates are available instantly when the user loads the web application, making it easy to maintain.

The *MVC* design pattern was adopted for its simplicity and ability to separate the system components according to responsibility. The architecture derived from the *MVC* pattern represents tried and tested techniques adopted for the development of scalable web applications that can support the access of resources by many distributed users [73].

*Samtla* operates with a range of different corpora that are stored separately in the *model*. The data represented each corpus is passed to the program logic of the *model* using a small set of wrappers to extract the text from common file formats.

When a change to the data is requested by the *view* as a result of the user interacting with the search or textmining tools, the *controller* component is responsible for validating the request and instigating an update to the data stored in the *model*. The *controller* is responsible for translating the users interaction with the interface through mouse, keyboard, and touch input in to a series of *HTTP* requests that are mapped to a function in the system, which instigate a change to the data stored in the *model* component. The *view* receives an update from the *model*, and renders the result of the change to the system state to the user.

A number of variations on the *MVC* pattern have been proposed. These include the Model-View-Presenter *MVP*, and Presentation-Abstraction-Control *PAC* where the components are organised in a hierarchical structure of abstractions, which allow the components, such as the view, to be represented by various levels of abstraction, each with their own purpose or utility. This pattern is not commonly adopted due to the complexity of designing an ap-

appropriate level of abstraction for even simple interactive elements of the user interface such as a query submission button. However, the idea is that the approach decouples the user interaction from the strict structure imposed by a layered system architecture (like the *MVC*) [84].

The simple *MVC* pattern has several benefits over other variants of the pattern, including; the ability to write reusable, and easily extendable code. In addition, the separation of the interface logic from the stored data makes the architecture of the system easier to update and maintain. This means that the *model*, representing the application logic and data, can be re-used across different web applications. In the same way, the *view* component can be represented by more than one type of UI, without interfering with the main program logic stored in the *model* (e.g. a system administrator user interface). For very large web applications, it is common to find developers working on separate components of the *MVC*, and the separation of concerns means they can work independently of each other, or specialise in one area of development at a time [73].

The architecture presented in the chapter is purely data-driven as a result of the character-level representation for the document index and the *SLM* responsible for scoring of sequences of characters in a given language. The approach allows any digital archive to be deployed very quickly, by substituting the data underling the *model* component. The tools are exposed through the *controller* component and their availability is conditioned on whether the data exists for the particular archive, which is defined by a separate configuration file. This means that, unless a new tool is developed in response to the new archive, there is no requirement to update the existing components each time a new collection is introduced. The data model reflected by the character-level *SLM* approach provides a lot of flexibility with respect to the archives that *Samtla* can operate with, as demonstrated by the case studies presented in Section 3.3 which addresses the need for a general-purpose infrastructure that can support the development of digital tools to address the varied needs of researchers in the humanities who use digital archives and digital representations of primary source material as a key resource for their research [184].

# CHAPTER 4

## SEARCH

This chapter introduces the data model stored under the *model* component of the architecture, which supports the search and mining tools developed for the *Samtla* system (see Chapter 3). The chapter begins with a brief overview of information retrieval in Section 4.1. One of the main components of the novel infrastructure, represented by the Statistical Language Model (*SLM*) is presented in Section 4.2. This is followed, in Section 4.3, with a description of the novel character-level  $n$ -gram *SLM* data model, and the character-level  $k$ -truncated suffix tree data structure used to store *SLM* to act as the index of the  $n$ -grams of the documents for the provision of full and partial matching of the  $n$ -grams in the query. The *SLM* assigns a probability to each  $n$ -gram of the documents, which are then retrieved in response to the  $n$ -grams of the query. Section 4.4 presents the query model used to rank the documents, which assesses the contribution of each  $n$ -gram according to the *SLM* for the document, which returns the probability that the document generated the  $n$ -grams of the query. The documents are ordered according to probability to create a ranked list of the documents with the most “relevant” document at the top of the search results. The retrieval performance of the *SLM* approach is very much dependent on how well the model parameters have been estimated [194]. This requires some form of smoothing, which plays an important role in producing an accurate model of the language recorded in the documents (discussed in Section 4.5). The approach presented in the chapter is also easily extensible to specialised information retrieval tasks, such as metadata search, which is illustrated in Section 4.6. Furthermore, several case studies are presented, which demonstrate the flexibility of the approach to search over a range of

different domains, languages, and research needs, in Section 4.7. Lastly, the chapter concludes with Section 4.8, which summarises the important aspects of the infrastructure.

## 4.1 OVERVIEW

Search engines are one of the most important and widely used technologies that exist today as a result of the dramatic increase in digital material available through the internet, on ebook readers, and in mobile applications. Common applications include databases where information is retrieved through a querying language complete with its own syntax, and internet browsers that return information from web pages consisting of both structured and unstructured text data, through natural language queries. Search engines enable users to extract information on a specific topic from both structured and unstructured text data [142]. The user has an information need representing a prototypical idea of a document describing the particular topic of interest. This information need is expressed through one or more terms, where a term defines a meaningful sequence of characters, known as  $n$ -grams, in the language. A search engine locates documents by identifying the  $n$ -grams of the query in the documents, and then generates a ranking of the documents according to the “importance” of each  $n$ -gram. The notion of “importance”, otherwise referred to as “relevance”, describes the users’ expectation of which documents should appear at the top of a ranked list of search results, in other words, which documents the user may be looking for [194].

Natural language data is very complex and a digital archive represents only a small sample of a much bigger population of natural language data. In order to capture accurate statistics of the distribution of a  $n$ -gram in the documents according to any given language, the  $n$ -grams of the documents are often normalised through a preprocessing step, and the weights for the  $n$ -grams are smoothed to reduce the contribution of the those that are not descriptive of the topic described by the users search intent. Another role of smoothing is to account for  $n$ -grams of the query that return no match for any of the documents, which occurs when users over specify their query, or commit spelling errors. Furthermore, the performance of information retrieval systems is often conditioned on the choice of document representation for the

$n$ -grams, the retrieval model adopted for scoring the documents according to the matching  $n$ -grams of the query, and the smoothing method selected to account for missing  $n$ -grams of the query, and those representing the syntax of the language.

## 4.2 STATISTICAL LANGUAGE MODELS (SLM)

This section introduces the novel and relatively new approach to document and query term weighting that provides a consistent approach to the indexing, ranking, and smoothing of a retrieval model applicable to any domain and language with very little preprocessing of the documents. The *Samtla* system adopts a novel approach for extracting and weighting the  $n$ -grams of a document through the application of character-level  $n$ -gram *Statistical Language Models (SLM)* stored in a space-optimised  $k$ -truncated suffix tree data structure. *SLMs* provide a consistent approach to the scoring of sequences represented by natural language data recorded in a digital archive of documents, which can often be cross-domain, and multilingual. The application of *SLMs* to information retrieval is a fairly recent topic of research [142,159,194], where their recent adoption has largely been motivated by their successful application to the domain of speech recognition [142]. *SLMs* are built on well established theoretical and statistical methods, for instance the Maximum Likelihood Estimation (*MLE*) method [194]. A *SLM* provides a principled approach to term weighting. Each document is represented by a separate  $n$ -gram language model storing the probability distribution of the  $n$ -grams occurring in the document. The relevance of a document, given some informational need, is expressed under a probabilistic framework, where each  $n$ -gram in the document is assigned a probability according to its frequency in the document, and the document collection as a whole. The retrieval task then involves ranking the documents by extracting the  $n$ -grams of the query from the language model for each document, and sorting the documents so that the top search result contains the most probable document.

One of the novelties of the approach is that a *SLM* can be straightforwardly extended to tasks other than search, including recommendation [125]. The character-level model selected for the document representation has also been a recent topic of research, where it has been shown to be effective in

spam-email filtering [116], authorship attribution [106], neural networks [121], and named entity recognition (*NER*) [123] compared to the word-level models, due to the additional semantics represented by word-internal features that are not available in word-level models without some degree of natural language processing, such as, word stemming and lemmatisation [142]. A character-level model reduces the  $n$ -grams of the documents to a set of overlapping  $n$ -grams representing the valid character-sequences of the language of the documents. Depending on the size of  $n$ , the character-level representation is able to capture affixes, words, collocations, and larger linguistic constituents such as set phrases which represent the “parallel passages” of interest to researchers (see Chapter 2). However, A character-level representation for the documents, has a higher dimensional space than the word-level equivalent, which increases the complexity and storage requirements of the resulting model, which until recently, has been one of the reasons for their lack of wider adoption [144]. There are several advantages to a character-level representation that make them potentially more effective for information retrieval and text mining tasks than the current word-level models, which include the following:

- A language-independent approach to term indexing. Some languages do not have an explicit delimiter that can be used to segment the text in to morphemes. Furthermore, some languages are highly inflectional, where a root word receives different prefixes and suffixes according to the rules of syntax for the language.
- Reduces the need for preprocessing of the documents since all characters of the input are treated equally [76, 144].
- The additional information captured at the sub-word level reduces some of the issues associated with data-sparseness [150], which is a problem often encountered by word-level  $n$ -gram representations, where there will be many more  $n$ -grams with a zero count for the document.
- A space-optimised approach, where the number of valid character combinations in a given language is less than the possible word combinations. In other words, new words can always be introduced in to the language e.g. names for people, locations, and products, whereas, the number of valid character-sequences according to the phonological and morpholog-



ical rules of a language are fairly consistent and stable.

The combination of a character-level representation for the documents with a *SLM* for modeling the probability distributions of the  $n$ -grams, provides a fast, efficient, and flexible approach to providing search and text mining tools to researchers of digital archives that can support the types of information seeking behaviour, such as phrase search, which is commonly adopted by researchers in the humanities (see Chapter 2). In the remaining sections of the chapter, the implementation details of the infrastructure represented by the *SLM* and suffix tree data structure is presented, which provides the means for quickly retrieving a set of documents, matching the  $n$ -grams of a query, using a domain and language independent approach.

## 4.3 DOCUMENT INDEX

This section introduces the index of the  $n$ -grams and documents, responsible for supporting the fast and tolerant retrieval of character-sequences of variable length. The *SLM* is stored in a space optimised character-level  $n$ -gram suffix tree data structure, which stores the complete model in memory for fast retrieval [100, 182]. Suffix trees are well known and understood data structures used for indexing the characters of a text string. Given a text string, the resulting suffix tree represents a compressed “trie” data structure containing all the suffixes of the string as their keys and positions in the string as their values. A more useful implementation, known as the generalized suffix tree, is a suffix tree constructed over multiple strings for a set of documents represented by a corpus rather than just a single string.

### 4.3.1 Constructing the index

Any string can be converted into a series of  $n$ -grams by sliding over the characters of the string one at a time, resulting in an index of all the suffixes contained in the sequence. As an example, Table 4.1 illustrates how the string “banana”, which has a  $n$ -gram size of seven characters, can be converted to lower-order  $n$ -grams of variable length: The resulting generalised suffix tree in combination with a fixed-size sliding window is a powerful structure that can be used to locate all instances of a string and the corresponding docu-

n-gram order	n-gram
7	bananas\$
6	ananas\$
5	nanas\$
4	anas\$
3	nas\$
2	as\$
1	s\$

Table 4.1: The conversion of the string “banana” in to character  $n$ -grams.

ments very quickly. The indexing process involves converting each document to a set of  $n$ -grams. Each  $n$ -gram is appended with a special character \$ to signify the end of the character-sequence, and inserted in to the suffix tree one character at a time. For each new character, a node is created labelled with the character, and a count is updated with the number of times the character has been observed during insertion. When the “\$” character is encountered, a special instance of the node called a “leaf node” is created, which stores a list of start positions for all instances of the  $n$ -gram and an identifier pointing to the original document (see Figure 4.1). The construction of a suffix tree is performed with a depth-first search of the tree, and can be summarised as follows:

1. Create a node and label it as the root  $R$  of the tree. This creates the point of access to the data structure.
2. Convert each input sequence to a list of  $n$ -grams and append a special character “\$” to mark the end of the sequence.
3. Starting with the first  $n$ -gram, begin inserting characters by comparing the first character of the  $n$ -gram with the labels of any child nodes of  $R$ .
4. If there are no children of  $R$  with a label matching the inserted character, create a new node, assign it a count of 1, and make this the current entry point to the tree.
5. As this is the first  $n$ -gram, each character is inserted as a child node of the preceding character node until all characters are exhausted.
6. When we encounter the character “\$”, then the sequence has ended and a “leaf node” is generated to store a pointer to the start position of the  $n$ -gram in the document, and the document ID.

7. Set the current entry point back to the root node  $R$ .
8. Begin inserting the next  $n$ -gram by comparing the characters with the labels of the children under  $R$ . If there is a match then update the count of the matching node, and continue the search down through the tree setting the current entry point to the last matched node.
9. If there is a mismatch, then we create a new node at this point, and insert the remaining characters of the  $n$ -gram.

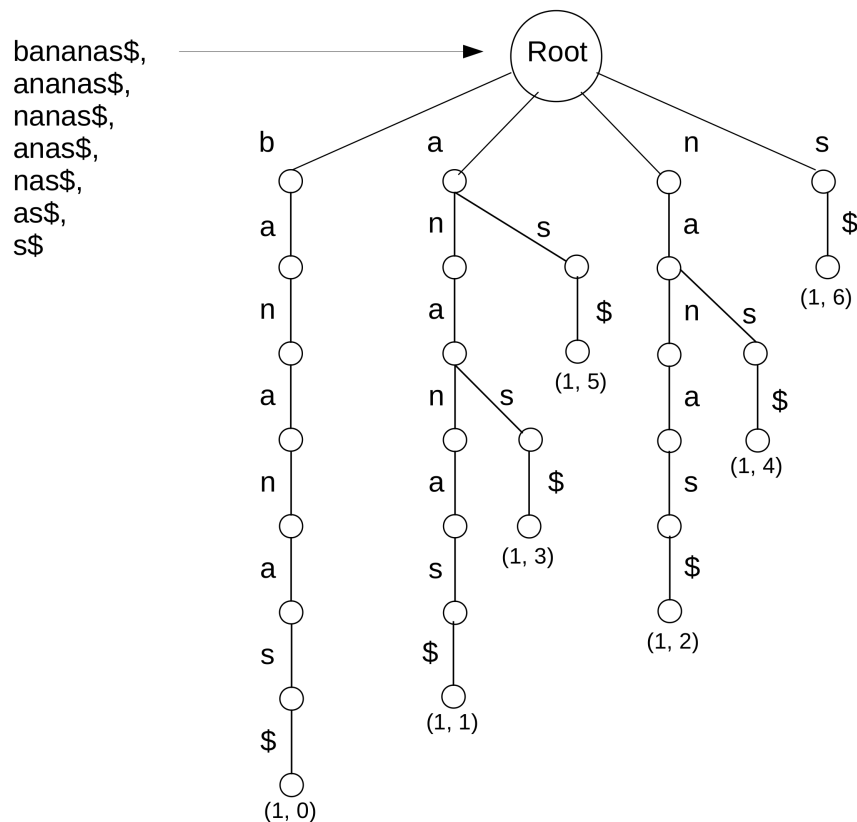


Figure 4.1: A suffix tree constructed from the string “bananas”.

### 4.3.2 Storing the index

One of the main challenges surrounding the suffix tree implementation is related to the cost of memory, which is considerably larger than the original input [61, 99, 182]. For instance, a 1MB text string would typically be represented by a suffix tree of size 10MB in memory. This is due to the way that strings are converted in to substrings of the original sequence.

Text strings can be quite large, and the depth of a suffix tree is determined by the length of the longest text string inserted, which would be represented by the full document text. This can be quite inefficient in terms of memory and disk storage due to the large number of internal nodes created. One method for reducing the memory requirement, is to limit the depth of the suffix tree to produce a  $k$ -truncated suffix tree [151,173], which restricts the maximum depth of the suffix tree to  $k$  nodes by setting the length of the sliding window to  $k$  characters. A high setting for  $k$  will enable the resulting data model to record sub-word level features (intra-word information), and between word features (inter-word information), but at the cost of a higher dimensional space [179].

The best value for  $k$  is determined by the average length of the words for the given language. McNamee and Mayfield (2004) [144] found that mean word length tends to correlate with the morphological complexity of the language, and consequently provides a good indicator of the best setting for  $k$ , where in their study they found that  $k = 5$  was sufficient for the majority of European languages in their study. The best value for  $k$  was determined by analysing the content of the archives with which *Samtla* currently operates. The majority of the words in the documents were found to be no longer than 15 characters in length, which is also supported by Figure 4.2 (adapted from [47]), which plots the average word length for several languages that differ with respect to their morphological complexity. Based on the average word length for each language in the plot, a setting of  $k = 15$  would seem sufficient for capturing complete word instances. However, languages such as Faroese, which attach affixes to a root form, may require a higher setting for  $k$ , since the addition of affixes can result in relatively long character sequences. Unigram models have often been adopted due to the fact that they are easy to implement, and more efficient in terms of estimation and storage, than the higher-order markov model adopted by the proposed infrastructure. However, the advantage of the approach proposed in this chapter, is that higher-order  $n$ -gram models (up to  $n=15$ ), provide support for phrase and proximity queries due to the dependency captured between the terms in higher-order language models, which has been an approach proposed as part of future work in the development of more sophisticated language models [194].

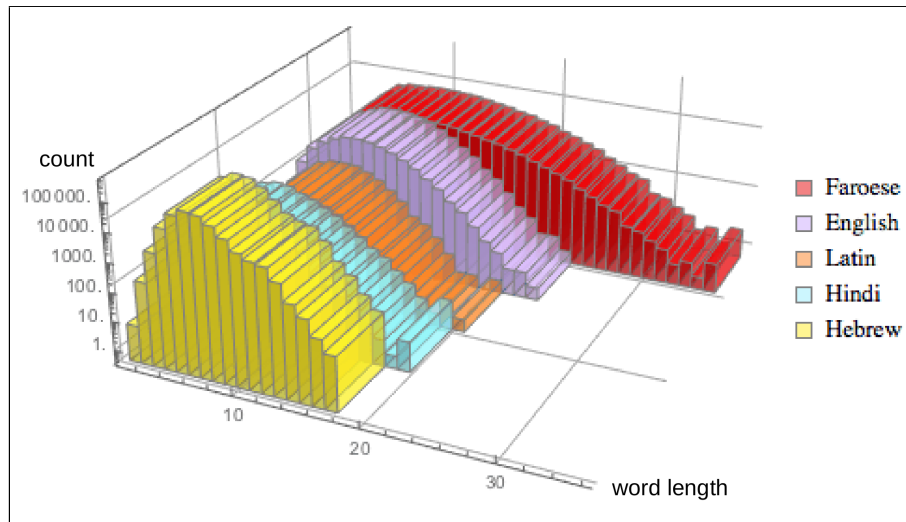


Figure 4.2: Average word length across several language groups represented by Faroese, English, Latin, Hindi, and Hebrew.

### Dereferencing

A further optimisation, known as dereferencing, assigns a unique number to each character of the input. This allows the suffix tree to represent any datatype, for example, a 1 can represent the character “a”, but equally it could also represent a syllable, word, or even a clause. As long as the original representation of the string can be obtained when comparing the node labels with a query string during search, anything can be indexed by dereferencing the input. This reduces the memory requirement further as integers are smaller than character strings (in Python 2.7 an integer consumes 12 bytes, whereas a string requires 32 bytes) [34].

### Path compression

Another method for reducing the space requirement of the suffix tree is to compress dangling nodes, which are nodes that have only a single descendent (or child). These nodes are gathered together during a depth-first-search and stored as a ‘supernode’, whose label is constructed from the concatenation of the collected node labels [100], see Figure 4.3. Dangling nodes will have the same count as they only contain a single child node and therefore we do not lose any information.

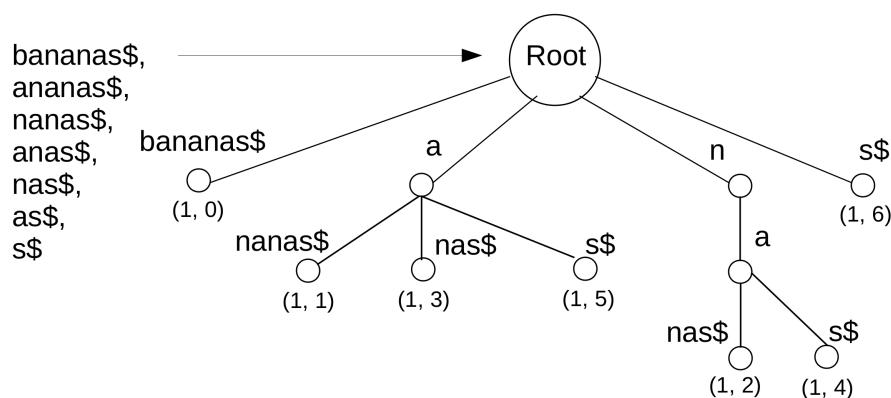


Figure 4.3: A compressed version of the suffix tree where the super-nodes are rendered as ellipses.

### On-disk construction

For real world applications, disk-based implementations of suffix trees [61] are more common, where the intermediate tree is flushed to disk periodically when the size of the tree grows to a specified limit in memory. Disk-based solutions are not without their own separate issues, for instance, disk trashing caused by successive reading and writing of the data, and latency caused by the time required to locate the correct sector of the disk for loading parts of the tree to memory.

#### 4.3.3 Searching the index

Search over a suffix tree is performed by starting at the root node and then descending the tree along a unique path by comparing characters of the query with the label stored at each node. When the characters of the query are exhausted or a mismatch occurs, the sub-tree rooted at the last-matched node is traversed with a breadth-first traversal, and all leaf nodes are collected to produce an index of tuples containing document IDs and start positions. Suffix trees allow us to capture both full and partial matches during the traversal as we are always returning the last matched node.

The language agnostic nature of the implementation has been tested with a number of corpora, which are both different in terms of their structure, but also in terms of the language, dialect, script, and domain. Suffix trees are useful structures for providing fast and flexible search over both unstructured data,

as that represented by the documents, and structured data represented by key-value databases; see Chapter 5 for a discussion on the differences between the two implementations. After construction, the suffix tree finds all instances of a character string in time linear to its length [100]. We can then identify all the documents containing the character sequence along with the start positions in the document. In essence, we have an index over a document collection, which provides the means for constructing a range of applications for information retrieval, classification, and clustering. Information retrieval is the primary use of the suffix tree in *Samtla*, and consequently we require an additional component for scoring documents retrieved by the suffix tree according to their importance to the search string.

## 4.4 QUERY RESULTS RANKING

Statistical language modeling is central to *Samtla*'s data model and are the foundation of *Samtla*'s search tool allowing users to locate documents through full and partial matches to queries. In *Samtla* the more probable a document is in the *SLM* sense, the more relevant it is to a users query. This enables us to avoid the philosophical debate on the meaning of "relevance" [148,171]. Instead, the system retrieves the most probable documents described by the *SLM* representing the probability distribution of the  $n$ -grams in the corpus being searched.

This means that documents are considered relevant to the query according to the notion of *systems relevance* or *algorithmic relevance* which describes how well the topic of the documents matches the topic of the query. The assumption of relevance under this context is that the users intent is to retrieve a subset of the documents that are highly relevant to a query [?]. In other words, the documents in the corpus being searched are sorted by the system according to the probability of the  $n$ -grams of the documents generating the the users' query. The most relevant documents will be those with the highest probability and ranked in the top positions of the search results.

An *SLM* is generated from the whole collection of documents, referred to as the *collection model*, and over each individual document, which we call the *document model*. The collection model  $C$  describes the global probabilities of the  $n$ -grams in the archive. As a result, it is not possible to identify a particular

document as being more important than another. Therefore, a further model is introduced to store the local probability of the  $n$ -gram according to each individual document, known as the *document* model  $D$ . The document model  $D$ , stores each individual  $SLM$  generated over the  $n$ -grams extracted for the given document. This section introduces the two models in more detail, and discusses their application to the ranking of the documents against the user's query. In the remaining sections of the chapter, a generic  $SLM$  will be denoted by  $M$ , whilst the collection and document models, are denoted by  $C$  and  $D$ , respectively. Each  $SLM$  stores the  $n$ -grams for the whole archive or each individual document, where  $n$  is taken to be a single character, or a character-sequence upto some pre-determined maximum number of characters  $n = 15$ .

The query  $n$ -grams submitted by the user represent a model of the users information need and provide the mechanism for estimating the probability of the query according to the language model produced for each document [127]. In short, each  $n$ -gram is extracted from the query and used to retrieve an estimate of its probability given a language model for that particular document in  $D$ . The documents are then ranked according to the extracted probabilities for the query, and ordered so that the documents containing the highest probability of matching the query are displayed at the top of the ranked list of search results [178]. Bayes theorem is one of the key principles underlying the  $SLM$  approach, and is well-motivated for information retrieval, as the aim of the task is to reduce the collection of documents to a small subset containing the most relevant documents, according to the users' query, which is not known in advance [137, 159]. Using Bayes theorem [134] allows the problem to be reduced to calculating the *posterior* probability  $P(A|B)$ , which is the probability of event  $A$  given event  $B$ , once we have established that we have already observed the event described by  $B$  [142, 147]. Using Bayes theorem and replacing the event  $A$  with the  $n$ -gram  $q$  to represent a query, and the event  $B$  with a language model  $M$ , then we can define a query model  $P(q|M)$ , denoted by  $P_M(q)$  and read as "the probability that the query  $q$  was generated by the language model  $M$ " to calculate the probability that a document is relevant to the query, through:

$$P(M|q) = \frac{P(q|M)P(M)}{P(q)} . \quad (4.1)$$



The conditional probability  $P(M|q)$  represents the conditional probability of the language model  $M$  given the query  $q$ . When  $M$  is a document  $D$ , this is the probability of the document  $D$  when the query is  $q$ , which will allow the system to rank the documents returned to the user. Thus, when a user submits a query, *Samtla* will compute the probability that the query was generated by the model,  $M$ , where  $M$  is either a collection model  $C$ , representing the (global) probabilities of the  $n$ -grams, or a document model  $D$  which stores the (local) probabilities. In other words, *Samtla* will compute the probability that a user who is interested in a given document would submit that query. The documents are then ranked according to the computed probabilities for each document, and the top scoring documents are returned to the user.

The collection model  $C$ , is stored as a space-optimised character-based suffix tree, whereas from an implementation perspective, the document model  $D$ , identifies each document by a unique document  $id$ , and the  $n$ -grams provide an index pointing to a list of tuples representing the document ID and probability inferred from the language model. Using a SQL database for the document model  $D$ , allows the probability distributions to be made available to other system components (see Chapter 5). The assumption throughout the remainder of this section is that a query  $q$  is taken to be a sequence of characters, meaning that there is no notion of a “word” or “morpheme”, simply a series of individual characters of a specified length  $m$ .

The right-hand side of (4.1) consists of the query model  $P(q|M) = P_M(q)$  multiplied by a *prior* probability,  $P(M)$ . The prior probability of the model  $M$ , when  $M$  is a document model  $D$ , is its presupposed probability, that is, the predefined importance of each document regardless of the submitted query. The current version of *Samtla* does not support a prior in the query model; although see Chapter 5 for more on how one could be incorporated. The prior probability for each document is therefore assumed to be uniform, i.e. the same for all documents. When the prior is uniform, the documents are ranked according to  $P_D(q)$ , and the probability of the query denoted by the denominator  $P(q)$ , is the same for all documents and removed from the equation for the purpose of ranking. If we assume that any query  $q$  can be represented as a sequence of  $m$  characters, such that  $q = c_1, c_2, \dots, c_m$ , then we

can define the probability of a query as:

$$P_M(q) = P_M(c_1, c_2, \dots, c_m) = \prod_{i=1}^m P_M(c_i | c_1, \dots, c_{i-1}). \quad (4.2)$$

Calculating the  $n$ -gram probabilities requires the adoption of the “Markov assumption” [146, 147]. The Markov assumption allows the probability of each  $n$ -gram to be approximated by the conditional probability of the  $n$ -grams preceding it, known as the  $n$ -gram history (or context). This means that we only make use of its  $n - 1$  character history (or less than  $n$  for shorter sequences) as an approximation to the conditional probabilities in (4.2). This makes sense as higher-order  $n$ -grams representing collocations or phrases occur less often than smaller  $n$ -grams represented by words, meaning there is less information on which to calculate a reliable probability. We can partly compensate for this issue by approximating the probability of the higher-order  $n$ -grams using the more reliable estimates of the lower-order  $n$ -grams. This enables the system to include the notion of dependency between large sequences and small sub-sequences, that is, the dependency or relationship between higher-order structures (such as clauses, and collocations), and lower order  $n$ -grams describing the syntax of a language.

A common approach adopted for approximating the conditional probability,  $P_M(c_i | c_1, \dots, c_{i-1})$ , on the right-hand side of (4.2), is the maximum likelihood estimator (*MLE*) [58, 114, 142, 150], where the count of a  $n$ -gram is normalised with the total occurrence of  $n$ -grams containing the same prefix, the *MLE* is defined more formally as:

$$MLE_M(c_i | c_{i-n+1}, \dots, c_{i-1}) = \frac{\#(c_{i-n+1}, \dots, c_i)}{\#(c_{i-n+1}, \dots, c_{i-1})}, \quad (4.3)$$

where the  $\#$  symbol before a sequence indicates that we are dealing with raw counts in the model  $M$ . The history is denoted by  $n + 1$  meaning we are referring to the  $n + 1$  characters preceding the current character  $c_i$ . Furthermore, for any sequence of characters we have  $1 \leq i \leq m$ , and  $MLE_M(c_1 | c_0)$  is taken to be  $MLE_M(c_1)$ .

Recall from Section 4.3, that each internal node of the suffix tree stores a frequency count for the character. These counts are used as the basis for constructing a probabilistic suffix tree to act as the collection model  $C$ . The

tree is traversed starting at the root node, and we divide the count of each node by the count of it's parent to obtain the conditional probability of the current node, defined earlier in (4.3), and illustrated in Figure 4.4 below. The main

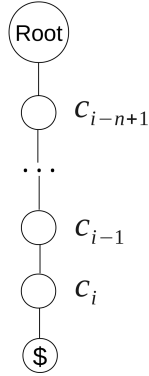


Figure 4.4: Calculating the *MLE* for each order of *n*-gram stored in the collection model *C* represented by the suffix tree data structure.

issue with the *MLE* is that it has a tendency to overfit the data, resulting in there being no probability mass available for unseen character-sequences [150]. That is, if the *n*-gram has not been observed, or the user makes a typographical error, then some *n*-grams of the query will not be present in any of the documents. Due to the character-based representation of *n*-grams, we are still able to capture partial matches to the query. The initial *MLE* scores require tuning in order to calculate a more accurate model of the language. A *n*-gram language model requires the estimation of  $O(K^n)$  free parameters, representing the unique *n*-grams in the archive [150], where *k* represents the number of unique words, or characters of the language, and *n* defines the order of the markov chain used to calculate the probability of the *n*-gram based its context. If we have twenty-six characters ( $k=26$ ) representing the finite alphabet of the English language, and a 3-gram language model ( $n=3$ ), then there are as many as 17,500 parameters to estimate, corresponding to all possible 3-gram character-sequences. In reality, the number of parameters is often lower since there are language-dependent rules that determine which character-sequences are valid as a result of the phonology, morphology, and orthographic conventions of the language.

## 4.5 SMOOTHING

To recap, information retrieval involves the application of methods and algorithms to the scoring of a set of documents according to the informational need of the user. It is important to note that the user is not necessarily familiar with the document collection. Consequently, the user's assessment of the system's utility will be determined by how well the retrieval model ranks according to a prototypical document that the user has in mind, which they describe through the  $n$ -grams of their query. Smoothing the  $SLM$  plays two important roles:

1. The first role of smoothing, adjusts the  $MLE$  estimates to account for the properties of the language represented by the syntax of the language. Researchers often submit a range of queries from short keywords to long verbose queries (see Chapter 2). Long verbose queries tend to be closer to natural language as they represent well-formed and grammatical sequences with many uninformative  $n$ -grams representing the syntax of the language (e.g. prepositions *of*, *in*, *to*, *by*, and determiners *a*, *the*). The vast majority of the documents will contain these  $n$ -grams with a relatively high probability compared to the  $n$ -grams representing the actual topic described by the document, such as the content words like nouns, adjectives, and verbs. As a result smoothing the estimates helps to reduce the negative impact of these  $n$ -grams on retrieval performance [107].
2. The second role of smoothing is to adjust the initial  $MLE$  estimates to compensate for certain  $n$ -grams of the query that may not be present for some of the documents. This is sometimes caused by a lack of familiarity with the content of the document collection, typographic errors, and novel sequences (for example acronyms representing company names or new developments e.g. *SSD* - Solid State Drive). When  $n$ -grams of the query do not match in the collection model  $C$ , represented by the suffix tree, or a specific document model  $D$  then the score for the  $n$ -gram will be evaluated as being zero. This is an issue for retrieval models that score documents through the product of the term weights [114,142], since the score for the document will also be evaluated as zero despite

matching some of the  $n$ -grams of the query. This is resolved by adjusting the  $n$ -gram weights through the assignment of a small non-zero value to the  $n$ -gram.

In short, smoothing has been shown to play a very important role in compensating for issues associated with missing and very common  $n$ -grams [196], and are an integral part of the *SLM* approach [194]. Smoothing can be achieved through a number of strategies including *additive* smoothing [78, 142]; where we increase the count of each  $n$ -gram by 1, *discounting* [150, 157]; where a fraction of a  $n$ -gram's probability mass is redistributed to other  $n$ -grams in the corpus. These methods assume that each new sequence is equally likely to occur, however this is not how language functions [150]. An approach adopted in *Samtla* is the *linear interpolation* method, which combines the scores from the document model  $D$ , and the collection model  $C$ , using a weighting scheme to define the contribution from each model to the final smoothed probability for the document [196].

### Smoothing the common $n$ -grams of the language

Frequently occurring  $n$ -gram are smoothed in two stages. First, the initial *MLE* probabilities for each  $n$ -gram are smoothed by interpolating them with information from lower-order  $n$ -grams, in (4.3), which is sometimes referred to as *deleted interpolation* [78]. The lower-order  $n$ -grams tend to have more reliable statistics as they are calculated on the basis of more data. A weighted term defines the contribution to the overall probability for each order,  $k$ , where  $k$  varies from a zero order, 0-gram model, when  $k = n + 1$ , to a  $n$ -gram model, when  $k = 1$ . Each weight, represented by  $\lambda_k$ , defines the amount of interpolation, with lower-order models contributing less to the final probability. The approximation of the conditional probabilities on the right-hand side of (4.2) is given by the interpolation,

$$\hat{P}_M(c_i|c_1, \dots, c_{i-1}) \approx \sum_{k=1}^{n+1} \lambda_k MLE_M(c_i|c_{i-n+k}, \dots, c_{i-1}), \quad (4.4)$$

where we use  $\hat{P}$  to make clear that we are approximating  $P$ , and the weighted term for each  $k$  is given by

$$\lambda_k = \frac{n+2-k}{(n+1)(n+2)/2}, \quad (4.5)$$

where  $n$  is the order of the  $n$ -gram, which is generated by interpolating the  $n$ -th order model with a lower-order model. When  $k = n + 1$ , then  $MLE_M(c_i|c_{i+1})$  is taken to be the 0-gram model,  $\frac{1}{|V|}$ , where  $V$  is the finite alphabet of the language (for English this is 37 representing the characters of the Roman alphabet, numbers 0 to 9, and the whitespace character). As an example, suppose we have the 5-gram ( $n=5$ ) “abcde”, and we wish to approximate the conditional probability  $P_M(c_i|c_{i-1}^{i-1-n+1})$ . Each order of  $n$ -gram, based on a 5-gram language model, is interpolated with lower-order  $n$ -grams by reducing the history of the 5-gram one character at a time from the start of the string through (4.6). We extract the corresponding  $MLE$  score for each  $n$ -gram order, and combine it with the appropriate  $\lambda$  weight to define its contribution to the approximation of the probability for the 5-gram, see (4.5):

$$\begin{aligned} P_M(c_i|c_{i-1}^{i-1-n+1}) = & \\ & \lambda_{\frac{6}{21}} P(c_i|c_{i-1}c_{i-2}c_{i-3}c_{i-4}) + \\ & \lambda_{\frac{5}{21}} P(c_i|c_{i-1}c_{i-2}c_{i-3}) + \\ & \lambda_{\frac{4}{21}} P(c_i|c_{i-1}c_{i-2}) + \\ & \lambda_{\frac{3}{21}} P(c_i|c_{i-1}) + \\ & \lambda_{\frac{2}{21}} P(c_i) + \\ & \lambda_{\frac{1}{21}} \frac{1}{|V|} \\ \sum \lambda_k = & 1 \end{aligned} \quad (4.6)$$

The contribution for each order of  $n$ -gram is illustrated in Table 4.2, which shows the value calculated for each weight on the basis of our 5-gram example. The smoothing of each  $n$ -gram in (4.4) is an offline process. For the collection model  $C$ , smoothing is performed through a second traversal of the suffix tree where the interpolation is applied to the  $MLE$  scores stored at the node for each order of  $n$ -gram. The document model  $D$  for each document is smoothed by extracting all  $n$ -gram  $MLE$  scores for a particular document, and then

$n$ -gram	$\lambda_k$	weight
abcde	6/21	0.2857
bcde	5/21	0.2380
cde	4/21	0.1904
de	3/21	0.1428
e	2/21	0.0952
$\frac{1}{ V }$	1/7	0.0476

Table 4.2: The lambda weights defining the contribution of the probability for each order of  $n$ -gram, to the final approximation of the 5-gram “abcde”.

applying the interpolation over each order of  $n$ -gram, and finally, updating the database record. The smoothed  $n$ -gram scores are retrieved from the collection model  $C$ , together with the document model  $D$  for each document contained in the index constructed by the suffix tree at query time.

Ranking the documents against a query requires an online process, as we do not know the terms of the query in advance. There are two issues that need to be compensated for in this instance, first we need to reduce the influence of common terms, such as prepositions and determiners, which do not have very much descriptive power since they will appear in a large number of documents, but have a high probability of occurring and therefore have a big influence on the query score. Secondly, some terms of the query may not exist in the document model for a given document, and consequently we need to approximate the probability using the information available.

The *Jelinek-Mercer smoothing* ( $JM$ ) method is adopted to reduce the influence of common terms, where it has been shown to be well-suited especially for long verbose queries [196, 197]. The method involves a linear interpolation of the document model  $D$  with the collection model  $C$  using a coefficient represented by  $\lambda$  to control the influence of each model to the final probability of the query given the document.  $JM$  smoothing is defined more formally below, where the term  $\hat{P}$  in (4.2) is substituted by  $P$ :

$$P_D(q) \approx \lambda \hat{P}_D(q) + (1 - \lambda) \hat{P}_C(q), \quad (4.7)$$

where  $\lambda = 0.6$ , which equates to a 60% contribution from the probability estimate for the query generated by the document model  $D$ . The document model  $D$  contributes more to the query score in order to distinguish the documents from one another, otherwise, too low a setting for  $\lambda$  would result in a ranking

based on the statistics of the collection model  $C$ . The additional smoothing step in (4.7) is required, as even after the initial interpolation, the maximum likelihood document model probabilities may be low, while the maximum likelihood collection model probabilities will provide a better global estimate of the probability. In other words, the terms of the query may occur only once in the document, and so the collection model provides more information about the actual frequency of the term given the statistics from the whole corpus.

The value of  $\lambda$  can be tuned through experimentation, in the case of *Samtla*, the appropriate setting was chosen on the basis of empirical assessments made by the researchers in the case studies. Long verbose queries require more smoothing than keyword or title queries, due to the number of uninformative terms, and in these situations, a higher setting for  $\lambda$  is more appropriate [196].

### Smoothing the missing $n$ -grams of the query

When documents match only part of the query, an additional smoothing step in the form of a *backoff* process, estimates the probability for the missing  $n$ -gram of the query according to information from lower order  $n$ -grams. The backoff is an iterative process that attempts to locate the next available lower order  $n$ -gram, by removing a character from the beginning of the sequence. The backoff process terminates when a lower order  $n$ -gram is located in the document model  $D$ , or when we backoff to the 0-gram model. If a match is found for the lower order  $n$ -gram, we calculate the correct probability for the  $n$ -gram on the basis of the appropriate  $n$ -gram model reflected by the full query. To illustrate the process, we assume a 14-gram query represented by the character-sequence “the Lord Jesus” submitted as a query over the *King James Bible* archive. The query returns a small subset of the documents in the collection with both full matches, and partial matches represented by the 5-gram “Jesus”.

The contribution of each  $n$ -gram stored in the model  $M$  is defined by the approximation in (4.4), which means that the partial match for a term of the query will have a higher probability given a 5-gram model compared to the 14-gram model representing the full query. This is due to the fact that we interpolate each order of  $n$ -gram according to the 5-gram model instead of the 14-gram model, where the contribution of the lower-order 5-gram is relatively



lower according to the weighted terms defined in (4.5).

To compensate, the backoff process locates a lower order  $n$ -gram to approximate the missing  $n$ -gram of the query, and adjusts its contribution to the final smoothed probability through a weighted term  $\mu$ , defined as  $\frac{n+1}{n+2}$ , adjusts the contribution of the missing  $n$ -gram to match the appropriate  $n$ -gram model for the query. However, if the lower order  $n$ -gram does not exist, we store the intermediary result of  $\mu$ , and continue the backoff process. More formally, we have

$$\begin{aligned}
 P_M(s|the\ Lord\ Jesu) = & \\
 & \mu_{\frac{15}{17}} + \\
 & \mu_{\frac{14}{16}} + \\
 & \mu_{\frac{13}{15}} + \tag{4.8} \\
 & \mu_{\frac{12}{14}} + \\
 & \dots \\
 & \mu_{\frac{6}{8}} P(s|Jesu) +
 \end{aligned}$$

where the backoff starts from the 14-gram model, and terminates at the 5-gram model, where we obtain the probability for the lower order 5-gram “Jesus”. The probability for the full query given the model  $P_M(s|the\ Lord\ Jesu)$ , is then the sum of the values stored during the backoff process. The approach to smoothing defined by the presence of absence of certain  $n$ -grams of the query can be generalised as follows:

$$\hat{P}_M(c_i|c_1, \dots, c_{i-1}) \approx \sum_{k=1}^{n+1} \begin{cases} \mu_n P_M(c_i|c_{i-n+1}), & \text{if } P_M(c_i|c_{i-n+1}) > 0 \\ \mu_n, & \text{otherwise} \end{cases} \tag{4.9}$$

where  $n$  is the length of the lower order  $n$ -gram we are backing off to, and  $\mu_n = \frac{n+1}{n+2}$  is the weighted term defining the contribution of the lower order  $n$ -gram probability to the approximation of the full query, if it exists for, otherwise, we store the result of  $\mu_n$ , and continue to back off.

To summarise, the backoff process compensates for missing  $n$ -grams of the query, which are not stored in the document model  $D$  for a given document, by re-estimating the contribution of any matching lower order  $n$ -grams to the

final query score for document according to the appropriate language model. When documents only match part of the query, it is necessary to adjust the probability estimates that were calculated offline, by approximating them using the lower order  $n$ -gram model where it is available.

In conclusion, smoothing is an important aspect of statistical language models as it produces a more accurate statistical model of the  $n$ -grams than that provided by the  $MLE$  alone, which maximises the likelihood function based on the observed data, but does not take in to account unseen events, such as those represented by missing  $n$ -grams [150]. The interpolation method allows us to take in to account information from lower order  $n$ -grams, which have more reliable counts, to better approximate the probability for a given  $n$ -gram. Furthermore, the interpolation between the collection model  $C$ , and the document model  $D$ , allows the system to account for the non-descriptive terms of the query, whilst the backoff process compensates for missing terms as a result of poor query formulation, on the part of the user either due to over specification of the query, or spelling errors in the document text, which may result in a mismatch for some or all of the terms of the query.

The  $JM$  smoothing approach provides a lot of flexibility with respect to extending the basic language model through the combination of any number of language models using a weighted term to control the contribution of each model. For example, in Chapter 5, an extension to the retrieval model is presented, which involves the addition of a  $SLM$  over the metadata record for each document.

## 4.6 METADATA SEARCH

Allowing researchers to identify relevant documents according to the matches for the query in the title text has been shown to be a particularly effective form of support for search, since authors may not make explicit mention of the subject matter of the document in the body of the text [86]. Search over structured data, such as the metadata that tends to accompany the documents, was developed to support the retrieval of documents in the *British Library Microsoft archive*, which contains extensive metadata about the documents, but very poor quality  $OCR$  for the document text.

Metadata search is supported through the addition of a  $SLM$  constructed

from the metadata record for each document, which is stored in an additional suffix tree data structure, defined as the metadata model  $B$ . The metadata model  $B$  represents a global language model of the metadata, and an index of all the  $n$ -grams in the metadata. The difference between the metadata model  $B$  and the collection model  $C$ , is the further storage of the metadata field column label together with the document  $id$ , and start positions of the  $n$ -grams.

Both the collection model  $C$  and metadata model  $B$  are traversed at search time, and the resulting indexes are retrieved for ranking the documents, together with the probability for the query extracted from the last matched node in each suffix tree. The original query model, is updated through a linear interpolation of the metadata model  $B$  using a weighted term to define its contribution to the smoothed probability of the document given the query  $P_D(q)$ , as follows

$$P_D(q) \approx \lambda_1 \hat{P}_D(q) + \lambda_2 \hat{P}_C(q) + \lambda_3 \hat{P}_B(q), \quad (4.10)$$

where  $\lambda_i$  is the weighted term that defines the contribution according to each of the  $SLMs$  represented by the document model  $D$ , collection model  $C$ , and the additional metadata model  $B$ . The weights were derived empirically through feedback from the research groups, with the document model  $D$  contributing the most to the final probability with a setting of  $\lambda_1 = 0.5$ . The collection model  $C$ , is interpolated with the weight  $\lambda_2 = 0.4$  to maintain its role as a global estimate of the probability for the query. The weight for the metadata model  $B$  is equivalent to  $\lambda_3 = 0.1$ . The resulting distribution of weights, means that matches for the query in the document text are more important, as they tend to provide a better description of the topic described by the content of the document. The  $SLM$  combined with the  $JM$  interpolation smoothing approach, provides a unified, consistent, and flexible query model, which can be easily supplemented with any number of  $SLMs$  to incorporate additional information in to the ranking of the documents, through the definition of an appropriate weighting scheme.

The flexibility of the approach is demonstrated by the range of digital archives that are supported, which are not always consistent with regard to the breadth and depth of the available data. For example, the *British Library*

*Microsoft archive* contains only a metadata model  $B$ , and so the lack of the document model  $D$  and collection model  $C$  components are compensated for in the query model through the backoff process, where the probability for the  $n$ -gram of the query given a missing  $SLM$  is evaluated on the basis of the 0-gram model (representing  $\frac{1}{|V|}$ , where  $|V|$  is the finite alphabet of the language). This ensures that the documents with matches for the query in more than one of the  $SLMs$ , will always be ranked higher in the search results. When there is no metadata for the digital archive, then the interpolation in (4.10) is performed between the document model  $D$  and the collection model  $C$ , which represents the original  $SLM$  query model in (4.7).

## 4.7 CASE STUDIES

This section presents several case studies demonstrating the flexibility of the search component to different digital archives. *Samtla* supports the retrieval of documents in a range of languages, including English, Aramaic, Hebrew, Syriac, Russian, German, French, Hungarian, and Malagasay.

### Aramaic Magic Bowl archive

The screenshot shows the search results for the query "חתיים ומחתם ביתה ואיסקופתה דודי ב". The results are displayed in a list format, with each entry showing a snippet of text and an etymology note. The snippets are highlighted in yellow, showing the search terms in context.

**AIT Bowls CAL 23 AIT 21**  
 חתיים ומחתם ביתה ואיסקופתה דודי בתה אחת מיכל פגעין בישי מיכל רוחין בישי ומן מבכלתא ומליתא ומיכל זיקין ומזיקין דלא תקרבון לה לביתה ולאסקופתה דודי בת אחת דחיתם ביתלתה עזקין ומחתם בשבעה חתמין ומיכל פגעין בישי ומיכל רוחין בישי ומימבכלתא מן אמן ואמן סלה...  
 Etymology: oday) is a hypocoristic from דודי, "friend, uncle," etc.; cf. the biblical name דודי and its variant דודי, also Dada in Bowl 12. The Parent: אהת

**AIT Bowls CAL 25 AIT 23**  
 ... רוחין בישי ומן מבכלתא ומן ליתא ומיכל זיקין ומזיקין דלא תקרבון לה לביתה ולאסקופתה דודי בת אחת דחיתמן ביתלתה עזקין ומחתם בשחתים ומחתם ביתה ואיסקופתה דודי בת אחת מיכל פגעין בישי ומיכל רוחין בישי ומן מבכלתא ומן ליתא ומיכל זיקין ומזיקין...ה עזקין ומחתם בשבעה חתמין מבכלתא ומיכל זיקין ומזיקין דלא תקרבון לה לביתה ולאסקופתה דודי בת אחת אמן אמן סלה  
 Etymology: oday) is a hypocoristic from דודי, "friend, uncle," etc.; cf. the biblical name דודי and its variant דודי, also Dada in Bowl 12. The Parent: אהת

**AIT Bowls CAL 24 AIT 22**  
 ... רוחין בישי ומן מבכלתא ומן ליתא ומיכל זיקין ומזיקין דלא תקרבון לה לביתה ולאסקופתה דודי בת אחת דחיתם ביתלתה עזקין ומחתם בשחתים ומחתם ביתה ואיסקופתה דודי בת אחת מיכל פגעין בישי ומיכל רוחין בישי ומן מבכלתא ומן ליתא ומיכל זיקין ומזיקין...ם בשבעה חתמין ליתא ומיכל זיקין דלא תקרבון לה לביתה ולאסקופתה דודי בת אחת מן יומה זין ולעלם אמן אמן סלה  
 Etymology: oday) is a hypocoristic from דודי, "friend, uncle," etc.; cf. the biblical name דודי and its variant דודי, also Dada in Bowl 12. The Parent: אהת

**Hilprecht Bowls Hilprecht 3**  
 ...ע חתמין מן כל פגעין בישי מן כל רוחין בישי מן מבכלתא ומן ליתא ומן כל זיקין ומזיקין דלא תקרבון לה לביתה ולאסקופתה דאדק בר מחלפתא ולממי אינה מחתים ומחתם ביתה אי ואיסקופתיה דאדק בר מחלפתא ..... אינה מן כל פגעין בישי ומן כל רוחין בישי ומן

Figure 4.5: The search results for a query representing a “parallel passage”, submitted to the *Aramaic Magic Bowl archive*.

The query “Sealed and countersealed are the house and threshold of Dodi bath Ahath from all evil Plagues, from all evil Spirits, and from the Tormentors, and from the Liliths, and from all Injurers, that ye approach not to her, to the house and threshold of Dodi daughter of Ahath.” [149], represents a “parallel passage” repeated across several documents. The name of the client “Dodi daughter of Ahath” appears across several of the documents, which suggests that the client had more than one bowl commissioned on their behalf. The scribes who authored the bowls on behalf of their clients, inserted set phrases from oral history and well-known religious texts such as the Aramaic translation of the Bible. The researchers studying these texts require very tolerant search tools to facilitate the discovery of these set phrases. The search results, in Figure 4.5, are composed of full and partial matches to the query,

which reveal three very similar documents *AIT Bowls CAL 23 AIT 21*, *AIT Bowls CAL 25 AIT 23*, and *AIT Bowls CAL 24 AIT 22*.

### Financial Times newspaper archive

The screenshot shows a search interface for the Financial Times archive. The search bar contains the query "american air strikes over north ...". Below the search bar, five search results are displayed, each with a title, a snippet of text, and a copyright notice. The results are:

- Easter Road Deathers Up—86 Die in Four Days**: Snippet: "... there was a 10- mile solid jam-accenuated by major road repairs. Bomb shortage American air strikes over Vietnam have been temporarily redu... t to complete the course, and named second overall. were Kenvans Vic Preston and Bob Gerrish (Ford Cortina). Ford won the team prize. Two Honda-powered B... temporarily reduced because S. Vietnamese workers at Da Nang have f  
Article subtitle: Record Review Copyright holder: The Financial Times Limited. All rights reserved
- Tories Will Strive for Debate on Gibraltar**: Snippet: "...were sho down during 139 sorties over North Vietnam on Monday, saic U.S. officia... medal total at th Kingston (Jamaica) Common wealth Games rose to 12 as- Charles S...carriers are to concentrate on North Vietnam attacks, said Washington. Br  
Copyright holder: The Financial Times Limited. All rights reserved
- Vietnam: No U. S. Moves to Escalate the War**: Snippet: "...NEWS SUMMARY Vietnam:No U.S. moves to escalate the war American air strikes at missue sites near Hanoi ...y's raids were closer to the N. Vietnam capital than at any time sin...mittee: "I don't think the South Vietnamese will be the first people  
Copyright holder: The Financial Times Limited. All rights reserved Illustration: En Ces Temps Noirs De Jactance Et D'incroyance, Notre Dame De La Fin Des Terres Vigilante—from the Mi
- Amsterdam Police Fire on Rioters: 30 Hurt**: Snippet: "...ation by senior priest Thich Tam Chau. Bombng sorties over North Viet- nam totalled 60 on Monday-an... dropped 12%m. leaflets in South Vietnam, telling North Vietnam troops: Defect or surrende...ion by senior priest Thich TJ- am Chau. Dombng sorties over North Viet- nam totalled 60 on Monday-an  
Author titles: Our Industrial Editor Copyright holder: The Financial Times Limited. All rights reserved
- No Vietnam Bombing Pause, Says U. S. Unless**: Snippet: "...nother pause in its bombing of North Vietnam unless Hanoi ( gives an indi...et for U.S. aircraft attacking North Vietnam-besides 105 barges and junks...venth Fleet destroyers shelled North Vietnam shore batteries near Dong  
Copyright holder: The Financial Times Limited. All rights reserved
- 2—1 against Portugal—England Reach Final**: Snippet: "...rt) advocacy of an invasion of North Vietnam. Fuel dumps hit American aircraft bombed fuel...rtugal's lone goal-a penalty-came in the 83rd minute when Benfica's glamour boy Eusebio beat Banks with a real pile-

Figure 4.6: The search results for the query “American air strikes over North Vietnam”, submitted to the *FT newspaper archive*.

In Figure 4.6 the researcher is interested in news articles on the bombing of North Vietnam by America during the Vietnam War (1955 - 1975). Scanning the news articles with partial matches for the query, reveals that the researcher could consider reformulating the original query to “sorties over North Vietnam”, where the word sorties appears to be more prevalent across multiple documents than the term “air strikes”. Furthermore, the fourth document in the search results shows matches for the query for words that are hyphenated, such as “Viet-nam”, which demonstrates the flexibility of the character-level

$n$ -gram representation. A word-level  $n$ -gram model may not capture this instance without some level of preprocessing.

## King James Bible

The screenshot shows a search interface for the King James Bible. At the top, there is a search bar with the text "Search: in the beginning was the word". Below the search bar, the results are displayed in a list of entries, each with a chapter title and a snippet of text. The search results are as follows:

- Luke chapter 1**: In the beginning was the Word, and the Word was with God, and the Word was God. The same was in...e will of man, but of God. And the Word was made flesh, and dwelt am...he Word was God. The same was in the beginning with God. All things were made
- John chapter 26**: ...l the Jews; Which knew me from the beginning, if they would testify, that ... persecutest thou me? it is hard for thee to kick against the pricks. And I said, Who art thou, Lord? And he said, I am Jesus w... noble Festus; but speak forth the words of truth and soberness. For  
Origin: Textus Receptus; similar to the Byzantine text-type; some readings derived from the Vulgate. Edition: Authorized Version; Cambridge Edition Section: New Testament
- Philemon chapter 1**: ...bove thy fellows. And, Thou, Lord, in the beginning hast laid the foundation of...n, and upholding all things by the word of his power, when he had by
- Proverbs chapter 8**: ...ll fill their treasures. The LORD possessed me in the beginning of his way, before his work...an abomination to my lips. All the words of my mouth are in righteousness; there is nothing froward or perverse in them. They ... set up from everlasting, from the beginning, or ever the earth was. When
- Jeremiah chapter 49**: ...or any son of man dwell in it. The word of the LORD that came to Jeremiah the prophet against Elam in the beginning of the reign of Zedekiah ...ld, the days come, saith the LORD, that I will cause an alarm of war to be heard in Rabbah of the Ammonites...n my fierce anger, saith the LORD; and I will send the sword after them, till I have co
- Jeremiah chapter 28**: ...t came to pass the same year, in the beginning of the reign of Zedekiah king...et Jeremiah went his way. Then the word of the LORD came unto Jeremiah the pro...ich prophesieth of peace, when the word of the prophet shall come to pass, then shall the prophet be known, that the LORD hath truly sent him. Then
- Amos chapter 7**: ...Thus hath the Lord GOD shewed unto me; and, behold, he formed grasshoppers in the beginning of the shooting up of the l...nd is not able to bear all his words. For thus Amos saith, Jeroboam shall die by the sword, and Israel

Figure 4.7: The search results for the query “In the beginning was the word”, submitted to the *King James Bible*.

The query “In the beginning was the word”, represents a set phrase from *Luke, Chapter 1*. Researchers of the Bible may submit long set phrases as queries in order to retrieve a known-item. Digital archives such as the *Aramaic Magic Bowl archive* and the *King James Bible* lend themselves to phrase search due to the repetition of religious themes. The researcher could submit a whole document as a query to identify how prevalent a set phrase is across the archive of documents. Providing tolerant search tools allows researchers to feel that they have gained a comprehensive overview of the research topic and the relevant material.

## British Library Microsoft archive

Microsoft Corpus Search: the journal of the royal geograp...

Memor of the South and East Coasts of Arabia ... Part II. (Extracted from the London Geographical Journal.)  
 Note: , " was published in vol. 9 of the Journal of the Royal Geographical Society.

Madagascar and its People. With a map (taken ... from the Journal of the Royal Geographical Society, vol. 20)

Geographical and statistical memoir of the Konkun. The revenue and land tenures of the western part of India, considered with reference to their first institution and present working. Re-printed from the Journal of the Bombay Geographical Society for May, 1840

social and intellectual state of the Colony of Pennsylvania, prior to the year 1743. Read before the American Philosophical Society.

...  
 NamePart: TYSON, Job Roberts .

An Enquiry after Religion: or a view of the Idolatry, Superstition ... and Hipocrisie of all Churches and Sects ... also some Thoughts of a late ingenious Gentleman of the Royal Society concerning Religion

Extracts from a Journal of Travels in North America, consisting of an account of Boston and its vicinity. By Ali Bey, &c. Translated from the original manuscript or rather written by S. L. Knapp.

A new geographical, historical, and commercial grammar; and present state of the several kingdoms of the world ... The astronomical part by James Ferguson, F.R.S., to which have been added the late discoveries of Dr. Herschel ... Illustrated with a correct set of mans. engraved

Figure 4.8: The search results for the query “the journal of the royal geographical society,” submitted to the *British Library Microsoft archive*.

The *British Library Microsoft archive* represents a very diverse collection of topics and text genres. This archive is supported by a metadata *SLM*, introduced in Chapter 5. Although full-text search is not provided, the metadata records contain lengthy unstructured text that provides enough useful information about the documents, and the topics covered by the archive to make them discoverable. For example, consider a query “Journal of the Royal Geographical Society”. A researcher interested in publications by this journal can locate these documents on the basis of matches in the title or note field of the metadata. Furthermore, the document ranked third in the search results, retrieved through a partial match for the query, suggests a potentially relevant document published by the *Journal of the Bombay Royal Geographical Society*. However, the length of the titles for these documents is exceptional, and so without such a descriptive set of metadata, the document may not have been retrieved. Consequently, metadata plays an important role when the full



text of the documents is not available, which may result from the presence of different media types such as images, or documents represented by poor quality *OCR*.

### Giorgio Vasari archive

The screenshot shows a search interface for the Giorgio Vasari archive. At the top, there is a search bar with the text "Search: madonna and child 1380". Below the search bar, there are several search results displayed in a list format. Each result includes the name of the artist or subject, a snippet of text from the original document, and an image title. The results are as follows:

- Antonio Veneziano**: ... VENEZIANO/ ANTONIO VINIZIANO **Madonna and Child**, c. **1380**, Museum of Fine Arts, Bost...es of the Artists MANY who would rain stay in the country wher  
Image title: **Madonna and Child**, c. **1380**, Museum of Fine Arts, Bost
- Antonio Veniziano**: De BibliothecaLa biblioteca di BabeleBiblioteca TelematicaCLASSICI DELLA LETTERATURA ITALIANATesti senza diritti d'autore LE VITE DE' PIU' ECCELLENTI ARCHITETTI, PITTORI, ET SCULTORI ITALIANI, DA CIMABUE INSINO A' TEMPI NOSTRINell'edizione per i tip  
Image title: **Madonna and Child**, c. **1380**, Museum of Fine Arts, Bost
- Margaritone**: ...VASARI'S LIFE OF MARGARITONE **Madonna and Child**, c. 1270. National Gallery o...ves of the Artists AMONG THE **OLD PAINTERS** who were much alarme  
Image title: **Madonna and Child**, c. 1270. National Gallery of Art, Washington, **D.C.**HIC JACET ILLE BONUS PICT **Current**  
Location: I Gallery of Art, Washington, **D.C**
- Margaritone**: De BibliothecaLa biblioteca di BabeleBiblioteca TelematicaCLASSICI DELLA LETTERATURA ITALIANATesti senza diritti d'autore LE VITE DE' PIU' ECCELLENTI ARCHITETTI, PITTORI, ET SCULTORI ITALIANI, DA CIMABUE INSINO A' TEMPI NOSTRINell'edizione per i tip  
Image title: **Madonna and Child**, c. 1270. National Gallery of Art, Washington, **D.C.**HIC JACET ILLE BONUS PICT **Current**  
Location: I Gallery of Art, Washington, **D.C**
- morto da feltre and andrea di cosimo**: ...LTRINI Morto da Feltre. Virgin **and Child**. Museo Civico, Feltre. LIVES OF MORTO DA FELTRE (14**80** circa-1527) and of ANDREA ...ade, which caused him to be hel**d** in great estimation, found hi  
Translation title: Morto da Feltre and **Andrea di Cosimo**
- giovann francesco caroto and giovanni caroto**: ...rona3 Giovan Francesco Caroto. **Child** with a drawing . Verona, Muse...3: GIOVAN FRANCESCO CAROTO (14**80** circa-1555) and GIOVANNI CAR  
Translation title: Giovan Francesco Caroto and **Giovanni Caroto**
- nicola colli** **tribolo**

Figure 4.9: The search results for the query “Madonna and Child 1380”, submitted to the *Giorgio Vasari archive*.

The *Giorgio Vasari archive* supports the research of art history. To illustrate, the researcher in this example is searching for a particular work of art “Madonna with Child”, relating to the Virgin Mary and Jesus Christ, which was a popular theme for artists, who were influenced by the history, culture, and religious teachings in Italy at the time. Figure 4.9 presents the search results for the English and Italian translations of the documents. Here the researcher is particularly interested in the work of Antonio Veneziano in the year 1380. The metadata search component is useful in this example, as it provides a “bridge” between the two language corpora.

## 4.8 DISCUSSION

This chapter described the search engine underlying the *Samtla* system and discussed the decisions that were taken with regard to the representation, indexing, and ranking of the documents. The probabilistic approach allows relevance to be expressed in terms of the documents with the highest probability of matching the users' query, which is a concept familiar to many researchers across the disciplines. The character-level  $n$ -gram model provides both full and partial query matching and enables the system to cater to any language corpora, with little preprocessing of the text. The *SLM* approach provides the means for ranking documents on the basis of an informational need expressed by the terms of the users query. Tuning the parameters of the language model, reflected by the  $n$ -grams of the model, is achieved through the deleted interpolation method, which approximates the probability for the  $n$ -gram through by interpolating the probabilities from lower order  $n$ -grams using a weighted term to control the contribution from each order of  $n$ -gram, with smaller sequences contributing less to the final probability for the query.

Further smoothing is applied at query time, which combines the probability of the  $n$ -gram inferred from its individual document model  $D$ , with a more stable estimate of its true probability according to the collection model  $C$ . When the system is unable to locate a term of the query for a specific document with a partial match, the backoff process approximates the probability for the missing  $n$ -gram, by using information from lower order  $n$ -grams.

Although more sophisticated methods are available, with respect to different document scoring, and smoothing strategies [197], the implementation described in this chapter provides a lot of flexibility, and a principled approach for extending the basic query model to specialised retrieval tasks, such as the metadata search (Section 4.6), through the application of the *JM* interpolation method and an appropriate weighting scheme. The parameters of the query model, introduced in this chapter, were evaluated empirically with feedback from our research groups, and more formally through an evaluation based on crowdsourcing (discussed in Chapter 7).

## CHAPTER 5

# TEXT MINING

This chapter presents an overview of the *Samtla* mining tools that have been developed in response to a need for flexible digital tools to support the mining of “parallel passages” representing repeated structural text patterns recorded in the documents. Each of the tools are constructed from one or more of the components underlying the infrastructure represented by the *SLM*, stored in a suffix tree data structure, and demonstrate the flexibility of the *SLM* to tasks other than information retrieval.

The chapter begins with Section 5.1, which presents a brief overview of the mining tools developed to address the specific needs of the research groups introduced in Chapter 3. Section 5.2 describes how the metadata is leveraged by the *Samtla* system to filter and browse the documents to support the retrieval of documents across different media, data formats, and languages. Section 5.3 introduces the recommendation tools that generate alternative queries according to the search trends of the research community, and natural language processes, and support for document recommendation by recommending documents on the basis of the browsing behaviour of the research community. In addition, the *Jensen Shannon Divergence (JSD)* is adopted to measure the similarity between the  $n$ -gram probability distributions of document pairs, stored in the document model  $D$  of the infrastructure. This forms the basis for the related document tool, which presents semantically similar documents to the researcher, and acts as an entry point to a document comparison tool, presented in Section 5.4. The document comparison tool was developed specifically for mining of variable length “parallel passages” that are important to researchers. A named entity tool is introduced in Section 5.5, which uses

gazetteers for the names of people, locations, occupations, and commodities to generate additional navigation structures in the browsing tool, and as an information layer over the document text. Lastly, the chapter concludes with a discussion of the research groups in relation to the tools introduced.

## 5.1 OVERVIEW

Books, web pages, articles, and reports are all examples of unstructured text data where relevant information exists potentially anywhere within the document. Unstructured text data is often managed and retrieved via a search engine (see Levene (2010) [132]). Search engines provide the means to retrieve information but not to analyse it, this is where mining techniques are useful, as they provide different views of the data to facilitate the discovery and subsequent analysis of textual patterns [51]. These patterns can then be examined more closely through traditional research techniques such as the close-reading of the text, but generally only for small scale digital archives. Mining tools developed for the purpose of literary analysis of texts have existed since the 1940s, when researchers saw the immediate benefit of using computers to produce concordances of specific text patterns [163,164]. A review of the research and commercial systems provided to humanities researchers reveals that there are a common set of features and tools provided, which are summarised as follows:

- **Browsing.** Document browsing using the file structure of the archive and attributes of the documents such as the section or chapter. This is a feature supported by the majority of systems and tools reviewed.
- **Metadata.** Metadata search and browsing is supported by the *Blake Project*, *Responsa*, and *CULTURA and IBM LanguageWare* systems.
- **Comparison.** “Parallel passages”, are identified and mainly displayed as interlinear text, and tend to represent alternative translations of the text. Example systems include the *Responsa*, *Logos Bible Software*, and *Accordance*, but there is generally no support for comparing “parallel passages” on the basis of their semantic similarity to other sequences recorded in the archive, which also preserves the dependency between the constituents of the sequence.

- **Named Entity tools.** Named entities are often identified and tagged in the document text, provided as search filtering options, or as browsing categories (*CULTURA and IBM Languageware, and Logos Bible Software*).
- **Recommendation.** Recommendation tools are not often included, aside from those supported by the *Responsa*, and *Texcavator*. This appears to be attributable to the scope of the tools, which tend to focus on providing a single specialised function, rather than as part of a comprehensive system.
- **Data analytics.** Many of the tools are developed to support the analysis of word frequency distributions in order to identify patterns in word usage between texts. Examples include *Voyant Tools, CULTURA, and IBM Languageware, and Logos Bible Software*, which use the statistics for generating visualisations such as word clouds.
- **Visualisations.** The most popular approaches adopted for summarising the documents, include the word cloud (supported by *Voyant Tools, Texcavator*), social network graphs (*CULTURA, and IBM Languageware*), and time lines generated from events identified in the document (*Texcavator, Logos Bible Software, and Accordance*).

It is not always clear from the literature what approaches have been adopted for providing the current set of tools, but the majority would appear to be developed from manually tagged data rather than through mining techniques, which is generally due to the fact that the tools operate with small scale archives, and the existence of tagged data for the most popular archives, such as the Bible. The infrastructure, represented by the *SLM* and suffix tree data structure, is applied to the task of information retrieval (introduced in Chapter 4), which has been a natural application of *SLMs*, due to their success in the speech recognition domain [159]. The *SLM* is easily extendable to tasks other than search such as recommendation [128], hand writing recognition [143], and machine translation [67]. In addition, character-level  $n$ -grams have been shown to out perform word-level  $n$ -grams in applications such as plagiarism detection [179], and spam-email filtering [116, 156]. The approach to mining presented in this chapter has permitted the development of several

generic tools for the display, search, and analysis of document content and associated metadata.

## 5.2 METADATA

This section presents the tools developed from the metadata, which reflects information about the properties of a document with regard to the context and circumstances for its creation and use [104]. Metadata is often referred to as “data about data” [56], and can be divided into two distinct categories based on usage, which is summarised as follows:

- **Bibliographic:** Normally represented by unstructured text, and often created by experts familiar with the document collection. A common example is a bibliographic record for an item stored in an archive or library, which forms the basis for matching relevant material to visitors’ requests for information [101].
- **Technical:** Records the technical attributes of the digital item, for example, the file format, file size, and dimensions of the scanned image.

Bibliographical metadata describes specific attributes such as a reference to the original source or author. In addition, bibliographical metadata is also used to provide contextual information on documents in different media formats, e.g. photographic images, sound recordings, and video. Consequently, making bibliographic metadata searchable can facilitate the discovery of materials across different media-types [56]. An example of bibliographical metadata can be found in the *Aramaic Magic Bowl archive*, which contains information on the literary genre of the text, references to previous publications, and extensive notes made by the researchers on the content and historical context of the texts. Furthermore, photographs of the original artefact enable researchers to view the original artefact for comparison.

Technical metadata, on the other hand, describes the technical attributes of the documents. For instance, a collection like the *FT newspaper archive*, contains several levels of technical metadata, one for the newspaper with the publication date, the extent in pages, and the cost for each individual newspaper. And the article level technical metadata, which records the pixel coordinates of the article in the original scan (for cropping or highlighting), the

page number, and the extent of the article text in columns.

In *Samtla*, each value of the metadata is stored in a *SQL* database as a single record indexed by the document *id*. The metadata is then accessed by the *Samtla* system tools through a wrapper function for each digital archive, which passes the raw text data to the tools for processing, or display with the document. In addition, two tools were developed from the metadata, which address a need for metadata search and browsing over large-scale digital archives, such as the *British Library Microsoft archive*, and the *FT newspaper archive* (refer to Chapter 3 for an overview of the research groups). These archives are highly variable with respect to the quality of the *OCR* of the document text, where many are of poor quality, which makes it difficult to retrieve the documents using the full text alone. Consequently, metadata may provide the only source of reliable textual data that can be leveraged for search and mining these collections. Furthermore, not all of the documents in the archive represent textual data, but include images representing maps, plates, paintings, and illustrations. This makes the metadata particularly important, as it provides a basis for researchers to locate documents across a range of different media types that would otherwise not be retrieved through search over the document text.

The query model in Chapter 4, was updated with an additional *SLM* over the metadata record for each document, stored in a separate suffix tree data structure from the documents. The resulting index supports the retrieval of documents according to their attributes, for instance, year of publication, topic, and other data generated by the researchers, such as notes, comments, and references to external sources. The main browsing architecture in *Samtla* is also supported by the metadata, where documents are grouped according to shared attributes stored in the values of the metadata (discussed in Section 5.2.2).

### 5.2.1 Search filter tool

A search filter is constructed from the metadata model *B* using the set of metadata fields extracted from the corresponding suffix tree index. The search filter tool is presented to the researcher at search time, and is constructed from the set of links generated from each metadata field label extracted from the

leaf nodes of the suffix tree for the metadata model  $B$  (discussed in Chapter 4). The links are mapped to a function, which hides documents from the search results that do not have a match for the label of the metadata field selected by the researcher. Search filter tools are useful as they allow researchers to quickly filter and reduce the search results to relevant documents on the basis of information that is separate from the document content, but which describes the topic or context of the document more explicitly. For example, the topic of the articles in the *FT newspaper archive*, is often expressed in the headline of the article, or the name of the section. Consequently, selecting the “title” or “section” label in the search filter, would reduce the list of search results to those documents that contain a match for the query in these fields only, enabling researchers to filter documents from the results very quickly to those with mentions of “war” in the headline text, or only those documents falling under the “Arts and Entertainment” section of the newspaper.



## Case studies

This section introduces several case studies that demonstrate how the metadata search and search result filtering tools support the research of the digital archives. The search filter supports all research groups due to the availability of some form of metadata. The metadata for each archive varies with respect to breadth, quality, language, and media, such as those with image collections (*Aramaic Magic Bowl archive*, *Giorgio Vasari archive*, and *FT newspaper archive*).

### Aramaic Magic Bowl archive

The screenshot shows a search interface for the Aramaic Magic Bowl archive. The search bar at the top contains the text "חחים ומחתם ביתה ואיסקופתה דודי ב". Below the search bar, there are three search results, each with a title and a snippet of text. The first result is "AIT Bowls CAL 23 AIT 21" and the second is "AIT Bowls CAL 25 AIT 23". Both snippets are highlighted in yellow. On the right side of the interface, there is a "Search Filter" dropdown menu with the following options: "All", "Parallels", "Parent", "Notes", "Etymology", "Family tree", "Questions", and "Type". The "Parent" option is currently selected and highlighted in blue.

Figure 5.1: The search filter for a query representing a “parallel passage”, submitted to the *Aramaic Magic Bowl archive*.

The *Aramaic Magic Bowl archive* has some of the richest metadata, which includes the names of clients, references to related texts, research notes documenting the content of the inscriptions, and open research questions. This means that the search results return a lot of useful information aside from matches for the query in the document text. Consequently, the search filter is particularly important for this archive as it provides users with a tool for narrowing the search results to specific features of the documents and metadata, see Figure 5.1. For example, a researcher may wish to focus on specific texts that have a match for the name of the parent in the metadata by selecting the “parent” filter. Alternatively, the “notes” filter reduces the results

to those with a match in the comments made by researchers, which provides an overview of the current thoughts on the cultural context recorded by the document text.

## Financial Times newspaper archive

Financial\_Times Search: vietnam cease-fire

Canadian Initiative for **Vietnam Cease-fire**  
 Canadian initiative for Vietnam cease-fire From Our Foreign Editor OTTAWA, May 5. The Canadian Government is understood to be very actively conducting diplomatic probes designed to bring about a cease-fire in Vietnam through the agency of the I  
 Author titles: Our Architectural Correspondent AllSource: From Our Foreign Editor Copyright holder: ial Times Limited. All rights reserved

S. Vietnam Junta Speeds **Return to Civilian Rule**  
 ... clash UN truce men arranged a **ceasefire** after a four-hour clash imvo... frigate Blackpool is to be **hired** to New Zealand for up to fi...danger" washed up in western **Ireland**. Princess Marina became  
 Category: News in Brief Copyright holder: ial Times Limited. All rights reserved

Meetings with Smith **Regime Should Now Cease—Margal**  
 THELCoWfa4 Ot IMLICs regime said argal 4-1 fl By Our forelgn Staff ha two S ISI uml ofM the l " iatlitbe British Governmeti taV Ofirmer action from now on to bring down the Smith t4ime prerably by the use of force, o1 bj comphlete mandatory s  
 AllSource: By Our Foreign Staff Copyright holder: ial Times Limited. All rights reserved Fixed publication date: September 09, 1966

Gibraltar Car Ferry **Ceases Operation**  
 GIBRALTAR CAR FERRY CEASES OPERATION From Our Own Correspondent GIBRALTAR, Nov. 1. The Southampton-Gibraltar car ferry Sunward, operated by Klosters, called here to-day for the last time. This is a direct result of the Spanish closure of the La Li  
 Article subtitle: Willing's European Press Guide First name: Daré AllSource: From Our Own Correspondent Copyright holder: ial Times Limited. All rights reserved

19 Nations Expected at Lagos **Conference**  
 ...ferA a loern getinel, a cn and**sefire**. Thakist far Moscow has berions **are**luc Nor (Pakid stam Hand Amritsar In. WMotoris **Pre**ident John," form swaitte... coast in an incident between **Elie** and N. **Ireland** foivngher- meord North-of-the-border boatmen claim that **Re**public fishermen **are** sabotaging be sent the...nia, Australia and Trinidad **-are** not expected to attend. In Dar-es-Salaam, **Pre**sident Nyoy**are** said the **confer**ence could only be a " f  
 Copyright holder: ial Times Limited. All rights reserved

£2m. Contracts for Sir Lindsay Parkinson: Hull **Se**wage Works Extension: £600,000 Costain Award  
 10Bu lldl. a Civil Enimeer g £2m. contracts for Sir Lindsay Parkinson: Hull sewage works extension: £600,000 Costain Award By Our Industrial Staff CONTRACTS awarded by Leeds Housing Committee include one worth over £1.9m. to Sir Lindsay Parkinson  
 Article subtitle: Insurance Author titles: Chief Press Officer Illustration: Shaft and Other Internal Surfa**es** is Provided by Tumers Asbestos Cement's Sprayed Limps**et** Asbestos Copyright holder: ial Times Limited. All rights reserved Fixed publication date: December 19, 1966

One-eleven **Fire**ball' Crash Kills 42

Search Filter  
 All  
 Article subtitle  
 First name  
 Title  
 Author titles  
 Illustration  
 AllSource  
 Copyright holder  
 Fixed publication\_date

Figure 5.2: The search filter for the query “American air strikes over North Vietnam”, submitted to the *FT newspaper archive*.

The *FT newspaper archive* contains both technical and bibliographic metadata, which enables researchers to filter the search results according to the “author”, “image captions”, as well as across different levels of the newspaper such as the section heading, and “article title”, and “subtitle”. The most useful filter would be the “article title”, since title text, although short, often tends to reference the topic described by the article. In Figure 5.2, the top document contains a full match for the query, however there are documents further down the search results which explicitly mention the term “Vietnam”, which reveal a number of documents related to the topic of “air strikes”. For example, the document ranked fifth “No Vietnam Bombing Pause” mentions “U.S. aircraft attacks” in the snippet text, but may not have been retrieved without the additional support from the metadata. This would enable the researcher to reformulate the terms of their query to identify further sources that may be relevant.

## King James Bible

The screenshot shows the King James Bible search interface. At the top, there is a search bar with the text "Search: new testament". Below the search bar, there is a "Related queries" section with a dropdown menu showing "new test2ment". The main content area displays search results for "new testament", listing several chapters of the Gospel of John (John chapter 23, John chapter 22, John chapter 21, John chapter 20, and John chapter 27). Each result includes a snippet of text and a "Section: New Testament" label. On the right side, there is a "Search Filter" sidebar with a dropdown menu showing "All", "Origin", and "Section". The "Section" filter is currently selected and highlighted in blue.

Figure 5.3: The search filter for the query “New Testament”, submitted to the *King James Bible*.

The metadata for the *King James Bible* mainly describes the structure of the archive, where the documents are arranged in to sections, books, and chapters. Therefore, the metadata search emulates some aspects of the browsing tool by allowing researchers to locate a document by searching for the name of the section e.g. “New Testament”, which is often a faster method for accessing the documents than the browsing tool, which may require several selections to arrive at the correct document.

## British Library Microsoft archive



The screenshot shows a search interface for the 'Microsoft\_Corpus' with the query 'war office'. The search results are displayed in a list format, with each entry showing a title and a publisher. The publisher field is highlighted in yellow for each entry. On the right side, there is a 'Search Filter' dropdown menu with the following options: All, Publisher (selected), PlaceTerm, Title, Note, Edition, and NamePart.

Title	Publisher
Table of Distances in Russia, Central Asia, and India. Compiled at the Intelligence Branch of the War Office, May 1885, by Lieut. Wisely	War Office
Précis of Information concerning Zululand. With a map. Prepared in the Intelligence Division of the War Office. Corrected to Dec. 1894 by Captain G. Wemyss	Stationery Office
North Sea Pilot	4 pt. Hydrographic Office: London, 1857-63. 8°.
Oilman, first series complete. With poems and letters, containing a variety of satire, and an autobiographical sketch	Oilman Office
Sailing Directions. Channel Islands	Hydrographic Office
Shirley	Service & Paton

Figure 5.4: The search filter for the query “war office”, submitted to the *British Library Microsoft archive*.

As mentioned earlier, the metadata for this archive is integral to the search and discovery of the digital sources. This case study demonstrates that the *Samtla* system is very flexible to the type and availability of metadata, and can provide browsing and tolerant search tools to researchers with nothing more than a collection of metadata. Furthermore, when the quality of the digital object is not reliable, describes digital objects in formats other than text, or in different languages, then the metadata search tools may provide the only means for searching and retrieving the documents. The example, in Figure 5.4 shows a search for the “war office” in the *British Library Microsoft archive*. The search results return matches in several of the metadata fields including the title of the document, and the “publisher”. A researcher interested in prioritising search results reflecting by primary source material actually published by the “war office”, would be able to identify the relevant entries by selecting the “publisher” filter.

## Giorgio Vasari archive

The screenshot shows a search interface for the Giorgio Vasari archive. The search bar contains the query "Search: the last supper". On the right, a "Search Filter" sidebar is visible, with "Image title" selected. The main content area displays search results for the query, including entries for "Stefano and Ugolino", "Stefano", "Ugolino", "Cimabue", and "Gaddo Gaddi". Each entry includes a snippet of text and an "Image title" field.

**Vasari** Search: the last supper

**Stefano and Ugolino**  
 ...A SIENA Ugolino da Siena (di Ner) . The Last Supper . Metropolitan Museum of ... of the Artists STEFANO, PAINTER OF FLORENCE and disciple of Giotto, was so excellent, that he not only surpassed all the others who had labored in the ar... OF STEFANO (1301-1350), PAINTER OF FLORENCE AND OF UGOLINO SANESE UGOLINO DA SIENA ; UGOLINO DI NERIO (active 1317, d. 1339/4  
 Image title: Ugolino da Siena (di Ner) . The Last Supper . Metropolitan Museum of Ar

**Stefano**  
 ...' TEMPI NOSTRINell'edizione per i tipi di LorenzoTorrentino - Firenze 1550 di Giorgio Vasari STEFANO Pittor Fiorentino Era tanta la fama della nuova ... TelematicaCLASSICI DELLA LETTERATURA ITALIANA Testi senza dir...ella nuova pittura, e tanto erano onorati gli artefici di quella per le maraviglie che Giotto fa  
 Image title: Ugolino da Siena (di Ner) . The Last Supper . Metropolitan Museum of Ar

**Ugolino**  
 ...felicissima l'età di Giotto per tutti coloro che dipingevano, perché in quella i popoli, tirat... TelematicaCLASSICI DELLA LETTERATURA ITALIANA Testi senza dir... TEMPI NOSTRINell'edizione per i tipi di LorenzoTorrentino -  
 Image title: Ugolino da Siena (di Ner) . The Last Supper . Metropolitan Museum of Ar

**Cimabue**  
 ...anta Trinita' . 1285-1286. Tempera on panel. Now in the Uffizi Gallery, Florence. CIMABUE (circa...taly had been afflicted and overwhelmed, had not only reduce... importance, had caused an utter lack of the very artists themselves. At th  
 Image title: Santa Trinita' . 1285-1286. Tempera on panel. Now in the Uffizi Gallery, Florence. Current location: Uffizi Gallery, Florence.

**Gaddo Gaddi**  
 wrought after the Greek manner than did Andrea Tafi and the other painters that were before him and this

**Search Filter**  
 All  
 Current location  
 Title  
 Image title  
 Translation title

Figure 5.5: The search filter for the query “the last supper”, submitted to the *Giorgio Vasari archive*.

Figure 5.5 displays the search results for “the last supper”, which returns matches for the documents from both the English translation, which is condensed in to a single chapter entitled “Stefano and Ugolino”, and the original Italian text divided in to two chapters “Stefano” and “Ugolino”. The search filter enables researchers to filter the search results for matches in the image captions, allowing them to locate documents that contain an accompanying image. This particular example together with the *British Library Microsoft archive* demonstrate how metadata can be important for retrieving documents over a digital archive that contains multilingual documents.

### 5.2.2 Browsing tool

Browsing refers to the unstructured way in which a user explores information [56], where users select a series of categories that produce pre-defined groups of information. Browsing activities can include following a chain of links, or switching from one view to another, and is usually a casual, and undirected method for users to explore navigation structures. Browsing tools adopt pre-defined navigation category structures, which groups or clusters the documents according to a particular feature to provide users with an indication of the type and availability of data in the system [180]. Navigation category structures are useful for encouraging users to explore and discover information within a collection [102], and can be defined manually by domain experts, or semi-automatically using clustering algorithms, particularly for large-scale collections [103].

The category structure for the browsing tool in *Samtla* is created semi-automatically by leveraging the metadata and named entities to construct a hierarchical graph structure (see Section 5.5 for more on named entities). The graph structure describes a series of clustered views [69, 85], which define a hierarchical relationship between the categories.

The hierarchical graph structure is constructed from the field names and values stored in the metadata. The root node of the graph represents the top-level category, which is labelled with the name of the digital archive e.g. *King James Bible*, *Giorgio Vasari*, or the *Financial Times*. The process is semi-automatic, due to the fact that we may not wish to use all metadata as categories, since some fields may not be descriptive, or useful for browsing e.g. an *ISBN* number is not the most intuitive category for exploring a document collection. And so a small set of stopwords are defined, which allow the process to ignore non-descriptive fields during the construction.

In order to generate each navigation path, we iterate over the records in the metadata database, and append the document *id* to a list of documents grouped by the same metadata value. To illustrate, consider the metadata record presented in Table 5.1, which represents the document the *Book of Genesis, Chapter 1* in the *King James Bible*:

Several unique paths can be generated from the keys and values of this record, to produce a graph structure that provides the user with a number of intuitive

Key	Value
document <i>id</i>	1
Book	Genesis
Chapter	1
Section	Old Testament

Table 5.1: A minimal metadata record for the *King James Bible*.

ways to locate the document, which are listed below, where [1] represents a leaf node storing the document *id* for document 1:

1. Genesis → Chapter → 1 → [1].
2. Book → Genesis → Chapter → 1 → [1].
3. Section → Old Testament → Genesis → Chapter → 1 → [1].

The browsing tool also generates a snippet from the child node labels, in order to create a summary of the topic described by each category in the navigation structure. The snippets are generated by sorting the child node labels alphabetically (or numerically depending on the data-type), and then concatenating the labels of the first four children, and the last child, through a depth-first traversal of the graph in a post-processing step. For instance, the browsing tool for the *FT newspaper archive* includes a category entitled “Section”, with the snippet *Arts and Entertainment, Births Deaths and Marriages, Business and Finance, Business Appointments, ..., Weather*, which provides a better description of the available topics in the archive, and also allows researchers to filter out irrelevant information very quickly since they have an overview of the type of information available at each selection.

The graph is visualised using a traditional vertical list view and a treemap view [112]. The list view, mimics a traditional file directory system where each row represents a folder of documents grouped according to the node label. The second view is represented by a *treemap*, specifically, an adapted version of the squarified treemap algorithm [70]. The treemap is motivated by the fact that it displays the hierarchical structure more explicitly than the vertical list view, where all information is visible simultaneously. Furthermore, various properties of the documents can be visualised through the use of colour, or scale to reflect that a node in the graph contains more children than others. Whereas, the vertical list view is usually not a good choice for visualisation,

since the researcher may need to scroll, which makes it difficult to obtain a comprehensive overview as some items will be out of view.

One potential issue with the treemap approach is that the hierarchical graph structure may not be balanced, with some nodes containing more children than others. If the dimensions of the available area is too small then a node with a large number of children results in a layout represented by many small elongated rectangles. This issue is resolved by defining a set of rules that partition the metadata in to additional categories. For example, the *FT newspaper archive* contains the daily editions of the newspaper over several years, with the metadata storing the “date of publication” for each document. Generating a path from this metadata value for documents spanning three years results in the user having to search through 1092 subcategories to locate an article on a single day. In this instance, a rule is defined that converts the datestamp in to the format DD-MM-YYYY, which can then be subdivided in to three components day, month, year and represented as the path:

$$year \rightarrow month \rightarrow day \rightarrow (newspaper\ id)$$

Applying the rule generates a navigation path with three high-level categories representing each year of publication, followed by twelve categories for the months of the year, and twenty-eight to thirty-one categories for each day of the month, which reduces the amount of information presented to the user.

Although the resulting paths require the user to make further selections in order to obtain the relevant document as a result of the additional partitioning, the advantage is the reduction in the cognitive load on the user, as the number of irrelevant documents are filtered from the input very quickly. Consequently, the main role of the manually annotated rules, is to enable the system to mitigate against issues associated with cognitive overload. In addition, this also compensates for the situation when the treemap is unable to subdivide the space appropriately.

Constructing the treemap begins with the largest available display area, which represents the root node of the hierarchical graph structure. A breadth-first traversal from the root node, recursively subdivides the display area in to smaller rectangles based on the number of children stored at each node in the graph. The recursion ends when all nodes of the graph have been visited. The resulting dimension of each rectangle in the treemap is therefore proportional



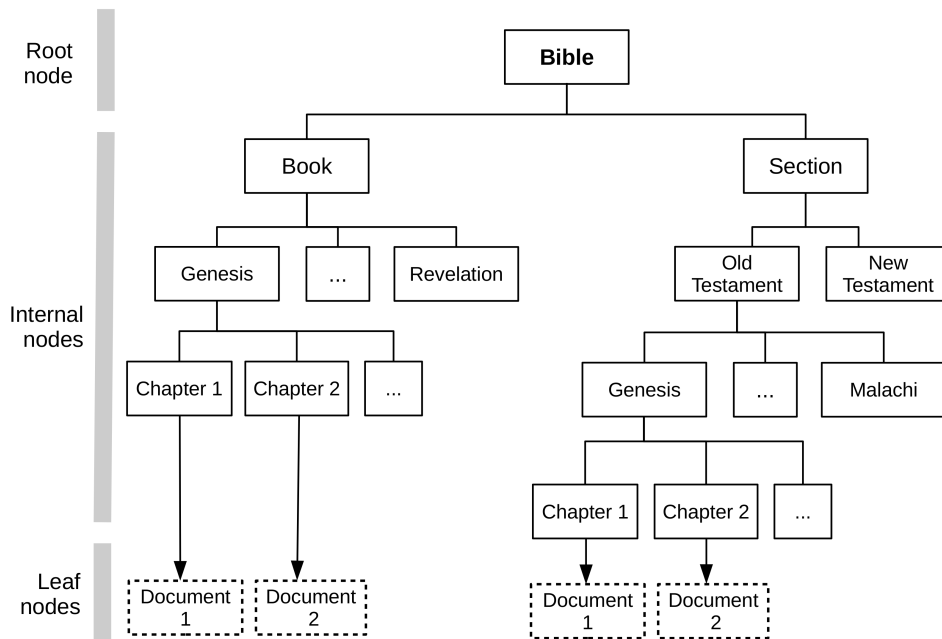


Figure 5.6: A portion of the browsing tree structure for the *King James Bible*.

to the number of children stored under each node. The treemap is generated from the squarified treemap algorithm in Bruls et al. (1999) [70], which scales the aspect ratio of the rectangles in the standard treemap to approximate them better as squares. The scoring function is defined as

$$\text{aspect ratio} = \max\left(\frac{h}{w}, \frac{w}{h}\right) \quad (5.1)$$

where  $w$  represents the width of the rectangle and  $h$  is its height. The aim of the squarified treemap algorithm is to achieve an aspect ratio as close to 1 as possible, which means the height and width are equal, and consequently square. However, it is not always the case that each layout will be optimised, and in the worse case, the results may not be much better than the standard treemap. This is mainly determined by the structure of the graph and how the nodes are distributed. To illustrate the process, assume a subtree with a root node containing 4 child nodes and the following distribution of children for each child node: [4, 2, 1, 1], making a total of eight nodes. The proportion of the area dedicated to each child node of the root is determined by the following distribution of ratios  $[\frac{4}{8}, \frac{2}{8}, \frac{1}{8}, \frac{1}{8}]$ . With this information, the treemap algorithm will attempt to generate a layout that reflects the size of each child, and at the same time approximate the available area for each child node as

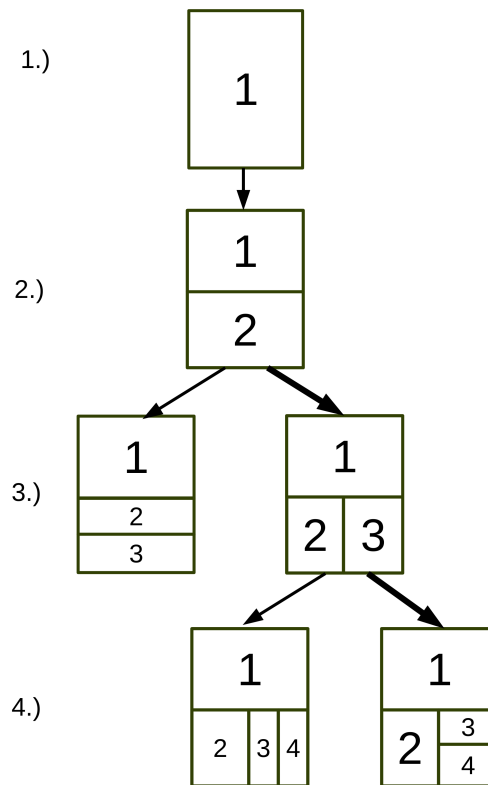


Figure 5.7: The layout process with 4 child nodes. Bold arrows indicate the flow of the construction, which preserves the best aspect ratio.

squares. The steps taken by the algorithm, with reference to Figure 5.7, are as follows:

1. The initial direction of the subdivision is determined by the width and height of the area. The initial area has an area with a greater height to width aspect ratio, and so the division starts along the horizontal axis, with the the initial area divided in to two equally sized rectangles (see step 2 in Figure 5.7). Otherwise, the first subdivision is performed vertically by default.
2. The subdivision continues from the newly created rectangle in the same direction (horizontally).
3. Adding the third child subdivides the available area in to three parts, which results in a poor aspect ratio for each rectangle, where the second and third child become too elongated. Consequently, this proposed layout is discarded and the process switches direction, and makes an attempt to divide the area vertically instead. The resulting rectangles have a higher aspect ratio, and so this layout is selected as the best layout on

which to proceed (as indicated by the bold arrow).

4. The third child is divided in half to provide space for the new rectangle represented by the fourth child. However, because the process is dividing the space vertically, we again end up with elongated rectangles with low aspect ratios. To remedy this, the process changes direction again, and subdivides the available area along the horizontal axis with the fourth child placed in the last rectangle. The resulting layout achieves a higher aspect ratio for each rectangle and is selected as the optimum layout (bold arrow).

The advantage of the squarified treemap is that it utilises the display space more efficiently, which makes navigation easier as thin rectangular areas are avoided, and comparison is made simpler when the aspect ratios are proportional [70].

To summarise, the two views produced by the browsing tool are flexible in terms of how users prefer to navigate the archive, but also the types of data that can be accommodated to make the discovery of documents more intuitive (see Chapter 6).

## Case studies

All archives are supported by the browsing tool, and in this section some examples from the case studies are presented. The browsing tool is unique to each document collection as none of the archives contain the same type or breadth of metadata. However, the browsing tool is data-driven, and so even when there is little or poor quality metadata, the tool can still generate a useful browsing structure, for instance, by generating the navigation paths from information extracted from the document content (see Section 5.5).

### Aramaic Magic Bowl archive

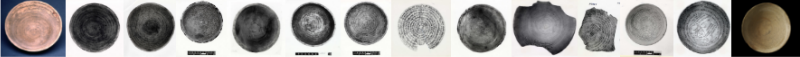
Aramaic Magic Texts	
Category	Subcategory
Category	Amulets, Bowls, Charm, Geniza, ..., VAM
Label	A Bowls, AIT Bowls, AMB Geniza, Amulet, ..., Y Bowls
Language	Jewish Aramaic, Mandaic, Syriac, unknown
Location	Antique dealers Gerhard Hirsch Nachfolger, Berlin Museum, Biblisches Institut Misericorde, Fribourg, British Museum, ..., unknown
Parent	[ ]', missing, unknown, תתאמאטשנודורדוס, ... , אדונשטאדאנס
Photo	
Provenance	Arban Syria, Kermnshh, Khuzistan, Nippur, ..., unknown
Scribe	AIT Bowl 17, AIT Bowl 21; AIT Bowl 22; Müller-Kessler 2005, Bowl 3a; Another parallel bowl in the same hand is Gordon 1941, Bowl 3 (=Müller-Kessler 2005, Bowl 3), AIT Bowl 21; AIT Bowl 23; Müller-Kessler 2005, Bowl 3a; Another parallel bowl in the same hand is Gordon 1941, Bowl 3 (=Müller-Kessler 2005, Bowl 3), AIT Bowl 22; AIT Bowl 23; Müller-Kessler 2005, Bowl 3a; Another parallel bowl in the same hand is Gordon 1941, Bowl 3 (=Müller-Kessler 2005, Bowl 3), ..., unknown
Snklnffid	ן / 9 ן / רן ן / 116 1 unknown

Figure 5.8: The vertical list view representing the landing page of the *Aramaic Magic Bowl* archive.

The browsing tool facilitates the discovery of documents in a number of important ways (see Figure 5.8). Researchers can locate the bowl texts using several different identifiers, for example, cataloguing schemes defined by museums, or the researchers in the literature [149, 177]. Many of the researchers also have specific knowledge and language expertise, and so documents are also categorised according to the language, such as Aramaic, Syriac, and Mandaic, which enables the researchers to locate a subset of the archive that is most relevant to their research. Researchers are also interested in the individuals mentioned in the documents, such as the name of the client and the names of the parents, who are often cited. Understanding why people commissioned these bowl texts, helps to describe the way people lived, their attitudes to illness, and the religious belief systems that existed at the time. Furthermore,

many of the lists compiled by the researchers have been integrated in to the metadata and used to support the browsing of client names, which enables researchers to identify all documents commissioned for a single individual.

The metadata also includes external links to photographs of the original artefact. The Aramaic magic bowls are commonly inscribed with a central image depicting the demon or bad spirit, who is the subject of the incantation or protection spell. Furthermore, the surface of the writing material determined how the text was organised, where on some occasions the text continues on to the outside of the bowl. Consequently, the images for each document are quite distinct, and so browsing by photograph enables researchers to identify the bowls very quickly, and in a more intuitive way, since their research tends to bring them in constant contact with the original artefact.

Researchers are also able to browse according to the provenance of the documents. Some of the documents have an unknown provenance due to many of them being discovered through the antiques market. Organising the documents according to their provenance enables researchers to identify groups of documents that were discovered in the same or similar location. By grouping the documents according to shared features, researchers may be able to identify the provenance of unknown texts by exploring shared client names, and scribes who are often tied to a single location or time period. Lastly, a scribe category, organises the bowls according to their known, or hypothetical author, which provides a simple form of authorship attribution based on the annotations and notes made by the researchers during their close-reading of the text.

## Financial Times newspaper archive

Financial Times	
Category	Subcategory
Commodity	Aluminium, Antimony, Barium, Cadmium, ..., Zirconium
Edition	City
Issue	January, February, March, April, ..., December
Location	City, Country
People	Abdulla, Abdullah, Abel, Abell, ..., Zug
Section	Arts And Entertainment, Births Deaths And Marriages, Business And Finance, Business Appointments, ..., Weather
Source	Bu Our Scientific Correspondent, By A Correspondent, By A Legal Correspondent, By A Member Of Our Commodities Staff Recently In Brazil, ..., The Financial Times Saturday June 11 1966
Supplement	A Financial Times Survey Of Industry In Scotland, Abbotsinch Airport, Advertising, Aerospace, ..., Yugoslavia

Figure 5.9: The vertical list view representing the landing page of the *FT newspaper archive*.

The browsing tool for the *FT newspaper archive*, in Figure 5.9, is constructed from the metadata for each newspaper, and each individual article. Researchers can locate any newspaper and the articles on a particular day, a specific section of the newspaper such as the “Stock Market” or “News In Brief”. The browsing tool is also supplemented with named entities (see Section 5.5), for people, locations, occupations, and commodities, which allows researchers to browse articles related to specific mentions of people, cities, and countries, providing a more natural way to explore the content of news articles than just the structure of the original publication.

## King James Bible

King James Bible	
Category	Subcategory
Book	1 Chronicles, 1 Corinthians, 1 John, 1 Kings, ..., Zephaniah
Location	Abarim, Abdon, Abel, Achaia, ..., Zorah
Occupation	Apostle, Archer, Astrologer, Baker, ..., Writer
People	Female Name, Heber, Hebrew, Male Name, ..., Misc Name
Section	New Testament, Old Testament

Figure 5.10: The vertical list view representing the landing page of the *King James Bible*.

The *King James Bible* contains very little metadata since it represents a single publication. However, the Bible is structured in such a way as to provide a natural set of browsing categories, allowing researchers to browse documents by section, book, and chapter. Like the *FT newspaper archive* the browsing tool is supplemented with additional named entities extracted from the documents. Lists for people and locations are readily available online together with additional encyclopedic knowledge, such as the etymology of the names. However, there are additional lists such as occupation that provide a novel way to explore the content of the Bible. The Bible is about people, locations, and events, and so the browsing tool enables researchers to explore the Bible in a less structured way, than the formal structure of the archive.

## British Library Microsoft archive

Microsoft Corpus	
Category	Subcategory
Genre	Abyssinian Expedition 18671868, Afghan Wars, Agriculture, Anecdotes, ..., Voyages And Travels
Language	English, French, German, Hungarian, ..., Undefined
Location	Aden, Aden Yemen, Afghanistan, Ahmadbd India, ..., Wynaad
Publication Date	1500, 1600, 1700, 1800, ..., 1900
Publisher	1, 2, 3, 4, ..., Z
Topic	A Collection Of Comic Pieces In Verse On Indian Subjects, A Commentary, A Companion To The Handbook Of British India, A Guide To The Stranger The Traveller The Resident And All Who May Have Business With Or Appertaining To India, ..., With The Southern Branches Of The Family Of Darrood Resident On The Banks Of The Webbi Sheb Eyil Commonly Called The River Webbi

Figure 5.11: The vertical list view representing the landing page of the *British Library Microsoft archive*.

The *British Library Microsoft archive* is the most diverse of all the archives that *Samtla* operates with. However, the metadata is quite sparse, but there are a few entries in the metadata composed of long descriptive text that are sufficient to support browsing of the documents. This archive is unique in that the metadata records information about the genre and topic of the texts, which enables researchers to quickly identify documents related to “poetry”, or the “Afghan War”. Furthermore, the named entities provide additional browsing categories for location, people, commodities, and occupations, discussed in Section 5.5.



## Giorgio Vasari archive


Vasari	
Category	Subcategory
Image	
Language	English, Italian
Location	Cities, Regions
People	Artist
Period	1200, 1300
Provenance	Metropolitan Museum of Art New York City., Museo dell'Opera del Duomo, National Gallery of Art Washington D.C, Uffizi Gallery Florence.
Section	Part 1, Part 2, Part 3

Figure 5.12: The vertical list view representing the landing page of the *Giorgio Vasari archive*.

There is no metadata provided by the research groups for the *Giorgio Vasari archive*. The browsing tool is therefore constructed on the basis of the structure of the book, which is divided into parts, and chapters. Each chapter references the life and work of a different artist, together with images of the work cited in the document text. The browsing tool was constructed from very little data, but it still supports researchers of art history in a number of ways.

First, the browsing tool caters to both English and Italian art history researchers due to the presence of both versions of the text. In addition, the researcher is able to locate artists on the basis of their work through image browsing, or by selecting the artist category to locate the document and related work. Furthermore, the lack of metadata is compensated for by the extraction of named entities for location, enabling researchers to specialise in artists from a particular region of Italy.

## 5.3 RECOMMENDATION TOOLS

Studies show that humanities researchers often rely on the research community as an important source of information [71]. When a digital archive is large, it is not possible for users to gain a comprehensive understanding of the content within. This affects their ability to retrieve information through search alone, as they may not have an idea ahead of time, which aspects of the archive will be of interest to them, and consequently, how to describe that interest through a query. Researchers therefore appreciate tools that direct them to relevant sources of information that they would may not have been able locate themselves, through search, or browsing [141]. Furthermore, the topic of research for any one individual researcher may evolve over time, which needs to be accounted for in order to support the current interests of the researcher. Assessing what will be of interest to a particular user consequently requires a model of the users preferences, which is then updated in response to the researcher's current area of interest.

Modeling the online navigation and search preferences of researchers using the *Samtla* system is the responsibility of the *Recommender System (RS)*. A recommender system stores a model of the researchers' search and browsing behaviour, in order to construct a series of recommendations that will guide the researcher to aspects of a large space of possible options that might interest them [109], such as related queries and documents.

Recommender systems are popular components of a system, and many researchers will be familiar with them, for instance, when shopping online we encounter the "what other customers bought" link, and online newspapers often recommend popular and follow up news articles based on the article being viewed. Recommendation tools help users to refine their information need, defined by the terms of their query, which are often short and potentially imprecise [57], and the approach can often help users overcome common spelling mistakes and typographical errors caused by mistyping the query. Recommended items also enable users to expand their search to related concepts that they may not have considered when defining their initial search [53]. In the remaining part of this section, an "item" refers to the objects that are recommended to the researcher [161]. Recommended items are generated from the search and browsing habits of the researchers by aggregating the activity

log data for all users of the system, which then forms the basis for a *user-based* recommendation approach. Related items are often recommended on the basis of the similarity of the researcher's profile to other users of the system, referred to as a *system-based* approach. In the case of *Samtla*, *system-based* recommendation involves identifying semantically related queries and documents using the components of the infrastructure that are supported by the *SLM* and suffix tree data structure.

### User-based recommender systems

User-based recommendation is divided into *community-based*, and *collaborative filtering*:

- *Community-based recommendation*: Users receive personalised recommendations on the basis of their participation in a select group of individuals, who then share content and opinions with other users in the group that they trust.
- *Collaborative filtering*: Assumes that users seeking information, should be able to make use of information that other users have found and evaluated [140, 189], the aim of which is to direct a user to the most “interesting” items of a given collection or domain. A common example is the Google search engine that provides an autocomplete feature [35], which generates related items based on the user's own search history and that of all users.

### System-based recommender systems

System-based, or *knowledge-based* recommendation describes a data-mining technique that make suggestions on the basis of a similarity metric to compare the features stored in the user's profile or the description of the item being recommended. The system then ranks the items according to how well an item matches the users need or preferences [161]. The main issue with system-based recommendation is the “cold-start” problem [140], where the profile for a new user or record of their past purchasing history is empty, and consequently there is no data on which to model the user's preferences. System-based approaches may not identify potentially relevant information due to the fact that modeling user search and browsing preferences can be a complex task as there can be

any number of factors determining how users make choices on which queries and documents are related to their information need.

### Hybrid-recommender systems

Hybrid-recommender systems adopt aspects of both user-based and system-based recommendation to leverage the advantages of both, whilst at the same time reducing the shortcomings of the individual approaches [75]. A set of recommendation tools were developed for the *Samtla* system to construct a *hybrid-recommender* system [161]. The user-based component allows queries and documents to be identified through a form of collaborative filtering based on the activity of all users of the system. User-based recommendation uses the log data of past search and browsing activity to create a personalised history for each user, allowing them to return to previous items they have viewed or searched. Collaborative search is implemented by summarising the log data for all users to generate ranked lists of popular (or “trending”) queries and documents. The ranked lists are produced by assigning a “popularity” score, as a function of the frequency and recency of the query or document given the log data for an individual user or the whole community of users. The user-based approach to recommendation is straight-forward to implement and has the advantage that queries and documents can be easily identified through the log data recording the user activity in the system. More complex methods, such as data-mining techniques, may not be able to identify the most important documents, since there can be a range of interrelated factors that determine how users assess the popularity of a document.

The system-based component identifies items on the basis of the content of the queries and documents, and therefore enables the system to recommend items based on similarity rather than popularity, since what is considered popular may change over time and result in items that previously interested the user becoming undiscoverable once more. Furthermore, the system-based recommendation tool utilises the underlying framework represented by the suffix tree and *SLM* components (discussed in Chapter 4) to generate ranked lists of queries and documents similar in content to the submitted query or document being viewed by the user. The related query tool recommends similar queries by searching for potential permutations of the original query string

in the suffix tree, based on the character rules of the language, and then ranks them according to their probability in the collection model  $C$ . The related document tool creates a ranked list of semantically related documents by comparing the  $n$ -gram probability distributions between document pairs, according to the document model  $D$  for each document. Each component of the hybrid-recommender system is introduced in more detail, starting with a description of the user-based approach to recommendation in Section 5.3.1, and the implementation of the system-based approach is presented in Section 5.3.2.

### 5.3.1 User-based recommendation

The system log files store usage statistics that are gathered when users submit queries to the search engine, or view documents when arriving at the document level during search or browsing. This allows users to return to the point where they left off before signing out of the system, or to record favourite items to which they will often return. Users may also wish to discover what is popular in a collection, as a way to find new documents of potential interest. *Samtla* supports a community feature which suggests search terms and document views based on their popularity, this requires storing data such as unique user *ids*, timestamps, queries and document *ids*. The user data is then used to produce top-ten ranked lists of queries and document views per user and the community of users. The popular queries and documents for individual users and the community are selected and ranked using an algorithm similar to the *Adaptive Replacement Cache (ARC)* [145], where the frequency of each query or document is combined with its recency for the purpose of ranking. The recency of an item is measured by the number of days that have passed since the last record of the query submission or document view. Formally the popularity of a query or document is defined as

$$popularity = T^\beta R^{1-\beta} \quad (5.2)$$

where  $T = \frac{1}{S}$  and  $S$  represents the count in days since the last submission, with *today*=1, and let  $R$  be the raw count of submissions for the query or document. The two terms are combined through a weighted term  $\beta = 0.6$ , with more emphasis placed on the most recent queries and documents. The combination of the two terms  $T$  and  $R$  prevent submissions with high counts,

but longer time between submissions, from dominating the top entries of the recommended queries or documents.

### 5.3.2 System-based recommendation

System-based recommendation is achieved by scoring items on the basis of the content using a distance or similarity-based measure. In *Samtla*, the related queries tool searches the suffix tree storing the *SLM* for the collection model *C* to locate alternative query strings based on a series of predefined traversal methods that simulate common string permutations over character-sequences resulting from the rules of the language recorded in the documents. In other words, the related queries represent “popular” alternatives to the original query according to the morphological rules recorded in the language of the archive. When a set of related queries is located, the collection model *C* of the *SLM*, provides a ranking of the related queries according to their probability in the collection. The system also recommends documents, through the related documents tool, which produces a list of documents that are similar to a document viewed by the user, on the basis of shared-vocabulary represented by *n*-gram probability distributions, stored in the document model *D* of the *SLM*(see Section 5.3.2).

#### Type I related queries

The related queries tool in *Samtla*, addresses many of the language-specific issues, such as differences deriving from the syntax of the language e.g. the encoding of the past tense of the verb or attaching affixes to nouns to describe plurality. Furthermore, the archive may contain documents spanning several time periods, and under this context the system compensates for language change by making a distinction between old forms of the language and their modern day equivalents. To illustrate, the editions of the Bible over the centuries reflect updates to the language as it was recorded at the time of publication. As a result we find alternative spellings, such as in the singulars “Lord” versus “Lorde”, and plural forms “Lords” versus “Lordes”. The language-specific differences are recorded in the suffix tree as a series of unique paths rooted at the subtree for the sequence “Lord”.

The related queries are located by traversing these unique paths through

Position	Deletion	Substitution	Insertion
1	ord	?ord	?lord
2	lrd	l?rd	l?rd
3	lod	lo?d	lo?rd
4	lor	lor?	lor?d
5			lord?

Table 5.2: Related queries generated for the character-sequence “lord”.

permutations on the order and presence of characters given the full-sequence represented by the query. The related queries are generated through an online process at query time, using a method similar to the Levenshtein edit distance algorithm, which describes the minimum number of operations required to convert one string into another [100, 133]. The method adopted here, produces a series of alternative queries through deletion, substitution, and insertion of the characters of the original query. If we let  $q'$  represent the related query, where  $n$  is the length of the original query  $q$ , and  $i \in 1, 2, \dots, n$ , then the related queries are generated through the following string permutation methods, where “?” represents the wild-card character.

1. Deletion:  $q' = c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n$ .
2. Substitution:  $q' = c_1, \dots, c_{i-1}, ?, c_{i+1}, \dots, c_n$ .
3. Insertion:  $q' = c_1, \dots, c_{i-1}, ?, c_i, \dots, c_n$ .

Table 5.2, presents an example of the approach, where the original query is “lord”. Deletion does not require a wild-card character, as such, since we simply remove a character from the string where the wild-card character would appear. The wild-card character is either inserted into the string, or replaces a character of the original query, which is then submitted to the suffix tree to locate a match. As the wild-card character is not indexed by the suffix tree there will be a guaranteed mismatch at that character in the query. When a mismatch occurs, we execute each of the above mentioned functions, which traverse the suffix tree from the node where the mismatch occurred. Insertion and Substitution are achieved by replacing the wild-card character with the node label of each child rooted at the last matched node (its parent node). We then attempt to match the remainder of the query along a unique path to a leaf node. If there is a match for the rest of the string, then we extract the smoothed probability for the related query, according to the probability

stored in the collection model  $C$ .

The extracted queries are then ranked to produce a list of related queries that are most similar to the user's original query, where the query with the highest probability, is considered to be the one that is most "related" to the user's original search. To illustrate, given the sequence "lord?", the plural form of the word "lords", and an archaic form "lord $e$ " are ranked as the top-two related queries, since they represent common permutations according to the morphological rules of medieval and modern English.

The related queries are submitted to the suffix tree at the same time as the original query, and the number of additional searches performed by the approach, represented by  $\#Q$ , can be determined through:

$$\#Q = (m * j) + 1 \quad (5.3)$$

where the length of the original query  $m$ , is multiplied by the number of required operations  $j$ , which in our case is  $j = 3$  (representing deletion, substitution, and insertion). We add one to the result to account for the additional insertion of a character at the end of the query. The example query "lord", in Table 5.2, will therefore generate thirteen additional queries. Despite the additional search for the related queries, the results are returned within a few milliseconds as a result of the suffix tree data structure, which locates any sequence of characters in time linear to the length of the sequence. The approach to related query recommendation presented in this section, identifies potential queries with an edit distance of one [133]. However, the resulting related queries, could also be submitted to the suffix tree to locate permutations with an edit distance greater than one. However, this is left to the researcher, who can simply select successive entries from the related queries component of the user interface (see Chapter 6).

Several examples of the output generated by the *Type I related queries* tool are listed below, divided according to each case study. Each example, includes a description of the type of related query (in brackets), where *syntactic* refers to differences in the grammar of the language, *orthographic* differences include completely different words or an older spelling variant, and *spelling error* refers to an *OCR* or transcription error recorded in the document text. Furthermore, some related queries represent different languages from that specified by the



query, which are identified by the name of the language next to the appropriate related query.

- **Aramaic Magic Bowl archive.**

- ליליחא “female demon” → ליליא “male demon/night” (syntactic), ולייליחא “*and* female demon” (syntactic)
- ולא תילוט “that she may not curse” [115] → ולא ילוט “that he it may not curse” (syntactic), ולא תילוט “that she may not curse” (orthographic) [26].

- **British Library Microsoft archive.**

- India → Indian (syntactic).
- Poets → poems (syntactic), ports (orthographic).
- Russia → Russian (syntactic), Prussia (orthographic).

- **FT newspaper archive.**

- Journalist → Journalists (syntactic), Journalism (syntactic).
- American → Americans (syntactic), Americano (Spanish), American (spelling error).
- Vietnam → Vetnam (spelling error), Vienam (spelling error).

- **King James Bible.**

- Lord → Lords (syntactic), Lods (spelling error), Word (orthographic).
- His → hys (orthographic).
- Jacob → Iacob (orthographic).

- **Giorgio Vasari archive.**

- Andrea → Andreas (orthographic), Andrew (English).
- Sculptor → Sculptors (syntactic), Scultor (Italian).
- Veneziano → veneziani (orthographic), veneziano (English).

### Type II related queries

For some language data it can be a challenge to describe all possible string permutations using this approach alone. This is particularly the case for documents representing historic corpora. A further component of the related query tool enables researchers to define a small set of rules, or character-mappings, that replace certain characters in the query with a corresponding character-sequence. The set of rules describe particular processes in the language, such as differences in morphology, dialect, or spelling involving several characters. For example, in the *Aramaic Magic Bowl archive*, the rules describe a set of phonological processes representing differences between dialects, such as long vowels, for example  $i \rightarrow ii$ , and the mapping of characters from one writing system to another e.g. Aramaic script to the Syriac script.

The *King James Bible*, on the other hand, contains a selection of rules used to account for differences in spelling resulting from language change over time. The language of the Bible recorded in older forms of the English language compared to modern English can be quite different. For instance, consider the equivalence between the medieval form “saith” and the modern day form “say”. These are difficult to identify using substitution rules alone, but are easily described by the rule  $y \rightarrow ith$ . A further example from the *British Library Microsoft archive*, compensates for English texts from the 15th century, where the suffix “-ynge” has since been replaced by the modern suffix “-ing” in words such as “accordynge”  $\rightarrow$  “according”. Table 5.3, summarised a number of the rules constructed for the *King James Bible*.

Modern English	Old English	Examples
y	ith	<i>say</i> $\rightarrow$ <i>saith</i>
v	th, st	<i>have</i> $\rightarrow$ <i>hathe</i> .
WORD FINAL	th, yst	<i>mean</i> $\rightarrow$ <i>meanyst</i> .

Table 5.3: *Type II related queries*: a small set of character rules for the *King James Bible* that associate characters of the query with old spelling variants extracted from the *Tyndale* and *Wycliffe* Bibles.

The related queries are generated by replacing the characters of the original query with the output of any associated rule. The resulting related queries are then submitted to the suffix tree and the probability for each full match is extracted and appended to the list of related queries, which is returned to the *view* component of the system architecture for rendering in the user interface

at search time.

The combination of the character-based suffix tree data structure with the *SLM*, provides a good basis for constructing a query recommendation system. The suffix tree stores the dependencies between the characters, and the *SLM* produces a ranking of the related queries according to the smoothed  $n$ -gram probabilities stored in the collection model  $C$ .

The *Type I related queries* capture the processes of deletion, substitution, and insertion, which account for the majority of string permutations that are likely to be found in natural language data. On the other hand, the *Type II related queries* allow researchers to supplement the related queries tool to account for complex character mappings.

### Related documents tool

Document recommendation is the process of recommending documents to the user that discuss the same or similar topic to a *target* document. The target document is the document the user selected, either from a list of search results or through browsing, which they have identified as meeting their informational need. The task of a document recommendation, is then to identify a list of documents that are most similar to the target document. The comparison of documents requires that the documents are reduced to a common representation that can be measured in order to assess the degree of similarity between pairs of documents. Documents can be represented by feature vectors describing information about the documents such as URL, date of publication, language, topic, or author, or the content can be extracted in the form of  $n$ -grams that describe the semantics of the documents.

The role of the related documents tool is to provide a link to a document comparison tool, which enables the researcher to visualise the similarity between the two documents by comparing small and large “parallel passages” shared between the two documents (see Section 5.4). *Samtla* identifies related documents by measuring the similarity between the character-level  $n$ -gram probability distributions of the documents stored in the document model  $D$ . The size of  $n$  is fixed, where a small setting for  $n$  corresponds to a finer-grained document similarity measure. This results in a large set of related documents due to the presence of many short character-sequences representing the com-

mon terms of the language. Correspondingly, a higher setting for  $n$  reduces the set of similar documents to those that share long verbose sequences, representing a coarser-grained analysis. A range of settings were tested, and the 7-gram model ( $n=7$ ) was found to provide a good balance between small and large shared-sequences, based on the 15-gram language model defined in Chapter 4. This was also supported by the feedback provided by the researchers (introduced in Chapter 3), who empiracally assessed the output of the tool.

The similarity between probability distributions is computed through the *Jensen-Shannon Divergence (JSD)*, which is the symmetric version of the well-known *Kullback-Liebler Divergence (KLD)* [91,136]. Each document model  $M_d$  is extracted from the *SLM*, and compared to the probability distribution of every document in the digital archive.

The *KLD* is computed between  $n$ -gram probability distributions,  $P$  and  $Q$ , for *document*<sub>1</sub> and *document*<sub>2</sub>, respectively. The *KLD* is applied to two probability distributions,  $P$  and  $Q$ , which represent a probability distribution based on the 7-grams extracted from the document model  $D$  for each document, and is defined as follows

$$D_{KL}(P||Q) = \sum_i P(i) \log_2 \frac{P(i)}{Q(i)} ,$$

where  $i$  is a probability for a 7-gram drawn from the respective smoothed 7-gram distribution for the document. The *KLD* between the distributions  $P$  and  $Q$ , is obtained by summing the result of the probability for the 7-gram given  $P$  multiplied by the  $\log_2$  of the division between the 7-gram probability in *document*<sub>1</sub>, or  $P(i)$ , and the 7-gram in *document*<sub>2</sub>, represented by the term  $Q(i)$ . The *JSD* is derived from the *KLD*, as follows:

$$JSD(P||Q) = 1 - \sqrt{\frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M)} ,$$

where  $M$  is the average of the two distributions  $P$  and  $Q$ , which is defined as  $\frac{1}{2}(P + Q)$  [91]. After computing the *KLD* between  $P$  and  $Q$  with the average, the *JSD* is then one minus the square root of the interpolation between the two distributions which are interpolated with a weight term corresponding to a 50% contribution from each *KLD* score for  $P$  and  $Q$ . The resulting *JSD* returns a value between 0 and 1, where a score of 1 means the documents are

identical. For each document, the *JSD* scores are ordered in descending-order according to their similarity to the target document so that the most similar documents are ranked at the top of the related document result list.

### 5.3.3 Document prior

This section introduces a further use of the *JSD* as a document prior to update the *Samtla* query model with information about the semantics of the documents. The document prior is a useful component of the query model for favouring documents with certain attributes when ranking [195]. It is not always obvious, based on the query supplied by the researcher, which documents are likely to be more interesting, or relevant to the research topic. Several document priors have been proposed in the past, including those based on the document length in order to bias the ranked search results to shorter, or longer, documents [126], which is often adopted as longer documents provide better coverage of the topic expressed by the query [142]. Another approach is the PageRank algorithm [68], which measures the authoritiveness of each document according to the number of inbound links. Other researchers have found that *URL* length provides a good prior for specific retrieval tasks, such as identifying the home page of an organisations website [124]. The adoption of a document similarity measure as a prior has been shown to be effective [194]. In principle, the prior should describe the importance of each document with respect to how well it describes the range of topics covered by the documents in the archive.

The original formulation of the query model, in Chapter 4, assumed a uniform prior for all documents. However, it is possible to compute a non-uniform prior based on the *JSD* matrix generated for the related documents tool. The prior for the documents is generated from the *dominant eigenvector*, which is a measure of eigenvector centrality [64], corresponding to the eigenvalue with the largest magnitude given a  $n \times n$  symmetric matrix  $A$ . In *Samtla*, the matrix stores the *JSD* score for each document pair, with the diagonal of the matrix populated by the value 1, representing the comparison of a document with itself.

Given  $A$ , the task of identifying the *dominant eigenvector* involves extracting a vector represented by  $x$  that when used as a scalar to the original matrix

$Ax$ , creates an additional matrix  $\lambda$  that represents a scalar multiple of the form  $Ax$ . In other words, we have  $Ax = \lambda x$ , where  $\lambda$  is an eigenvalue, and  $x$  the corresponding eigenvector for  $A$ . To calculate the *dominant eigenvector*, let  $\lambda$  represent an eigenvalue of matrix  $A$  with size  $n$ , then we have the eigenvalues  $\lambda_1, \dots, \lambda_n$ , computed through the following combination of terms [54]:

$$\lambda^n + C_n - 1\lambda^{n-1} + C_{n-2}\lambda^{n-2} + \dots + C_0 = 0 \quad (5.4)$$

The resulting formulation represents a polynomial equation, which means that as the number of dimensions of the matrix increases, so too does the number of terms in (5.4), which can become computationally expensive to compute. A common approach to overcome this issue is to adopt an iterative algorithm, such as the power method [168], which is more efficient, as it only calculates the *dominant eigenvalue* for the matrix, rather than all of the eigenvalues for the matrix [54]. The approach is defined as follows

$$I^{n+1} = AI^n \quad (5.5)$$

where  $I^0$  is the unitary vector and  $J = I^\infty$  is the *dominant eigenvalue* for  $A$ . The dominant eigenvalue is obtained through

$$|\lambda_1| > |\lambda_2|, \dots, |\lambda_n|, \quad (5.6)$$

that is, we perform a search for the eigenvalue with the largest absolute value compared to all possible eigenvalues for  $A$ , and extract the corresponding eigenvector. The prior for each document is represented by a value drawn from the resulting *dominant eigenvector*. The resulting value of the prior for each document is then used for the purpose of ranking as part of the query model, defined in Chapter 4. The influence of the document prior on the ranking of the documents in the query model is incorporated through a weighted term  $\alpha$ , which controls the amount of interpolation between the probability inferred from the document model  $D$  for the query, and the value of the prior given the document, which is formulated as follows, where  $\alpha$  lies in the range  $0 \leq \alpha \leq 1$ .

$$P(M|q) \propto P_M(q)^{1-\alpha} \cdot J_\lambda(P||Q)^\alpha \quad (5.7)$$

The most appropriate value for  $\alpha$  is to be determined as part of future work, through experimentation and feedback from our research groups. The document prior in *Samtla* is novel with respect to other document priors based on document level features such as *URL*, content length. The motivation behind the prior is to generate a ranking that provides a better approximation to the researchers information needs, by supplementing the query model with information about the topicality of the documents, with documents containing a larger number of topics being ranked higher in the search results. Table 5.4 displays the top-ten documents ranked according to the value of their prior. The top document “Isaiah, Chapter 52”, is regarded as the document that summarises, the topics contained in the archive. The *Book of Isaiah* describes

Rank	Title	Prior
1	Isaiah chapter 52	0.10544
2	1 Kings chapter 9	0.08651
3	1 Kings chapter 14	0.08519
4	Jeremiah chapter 25	0.08336
5	1 Chronicles chapter 6	0.07770
6	1 Isaiah chapter 31	0.07674
7	2 Ezekiel chapter 21	0.07660
8	Ezekiel chapter 41	0.07652
9	Exodus chapter 6	0.07646
10	Leviticus chapter 11	0.07481

Table 5.4: The top-ten documents ranked by the value of their prior.

the return of the people of Israel to Jerusalem, who were previously exiled in Babylon, and reaffirms many of the religious doctrines introduced in earlier books of the Bible [88]. The document content contains many of the set phrases common to the Bible, such as “Lord God of Israel”, and “The Lord saith”, as well as a broad range of named entities representing people, locations, and occupations that are often cited by the other chapters of the Bible.

## 5.4 DOCUMENT COMPARISON TOOL

Document comparison is the task of comparing the difference or similarity between the content of two or more documents through analysis of shared vocabulary or features. Document comparison tools are widely available, such as the *Diff Doc* tool [25], and the *compare documents side by side* tool in Microsoft Word 2010 [22]. However, the focus of these tools is on locating the differences between pairs of documents. In contrast, the document comparison tool developed for *Samtla* identifies text regions of similarity, between documents that could be widely divergent overall. Divergence defines the permitted tolerance between two sequences before they are no longer classed as being similar, or identical. The underlying algorithm for identifying shared text patterns is a tailored variant of the *Basic Local Alignment Search Tool* (*BLAST*) algorithm, commonly used in bioinformatics for comparing *DNA* sequences [138]. The method uses a *local sequence alignment* approach that identifies a series of short sequences called seeds, that are common to both documents. These start seeds are expanded a character at a time simultaneously in both documents to produce a larger sequence, up to a predefined threshold. More precisely, the seeds are composed of the unique set of 3-gram character strings shared between two documents, one representing the “target” document, and the other a document drawn from the list of related documents (in Section 5.3.2). The 3-grams are expanded one character at a time, first from the left, and then from the right, through an iterative extension process. Each pair of (approximately) matched sequences is then scored according to their *Levenshtein edit distance* up to a predefined limit [100,133]. Given an expanded seed  $s_1$  from *document*<sub>1</sub>, and  $s_2$  in *document*<sub>2</sub>, the measure for scoring each sequence is defined as follows:

$$ed(s_1, s_2) \leq \lfloor m\delta \rfloor \quad (5.8)$$

where the term  $\lfloor m\delta \rfloor$ , on the right-hand side, defines the threshold determining the limit of the extension process. The limit is met when the edit distance is greater than the floor of the length of the shorter sequence  $m$ , multiplied by a tunable tolerance parameter  $\delta$ . The default setting for the tolerance is  $\delta = 0.2$ , which allows the two sequences to differ by as much as 20%, before the



extension stops and moves on to the next seed. The output is represented as a list of start and end positions identified by a unique seed  $id$ . The algorithm for scoring and extending the initial seeds, represented by 3-gram character-sequences, is as follows:

---

**Algorithm 1** The seed extension algorithm.

---

```

1: procedure SEED-EXTENSION
2:   Retrieve all shared 3-gram sequences  $s_1 \in D_1$  and  $s_2 \in D_2$  with index
   of start positions.
3:   for each seed  $s_1$  and  $s_2$  do
4:      $m = |s_1|$ 
5:      $ed = \text{editdistance}(s_1, s_2)$ 
6:     while  $ed \leq \lfloor m\delta \rfloor$  do
7:        $start -= 1$ 
8:        $s_1 = s_1(start, end)$ 
9:        $s_2 = s_2(start, end)$ 
10:       $m = \max(|s_1|, |s_2|)$ 
11:       $ed = \text{editdistance}(s_1, s_2)$ 
12:       $end += 1$ 
13:       $s_1 = s_1(start, end)$ 
14:       $s_2 = s_2(start, end)$ 
15:       $m = \max(|s_1|, |s_2|)$ 
16:       $ed = \text{editdistance}(s_1, s_2)$ 
17:    end while
18:    store the resulting sequence for  $s_1$  and  $s_2$ 
19:  end for
20: end procedure

```

---

An example of the output generated by the algorithm is illustrated in Table 5.5, which presents the largest sequence extracted from the *Book of Genesis, Chapter 10* for two English Bibles; the first is the *Douay-Rheims Bible*, published 1609 and based on a translation of the Latin original, and the second is the *King James Bible*, published in 1611 and based on the Hebrew and Aramaic original text. The initial starting seed for the shared-sequence is the 3-gram string “ham”:

Douay-Rheims Bible (1609)	King James Bible (1611)
Noe: Sem, <b>Cham</b> , and Japheth	Noah; Shem, <b>Ham</b> , and Japheth

Table 5.5: An example shared-sequence between the *Douay-Rheims* and *King James* bible, which were written in different forms of the English language.

The edit distance between these two sequences is broken down as follows:

- The strings *Noe* and *Noah* have an edit distance of two, since one sub-

stitution ( $a \rightarrow e$ ) and one insertion (final character  $h$ ) is required to translate the strings.

- the conversion of the string *Sem* to *Shem*, requires an insertion of character  $h$ , equal to an edit distance of one.
- the sequence *Cham* is converted to *Ham* with the deletion of character  $C$  at the beginning of the string for a total edit distance of four.

Despite the differences in the spelling of the names due to the different time periods, and changes to the language over the centuries, the two sequences might be considered semantically equivalent to a researcher of Bible scripture. A further example, illustrated in Table 5.6, shows how the approach reveals the lexical choices made by the authors of the different books within the same Bible, where the word “say” in the first example *Samuel 2, Chapter 24*, has been substituted for the word “tell” in *Chronicles 1, Chapter 21*: The differ-

Samuel 2 chapter 24 (1611)	Chronicles 1, chapter 21 (1611)
<p>...the LORD <u>came</u> unto the prophet Gad, David’s seer, saying, Go and <u>say</u> unto David, Thus saith the LORD, I offer thee three things; choose thee one of them, that I may do it unto thee. So Gad came to David, and <u>told</u> him...</p>	<p>...the LORD <u>spake</u> unto Gad, David’s seer, saying, Go and <u>tell</u> unto David, Thus saith the LORD, I offer thee three things; choose thee one of them, that I may do it unto thee. So Gad came to David, and <u>said</u> unto him, ...</p>

Table 5.6: An example “parallel passage” shared between two chapters of the *King James Bible*, which demonstrates the flexibility of the approach to word choice made by two authors.

ence between these two passages may be of potential interest to researchers as they reveal language preferences made by the writer or translator of the original text. As the examples demonstrate, the tailored variant of the *BLAST* algorithm captures text patterns that can be quite divergent overall. Furthermore, the approach is simple to implement, and flexible to the language due to the character-level representation for the  $n$ -gram seeds of the documents. In addition, greater flexibility can be achieved through setting the tolerance parameter to address the morphological complexity of the language recorded in the documents. For instance, a language that is morphologically complex, such as Aramaic, requires a higher setting  $\delta \leq 2$ , whereas languages like mod-

ern English, which has less complex morphological rules in comparison, would require a smaller setting, such as  $\delta \leq 1$ . In general, a tolerance of  $\delta = 2$ , has been found to perform well for the digital archives with which *Samtla* currently operates.

## Case studies

There are currently three archives supported by the document comparison tool, including the *Aramaic Magic Bowl archive*, the *King James Bible*, and the *Giorgio Vasari archive*. This section presents an example of the document comparison tool to illustrate how it is used by the research groups to identify “parallel passages” that help the researchers explore the similarities between the content of the documents.

### Aramaic Magic Bowl archive

The screenshot displays the document comparison tool interface. At the top, it shows 'Aramaic Magic Texts' and 'AIT Bowls CAL 19 AIT 17'. The main area is divided into two panels: 'Hilprecht Bowls 11a' on the left and 'AIT Bowls CAL 19 AIT 17' on the right. Both panels contain a list of 13 numbered items, with corresponding text segments highlighted in blue. The left panel's text is in Hebrew, and the right panel's text is in Aramaic. A 'Related' sidebar on the right lists various other bowl texts, including Hilprecht Bowls 11a, AIT Bowls CAL 9 AIT 8, Hilprecht Bowls 11c, Hilprecht Bowls 11b, Hilprecht Bowls 11, Sad Bowls SADBowl 24, M Bowls M110, Gordon Bowls Cal 60 Gordon K, Gordon Bowls Cal 44 Gordon 2, Gordon Bowls Cal 54 Gordon B, A Bowls 207A BM 117824, VA 2495, VA Bowls VA2484, Non-Categorized Cal 117, and Hilprecht Bowls 11d. At the bottom, there are controls for 'longest sequence' and 'shortest sequence' with radio buttons.

Figure 5.13: A document comparison between two “parallel passages” in the *Aramaic Magic Bowl archive*.

The comparison in Figure 5.13, shows several repeated structural text patterns shared between two Aramaic bowl texts. These sequences represent magic or religious “formulae” that differ due to a number of reasons. For instance, differences arise due to the grammar of the language, such as gender marking on the nouns due to the text being commissioned on behalf of a male versus a female client. Furthermore, the scribe’s dialect is often reflected in the choice of vowels, and they transcribed the “formulae” sometimes in full, or only in part due to a number of reasons, including space restrictions imposed by the medium (the surface of a ceramic bowl). One topic of research looks at the placement of these sequences, which seem as if they have been extracted from a hypothetical recipe book, and used as a basis to form a new text. The small

horizontal map at the top of each document displays the position of these sequences, which reveals where the “formulae” tend to occur, e.g. as part of the introduction, main body, or conclusion of the text (see Chapter 6 for an example).

## King James Bible

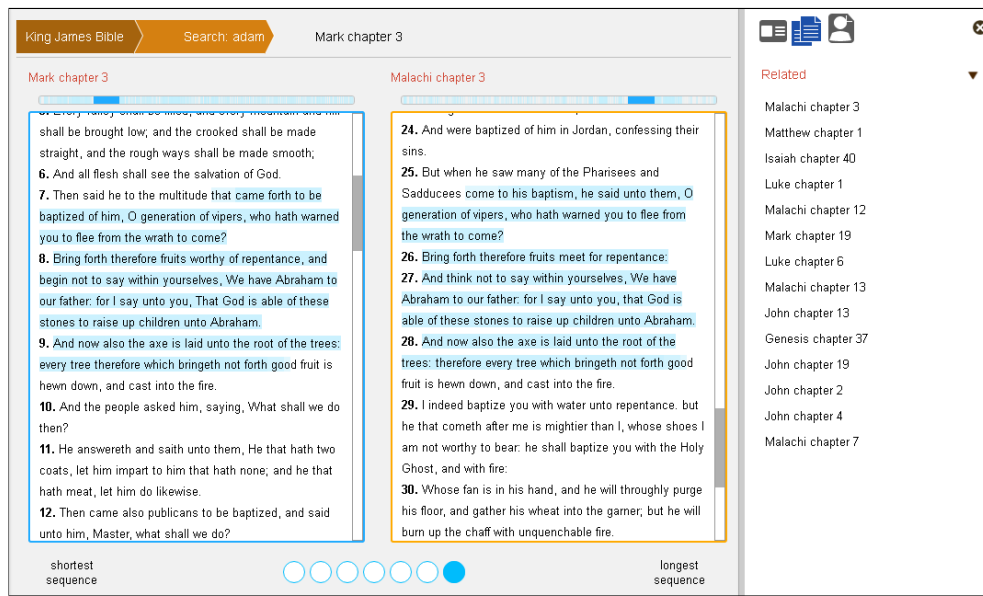


Figure 5.14: A document comparison between two “parallel passages” in the *King James Bible*.

Figure 5.14, shows a comparison between *Mark, Chapter 3* and *Malachi, Chapter 3* of the *King James Bible*, with the largest shared-sequence highlighted. Despite the similarity between the two sequences, there are some key differences between the two shared-sequences, summarised in Table 5.7.

Mark, Chapter 3	Malachi, Chapter 3
<p>that came forth to be baptized of <u>him</u>            Bring forth therefore fruits <u>worthy of repentance</u>,            and <u>begin</u> not to say within yourselves  <u>every tree therefore</u> which bringeth not forth good</p>	<p>come to his baptism, he said unto <u>them</u>            Bring forth therefore fruits <u>meet for repentance</u>:            And <u>think</u> not to say within yourselves,  <u>therefore every tree</u> which bringeth not forth good</p>

Table 5.7: The longest sequence shared between two chapters of the Bible, with the differences between the two sequences underlined.

The example illustrates how the authors of the Bible often quote or paraphrase the writings of previous authors, with slight differences in grammar and word choice. Examples range from a single word e.g. “begin” versus “think”, representing a different lexical choice, to a complete clause e.g. “every tree therefore” versus “therefore every tree”, and “that came forth to be baptized of him” versus “come to his baptism, he said unto them”, which differ in terms of lexical choice, clause structure, and tense.

### Giorgio Vasari archive

Figure 5.15: A document comparison between two “parallel passages” in the *Giorgio Vasari archive*.

The *Giorgio Vasari archive* is the work of a single author, and consequently the shared-sequences identified by the document comparison tool tend to be relatively short compared to the other examples presented in this section. Figure 5.15 illustrates a comparison between two chapters discussing the work of two different architects.

Cimabue	Arnolfo di Lapo (di Cambio)
he gave the form of a T, making it five times as long as it <u>was</u> wide.	he gave the form of a T, making it five times as long as it <u>is</u> broad

Table 5.8: The longest sequence shared between two chapters from the work of *Giorgio Vasari*, with the differences between the two sequences underlined.

Although short, the largest shared-sequence reveals that the two architects may have adopted similar approaches to the design of architecture. The role of the document comparison tool under the context of this archive, is to help researchers identify similarities between the lives, work, and styles of the different architects and artists at the time.

## 5.5 NAMED ENTITY TOOL

*Named Entity Recognition (NER)* [63, 152] describes the process of extracting words (or sequences of characters, in our case) that represent names of people, companies, and locations. *Samtla* adopts the gazetteer approach to extract named entities from the raw documents. Gazetteers have been used for some time as the data to improve the performance of named entity systems, other more sophisticated methods exist, such as semi-supervised learning techniques including the bootstrap and co-training approaches [51], however gazetteers are becoming popular again due to their simplicity, and the recent increase in structured data recording named entities, such as the lists and database records compiled by *Wikipedia* [12–14], and *DBpedia* [118], respectively.

A further motivation for the adoption of the gazetteer approach is that *Samtla* supports a number of historic text collections, such as the Bible and Vasari’s “The lives of the most excellent artists and architects”, which represent closed corpora that are rarely going to be expanded with new documents. Consequently, gazetteers are sufficient for these types of static and domain specific corpora due to the wealth of existing lists compiled by researchers that can be used to form the basis for a gazetteer approach.

The named entity tool in *Samtla* extracts named entities from the documents by submitting each entry in the gazetteer to the *SLM* suffix tree as a query. Each full match is then stored in a database according to the entity type, with the document *id*, and an index of start positions in the document text.

When the researcher views the named entities for a document, the indexes are rendered as an additional layer over the document text in document view (further discussed in Chapter 6). The named entity data is also parsed to the browsing tool, in a similar way to the metadata (see Section 5.2). This enables researchers to browse the digital archive according to a specific person,

location, or occupation, which provides a more intuitive way for locating documents, since the document content often makes reference to a set of named entities in relation to a specific event, particularly in archives like the *FT newspaper archive*.

In practice, capturing certain named entity types can be a challenge. For instance, person entities may be referenced in full only in the introductory text of the document. Subsequent mentions may be restricted to firstname or surname, depending on the level of formality. For example, it is quite common to find the names of politicians referenced in full (e.g. David Cameron) at the beginning of a news article, with subsequent mentions referencing the surname only (e.g. Cameron). On the other hand, entertainers and actors are often referenced by their first name (e.g. Elvis). Therefore, the best strategy for extracting named entities for person, is to store person names as two separate lists, one for firstnames, and another for the surnames. A set of heuristics determines when a person entity is identified, over other entity types, by conditioning on the fact that both first name and surname must be present in the document. The requirement for conditioning on both parts of the name, is due to the need to reconstruct the two parts of the name in order to link additional metadata about the individuals, especially when the entity represents a famous or well-known person. Dividing the names of person entities in to two lists is also more efficient for search and storage, and provides more coverage than a single list of fullnames of individuals, the majority of which may not be referenced in the archive at all.

Named entities for location are divided in to two separate lists, one for countries and the other for city names. After we have obtained all matching locations in the documents, a second phase extracts a set of geo-location coordinates from the *Open Street Map API* [16] for rendering the location on a Google map when the researcher hovers over the named entity (see Chapter 6). The approach is simple and easily deployable, however, the gazetteers could also be used as training data for a statistical learning approach [93].



## Case studies

This subsection presents several case studies demonstrating the application of the named entity tool constructed from the named entities extracted from the document content. The named entities are inserted in to the hierarchical graph storing the navigation structure of the archive, which supplements the browsing tool, allowing users to identify documents through the named entities for people, locations (cities, regions, and countries), occupations and commodities for some archives.

The named entities support the close-reading of the document text, by allowing researchers to select specific named entities, or entity types, which are highlighted in the text. Each named entity is colour-coded according to entity type in order to facilitate identification, and filtering.

## Financial Times newspaper archive

The screenshot shows the 'Financial Times' newspaper archive interface. At the top, there's a navigation bar with 'Financial Times', 'Location', 'City', and 'Southampton'. Below this, a zoom slider is set to 22%. The main content area displays a newspaper clipping titled 'BUILDING & CIVIL ENGINEERING' with a sub-headline '£1.5M. ZAMBIAN CONTRACT: £1M. WILSON LOVATT AWARD: LORDS DEVELOPMENT'. The text is annotated with yellow boxes highlighting named entities. On the right, a sidebar shows a list of named entities categorized by type: People (Grace Gates, John Laing, Wilson Lovatt), City (London, Lusaka, Southampton, Swindon, Zambia), and Country (Zambia).

Figure 5.16: The *FT* newspaper archive named entity view for the original image of the document.

The screenshot displays the Named Entity Tool interface. On the left, a snippet from the *Financial Times* newspaper is shown, featuring the headline "GETTING AROUND THE FRENCH OBSTACLE AND INTO EUROPE" by Joe Rogaly. The text discusses travel routes and motorways. A small inset map titled "THE MOTORWAYS" shows a network of roads. On the right, a larger Google map of Europe is displayed, with red dots indicating the locations of cities mentioned in the text, such as London, Paris, Brussels, Amsterdam, and others. A sidebar on the right lists "Named Entities" with two categories: "City" and "Country". The "City" category is currently selected, showing a list of cities including Amsterdam, Antwerp, Bologna, Brighton, Brussels, Coventry, Florence, Manchester, Milan, Paris, Rotterdam, Southampton, Trieste, Turin, and Venice. The "Country" category is also visible, listing Belgium, France, Germany, Italy, Portugal, Spain, and Sweden.

Figure 5.17: The named entity view for the *Financial Times newspaper archive*, with a Google map of the locations mentioned in the document.

The *FT newspaper archive* provides several categories of named entity based on gazetteers constructed from manually compiled lists across a range of topics [14]. The *FT newspaper archive* is unique in that it is composed of both text and images of the original document. The example in Figure 5.16, illustrates how the named entities are highlighted in the image of the document. The document content is stored in *XML* format, and each word of the text stores attributes for the pixel coordinates of the word. The pixel coordinates are extracted for each successful match for a named entity submitted from the gazetteer to the suffix tree. These are then stored as attributes of the named entity and rendered over the image of the document. Two additional tools are provided to support image magnification, and to allow users to alternate between the image and the text.

The example in Figure 5.17, demonstrates how additional metadata is attached the named entities for location, which is displayed on a Google map when the researcher hovers over the named entity.

## King James Bible

The screenshot displays a web-based interface for analyzing the King James Bible. At the top, navigation tabs include 'King James Bible', 'Occupation', 'Spy', 'Judges', and 'Chapter 18'. The main text area contains verses 1 through 10 of Chapter 18, with named entities highlighted in colored boxes: Israel (green), Dan (green), Eshtaol (green), Zorah (green), Micah (red), Laish (green), priest (blue), God (red), and Put (green). Below the text is a map of the region, with red dots and numbered markers (1-10) corresponding to the locations mentioned in the text. A sidebar on the right, titled 'Named Entities', contains three expandable lists: 'People' (Dan, Ephraim, God, Israel, Micah), 'Location' (Dan, Ephraim, Eshtaol, Israel, Laish, Mount, Mount Ephraim, Put, Zorah), and 'Occupation' (Priest, Spy).

Figure 5.18: The named entity view for the *King James Bible*.

Figure 5.18 illustrates the named entity tool applied to the *King James Bible*. Gazetteers for well known texts, such as this one are widely available. Researchers are able to discover the documents on the basis of the people and locations mentioned, but also a range of occupations identified by gazetteers related to biblical roles, which is not an entity type that is often supported by the tools currently available to researchers.

## Giorgio Vasari archive

The screenshot shows the Named Entity Tool interface for the Giorgio Vasari archive. The main text area displays a snippet of text from Vasari's 'Lives of the Artists', with named entities highlighted in red. The sidebar on the right shows the 'Named Entities' panel, which is currently set to 'People'. Below this, the 'Location' panel is expanded, showing a list of locations: Florence, Naples, and Tuscany. The text in the main area includes the following numbered paragraphs:

- 1 VASARI'S LIFE OF GIOTTO
- 2 GIOTTO (1267-1337)
- 3 Vasari's Lives of the Artists
- 4 NOW IN THE YEAR 1276, in the country of Florence, about fourteen miles from the city, in the village of Vespignano, there was born to a simple peasant named Bondone a son, to whom he gave the name of Giotto, and whom he brought up according to his station. And when he had reached the age of ten years, showing in all his ways though still childish an extraordinary vivacity and quickness of mind, which made him beloved not only by his father but by all who knew him, Bondone gave him the care of some sheep. And he leading them for pasture, now to one spot and now to another, was constantly driven by his natural inclination to draw on the stones or the ground some object in nature, or something that came into his mind. One day Cimabue, going on business from Florence to Vespignano, found Giotto, while his sheep were feeding, drawing a sheep from nature upon a smooth and solid rock with a pointed stone, having never learnt from any one but nature. Cimabue, marvelling at him, stopped and asked him if he would go and be with him. And the boy answered that if his father were content he would gladly go. Then Cimabue asked Bondone for him, and he gave him up to him, and was content that he should take him to Florence.
- 5 There in a little time, by the aid of nature and the teaching of Cimabue, the boy not only equalled his master, but freed himself from the rude manner of the Greeks, and brought back to life the true art of painting, introducing the drawing from nature of living persons, which had not been practised for two hundred years; or at least if some had tried it, they had not succeeded very happily. Giotto painted among others, as may be seen to this day in the chapel of the Podesta's Palace at Florence, Dante Alighieri, his contemporary and great friend, and no less famous a poet than Giotto was a painter.
- 6 After this he was called to Assisi by Fra Giovanni di Muro, at that time general of the order of S. Francis, and painted in fresco in the upper church thirty-two stories from the life and deeds of S. Francis, which brought him great fame. It is no wonder therefore that Pope Benedict sent one of his courtiers into Tuscany to see what sort of a man he was and what his works were like, for the Pope was planning to have some paintings made in S. Peter's. This courtier, on his way to see Giotto and to find out what other masters of painting and mosaic there were in Florence, spoke with many masters in Sienna, and then, having received some drawings from them, he came to Florence. And one morning going into the workshop of Giotto, who was not his labourer, he showed him the mind of the Pope, and at last asked him to give him a little drawing to send to his Holiness. Giotto, who was a man of courteous manners, immediately took a sheet of paper, and with a pen dipped in red, fixing his arm firmly against his side to make a compass of it, with a turn of his hand he made a circle so perfect that it was a marvel to see it. Having done it, he turned smiling to the courtier and said, "Here is the drawing." But he, thinking he was being laughed at, asked, "Am I to have no other drawing than this?" "This is enough and too much," replied Giotto, "send it with the others and see if it will be understood." The messenger, seeing that he could get nothing else, departed ill pleased, not doubting that he had been made a fool of. However, sending the other drawings to the Pope with the names of those who had made them, he sent also Giotto's, relating how he had made the circle without moving his arm and without compasses, which when the Pope and many of his courtiers understood, they saw that Giotto must surpass greatly all the other painters of his time. This thing being told, there arose from it a proverb which is still used about men of coarse clay, "You are rounder than the O of Giotto," which proverb is not only good because of the occasion from which it sprang, but also still more for its significance, which consists in its ambiguity.
- 7 tondo,
- 8 "round," meaning in Tuscany not only a perfect circle, but also slowness and heaviness of mind.
- 9 So the Pope made him come to Rome, and he painted for him in S. Peter's, and there never left his hands work better finished,

Figure 5.19: The named entity view for the *Giorgio Vasari archive*.

The named entity tool for the *Giorgio Vasari archive* in Figure 5.19 extracts the named entities from both the English and Italian translations of the original text, which supports the discovery of the documents across different language corpora. Furthermore, the named entities provide a natural way for researchers to explore the work of the various artists, which was not previously available to the researchers. Although the documents are divided according to artist, Giorgio Vasari compared the work of artists who depicted similar themes, or adopted similar approaches. Therefore, the named entities extracted from the document content, provide more coverage than the that supported by the metadata.

## 5.6 DISCUSSION

The chapter presented an outline of the generic set of mining tools developed for the current version of *Samtla*. The mining tools provide the means for researchers to interact with and explore the documents through browsing and comparison. The *SLM*, described in Chapter 4, was updated with a metadata language model to integrate the metadata model probabilities in to the ranking of the documents. The contribution of the metadata model is uniform for all documents, but can be easily adapted to include a weighted term for each metadata field to describe its contribution to the probability for the query given the metadata model. For instance, we may wish to give more weight to the title or the date of publication, to assign more importance to search results with query matches in the headline text of the article, or to the most recent articles, respectively. Furthermore, the metadata formed the basis for a browsing architecture that enables researchers to explore a collection of documents in a more intuitive way, where the main topics are extracted from the collection and provided as a series of unique paths to the documents. The search tool was supplemented with a search filter constructed from the index of the document metadata fields containing matches for the query.

The hybrid-recommender system implemented in *Samtla* was described in Section 5.3, and helps researchers explore a large space of related material, by leveraging the activity log data for all users in the research community in order to construct a profile of the search terms and documents that are attracting the most interest. The recommendation tools also provide new users with a starting point from which to begin exploring the archive, and also as a method to track the evolving research interests of the community as the popular searches and documents update along with time. The advantage of the hybrid-recommender system approach is that the system-based component mitigates against the “cold-start” problem that can affect purely user-based approaches e.g. collaborative filtering. This is due to the fact that the system-based approach does not require a specific amount of data to construct a user preference model, as related material is identified on the basis of archival content. Furthermore, the recommender system adopts many of the established components of the underlying infrastructure, described in Chapter 4.

The related query tool relies on the collection model  $C$ , stored in the

suffix tree data structure, to locate possible alternatives for the query, which are generated automatically through a series of string permutation methods, or the application of a small set of character rules defined by the researcher, which produce related queries on the basis of more complex language processes. The related queries are ranked according to their global probability given the collection model  $C$ , which provides researchers with the most likely related queries given the statistics of the archive.

The related documents tool was constructed from the *SLM*  $n$ -gram probability distributions for pairs of documents, which was measured using the *Jensen-Shannon Divergence*, a popular method for measuring the similarity between two probability distributions. The related document tool was designed to help researchers to identify semantically related documents very quickly given any document in the digital archive, and to provide access to a further tool, the document comparison tool (discussed in Section 5.4), which allows researchers to explore the “relatedness” of the documents through visual mining of large and small shared-sequences.

Named entity tools were also identified as being important to researchers. Identifying and extracting named entities from the document text is achieved using a simple approach involving gazetteers, which are supplied by the researchers, and have been found to be sufficient for many of the archives supported by the *Samtla* system. The mining tools were each developed to address the specific needs of the research groups. The current set of mining tools represent the most common tasks that were identified as being important to our researchers for which there were no tools available that could perform the required analysis. The majority of the mining tools were developed to address the needs of the historians working with the *Aramaic Magic Bowl archive* (see Chapter 3). The document comparison tool, in particular, was considered an integral tool due to the variability between the documents due to differences in language, dialect, author, and time period.

## CHAPTER 6

# USER INTERFACE

This chapter introduces the *Samtla* user interface (*UI*) reflected by the *view* component of the supporting architecture (refer to Chapter 3). The chapter begins with Section 6.1, which discusses the principles adopted for the design of the user interface, with reference to the issues faced by researchers (see Chapter 2). Next, a discussion of the main structure of the interface follows, in Section 6.2, which introduces the main regions of the web application page layout. The remainder of the chapter is divided according to the different tasks that researchers can perform in the *Samtla* system. The first is search, presented in Section 6.3, which discusses the user interface components related to the generation of the *search* results snippets, recommended queries, and the researcher’s search and browsing history. The second task, represented by *browsing*, is supported by two different structures, a vertical list view for easy navigation of large sets of items, and a treemap view for summarising the availability of data in the archive, both of which are introduced in Section 6.4. Once the relevant document is located the researcher is presented with a flexible document view for close-reading of the text, presented in Section 6.5, which displays the raw text, or original scanned image of the document. At the document level, the user has access to a further tool reflected by the *document comparison* tool, which is described in Section 6.6 and supports the comparison of small and large variable length character sequences represented by the “parallel passages”. Researchers also have access to additional tools for viewing the metadata, related documents, and named entities. The chapter concludes, in Section 6.7, with a summary of the main features of the user interface.

## 6.1 OVERVIEW

The user interface provides the mechanism for users to access and modify the data stored by a system. The user interface is therefore an important component of system development responsible for interpreting the input, representing the information need of the user, and rendering the appropriate output in response. In short, it provides the main interface between the user and the data. A user will often judge the utility of a tool on the basis of the user interface alone, and the research reveals that it has not generally been a priority for tool developers to address the need for intuitive interfaces when developing tools for humanities researchers [98].

An effective user interface is one which enables the user to access the data quickly with minimal interference or visual distraction. In other words, an effective interface is one that performs the majority of the work with very little information, or effort, on the part of the user. Visually distracting user interfaces are often the result of “visual clutter” due to an absence of white space resulting from unrelated or irrelevant options and information [95]. Poor interface design often results in users losing confidence in a system [175], and subsequently abandoning it before they have fully understood its potential to help them complete a specific task (discussed in Chapter 2).

Shneiderman (1986) [175] proposed eight heuristics, or “golden rules of interface design”, to act as a starting point for developing user friendly interfaces, which are summarised as follows:

1. **Consistency.**

Consistency is an important aspect of user interface design, and relates to the idea that the same sequence of actions carried out by the user, should generate the exact same output each time. Consistency also covers the style, and layout of the user interface, where adherence to well-established and understood user interface interactions increase the usability of the interface as users are familiar with them. For example, functionality is best represented by universally established iconography (e.g. a disk icon to denote a loading or saving function), and text descriptions (e.g. “find” or “search”, rather than “discover”, as a label for the search tool).



## 2. **Shortcuts.**

As users increase their use of the system there may be information that they wish to access regularly. Shortcuts provide the means for users to access this information without repeating the potentially time-consuming actions required to reproduce the results, such as searching or browsing the documents.

## 3. **Feedback.**

Every interaction made by the user should result in some form of visual feedback, or information from the system, in order to reassure the user that their interaction with the system has been acknowledged. This is particularly important for tasks that require further processing or searching of the data, which may require more time.

## 4. **Closure.**

Closure refers to the user feeling as though they have successfully completed a task. This means grouping the system's functionality together to create a stream of processes with a beginning, middle, and end. When the user completes a task they are able to release any information in their short-term memory related to achieving their goal, allowing them to focus on a new set of actions, or pause the analysis.

## 5. **Error handling.**

The system should detect errors and provide a simple solution to resolve them, or display information that would enable the user to respond to it appropriately.

## 6. **Reversal of actions.**

The ability to "undo" an action provides users with a sense of freedom to explore a system, knowing that any unintended actions can be undone. The interface should support the ability to not only undo single actions, but also whole groups of actions.

## 7. **Control.**

The user must feel in control of the system at all times, which means that functionality or changes to the state of the system should be instigated by the user, rather than by the system on the user's behalf.

### 8. Reduce cognitive load.

Users may find certain tasks such as search and close-reading of the documents cognitively intensive. The presence of unnecessary “visual clutter” in the user interface can have an impact on the user’s ability to perform tasks. Visual clutter tends to be caused by an absence of white space, or overloading the user with tool options and information that are not relevant to the analysis [95]. In order to reduce the cognitive load on the user, interfaces should consolidate information in to meaningful groups, and displayed when they make sense in the given context of the analysis.

These principles are intended to “place the user at the center of the design”, where they act as a “design partner” in a user-centered, or participatory design approach [50].

The *Samtla* user interface has been designed in collaboration with the research groups, where early prototypes of the interface were evaluated as part of an iterative design process, and users were encouraged to feedback on the different prototypes. From the feedback, it appeared that the researchers preferred an interface where the majority of the display area is dedicated to the content of the archive, and output of the tools, as represented by the search results, browsing, document view and comparison, which are tasks the users’ will want to perform regularly or have access to at all times. Supporting tools such as the user’s past activity, or recommended queries and documents, were considered as secondary information and tools, and could be optionally displayed when required. The main structure of the interface has been designed to make a distinction between primary and secondary information and tasks.

## 6.2 THE INTERFACE STRUCTURE

The display area of the *Samtla* interface is divided in to header and main body regions, which is further divided in to a three column layout (see Figure 6.1). The header displays tools reflecting primary actions, which include search, display of previous search and browsing history, changing the browser viewing preferences, and sending bug reporting or feedback (see Figure 6.2). The left column of the main body displays secondary information, such as users previous activity in the system, and up-to-date information about the research

interests of the community. The information is accessible at all times, and can be hidden or displayed by selecting the appropriate tool icon from the header, or resizing the column.

The central column displays all primary information such as the search results, document text, and the browsing and document comparison tool output. The default setting is to display the output of the browsing tool to enable users to explore the archive at any point in the analysis. The right column is dedicated to secondary information and actions that are dependent on the context of the analysis, for example, the display of the metadata search filter when viewing search results, or displaying the metadata record for the document, and secondary actions, such as the related documents, and named entity tool at the document-level. By partitioning the display according to functionality

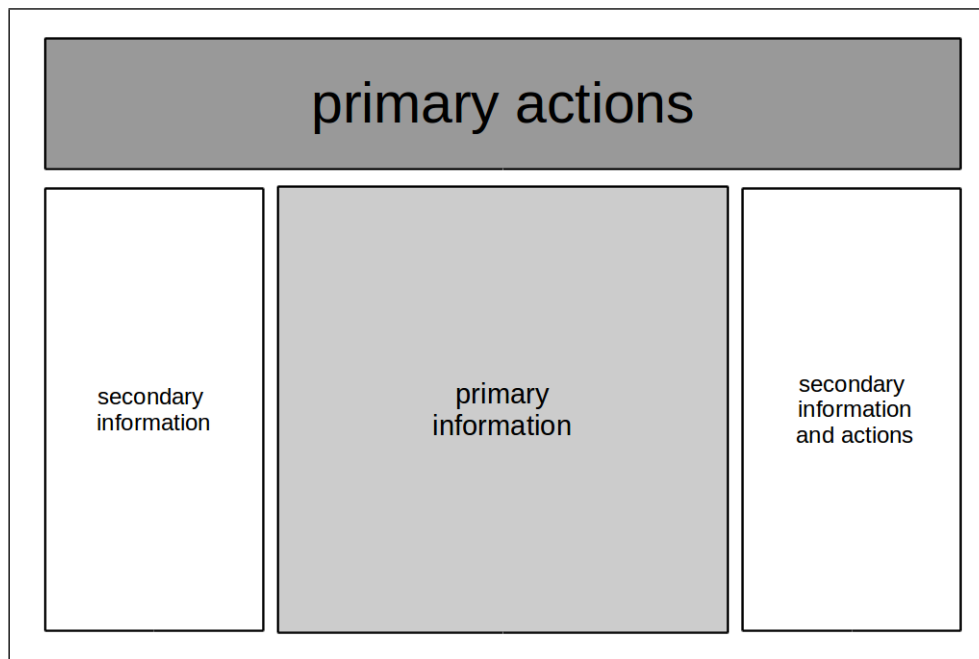


Figure 6.1: The basic structure of the user interface is composed of a header and main body divided in to left, middle, and right columns.

the user is able to focus their attention on the output of the system generated by primary actions, including search, browsing, and viewing the documents, whilst the display of secondary information and tools is optional. The actions in the system are displayed as icons, or text labels. Where possible, text labels have been preferred, as studies suggest that users often find them more meaningful than icons [95]. On the otherhand, icons have been adopted when space is limited, for instance, in the header of the page, and the areas dedicated to

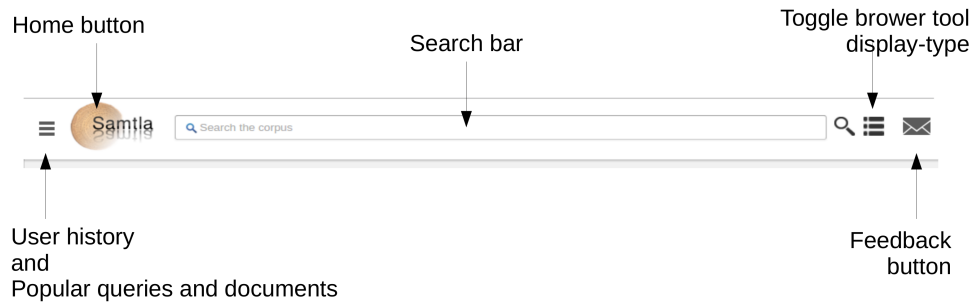


Figure 6.2: The header provides access or settings for the most often used tools represented by search, recommendations, and browsing preferences.

secondary information and actions (see Figure 6.2).

*Samtla* also displays information on the current state of the system as a trail of links known as a “breadcrumb” [103]. The breadcrumb acts as a secondary form of navigation [153], as well as a visual record of the user’s activity since entering the system. Other advantages of the breadcrumb include providing the main undo function, where users can return to the results of their previous actions by selecting higher-level elements of the breadcrumb.

User testing suggests that although breadcrumbs can be overlooked, or ignored by users, they are easy to interpret, with respect to what they represent, and how to use them. Furthermore, they take up very little space in the user interface, but can potentially facilitate navigation [153].

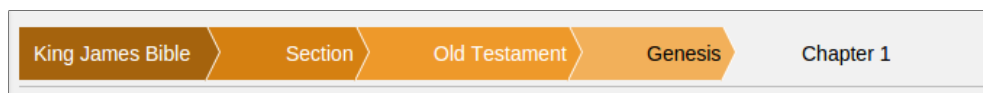


Figure 6.3: An example breadcrumb describing the path taken from the root level to the document level of the Bible version of *Samtla*.

The breadcrumb trail is generated from the path created by the users activity or navigation in the system. For example, when users are browsing, each node visited in the hierarchical graph structure, is appended to a list (introduced in Chapter 5). Any subsequent actions, such as search, document views, and document comparison, are also appended to the same list. Each item of the list is then displayed as a link in the area of the interface dedicated to primary information.

## 6.3 SEARCH

There are several processes involved when users are searching for information in a system. First, the user will formulate the terms of a query that describes the topic of the documents that they wish to retrieve from the system. Next, the query is entered and submitted via a search bar, or equivalent text prompt.

The user will then be presented with a ranked list of search results, which are often represented by a vertical list, due to the list structure being universally understood by the majority of users, as well as being particularly suited to the display of long lists of items, and the vertical order of the entries promotes usability as users can quickly scan the results compared to a horizontal ordering of the entries [102].

Depending on the relevance of the retrieved documents, the user may decide to refine the terms of their original query and repeat the search. Alternatively, the users may wish to filter the relevant documents for those containing specific attributes, aside from matches in the document text, e.g. matches for the query in the title text. Consequently, it is important to consider this cycle of search, review, and refinement in the development of the user interface when applied to search.

In *Samtla* the search bar is permanently displayed to the user in the header of the interface, allowing the researcher to search the archive at any point in the analysis. The search results are composed of snippets generated from the document text with highlighted matches for the  $n$ -grams of the query. The snippets act as the main form of feedback for the user to assess the relevance of each document in the search results (see Section 6.3.1). The system also displays a search filter to enable users to filter the documents for specific attributes stored in the metadata for each document retrieved, discussed in Chapter 5.

The process of search and refinement of the user's query is aided through the display of related queries, which represent alternative forms of the query according to the properties of natural language (see Section 6.3.2). Lastly, users may also require help in locating the interesting documents in the archive, which they may not be able to identify through an appropriate query. Consequently, in these situations the user may turn to a colleague or ask advice from an expert of the archival content such as a librarian. *Samtla* provides

similar functionality through recommended queries and documents that represent the queries and documents that the community of users rely on the most (see Section 6.3.3).

### 6.3.1 Search snippet generation

Snippets are one of the most common approaches adopted [102] for displaying matches to the query in the document content. Each snippet represents a summary of the document generated from the index returned by the retrieval model. Snippets are classified as either static or dynamic, depending on their application in a system. Static snippets will return the same summary of the document each time they are generated, for instance, from the introductory text of the document. On the hand, dynamic snippets are generated in conjunction with the scoring of queries, with each occurrence of the query highlighted in the document. The snippets are composed of one or more summaries containing all or part of the query terms, with additional context provided to the left and right of the query matches.

The challenge with snippet generation is to choose a method that captures coherent portions of text, for instance complete phrases, as users tend to prefer summaries that are easy to interpret [142]. Therefore the aim of snippet generation is to produce snippets that are informative enough to represent the topic described by the query, highly readable to facilitate the user in choosing the most relevant document, and concise enough to make the best use of the space available when presenting a long list of search results [142].

The *Samtla* system produces dynamic snippet windows for summarising the documents. The snippets are generated from the index returned by the suffix tree at query time. The index provides the means for extracting the matching portions of the query from the document text, using the start and end positions of each matching  $n$ -gram of the query. The extracted sequences are then expanded to the left and right to provide the user with some context to the query. The length of the context is tunable and is defined by a parameter  $w$  reflecting the maximum length of the context in characters. For our purposes this is set to  $w = 100$  characters, however in future versions this could be provided as a user setting.

The snippets are scored by calculating the length of the matching  $n$ -grams

found in the snippet, which is then interpolated with the total count of all  $n$ -grams in the snippet, where the algorithm assigns more weight to snippets containing all of the query  $n$ -grams. This ensures that the snippets are ranked in such a way that the top snippets will contain full matches to the query, before presenting snippets with only partial matches. The score for each potential snippet, extracted from the document text, is defined as follows

$$\text{SnippetScore} = \delta^\alpha \mu^{(1-\alpha)}, \quad (6.1)$$

where  $\delta$  represents the cardinality of the set of  $n$ -grams in the snippet, and  $\mu$  is the count of all  $n$ -grams (including repetition). The two terms composing the snippet scoring formula are interpolated with a weighted term, defined as  $\alpha = 0.9$ . The high setting for  $\alpha$  ensures that the snippets are biased towards those that contain full matches for the query. The snippets are sorted in descending order according to their respective score, and the top-three scoring snippets are selected as a preview for the document. As an example, Table 6.1 displays the potential snippets generated for a single document, together with the score for each snippet given by ((6.1)).

When displaying the search results, the documents are organised according to exact and partial matches for the query. The documents are partitioned into bins by the length of the query match. The documents for each bin are then sorted by probability, inferred from the *SLM*. The approach ensures that users will always be presented with full matches for query at the top of the search results, before any partial matches are presented. Partial matches do not encompass the full query, but may still be of interest to the user, or may aid the user in reformulating the terms of their query, in order to refine the search results. Each document in the search results is then represented by a title, and a corresponding snippet window.

A further component of the snippet window is the metadata snippet text. The matches for the query given the metadata model  $B$  (discussed in Chapter 5), are displayed at the bottom of each document snippet with the metadata field and the query highlighted in the corresponding value. Snippet generation is performed in the same way as described for the document text, the only difference is that only the top scoring snippet is returned for the metadata. This helps to maximise the number of documents that can be displayed

Rank	Score	Snippet text
1	0.648	...with the council, answered, <u>Hast thou appealed unto Caesar?</u> <u>unto Caesar</u> shalt thou go. And after certain days king Agrippa and Bernice came <u>unto Caesarea</u> to salute Festus...
2	0.072	...in him. And when he had carried among them more than ten days, he went down <u>unto Caesarea</u> ; and the next day...
3	0.072	...deliver me unto them. I appeal <u>unto Caesar</u> . Then Festus, when he had co...
4	0.054	...led to be reserved unto the hearing of Augustus, I commanded him to be kept till I might send him <u>to Caesar</u> . Then Agrippa said unto...
5	0.042	...that Paul should be kept at <u>Caesarea</u> , and that he himself would depart shortly thither...
6	0.042	...three days he ascended from <u>Caesarea</u> to Jerusalem. Then the...
7	0.042	...the temple, nor yet against <u>Caesar</u> , have I offended any thing...
8	0.042	...Then said Paul, I stand at <u>Caesar's</u> judgment seat, where I...

Table 6.1: The ranked search snippets generated for a single document matching the query “Hast thou appealed unto Caesar?”, submitted to the *King James Bible*.



in the available area to facilitate the quick scanning of the search results. In the event that it is not possible to generate a snippet for the document e.g. when the document represents an image, the system handles the error by rendering only the title of the document.

### 6.3.2 Related queries

The related queries are displayed to the user at the same time as the search results. A maximum of ten related queries are returned in response to the user's query, and sorted by probability according to the collection model  $C$  component of the  $SLM$  (see Chapter 4). The related queries are displayed horizontally above the search results, and sorted in decreasing order of probability from left-to-right.

An example, presented below, is represented by the query “Nebuchadrezzar”, which has a related query of “Nebuchadnezzar”. This related query represents a less common spelling of the name of a famous King, where the difference of one character  $r \rightarrow n$  is the result of a variation in the transcription of the name according to the Hebrew and Aramaic versions of the original text. When the user selects a related query, a further search is performed, which

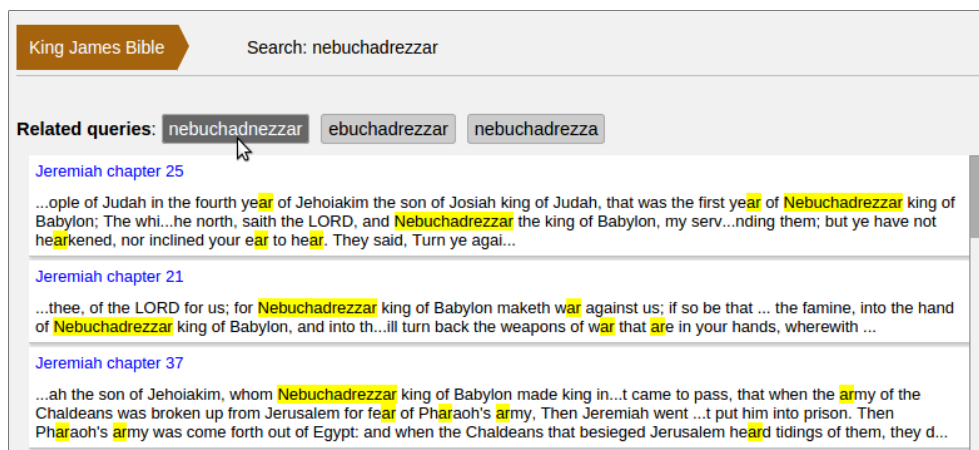


Figure 6.4: An example of the related queries for the query “Nebuchadrezzar”, a name of a King mentioned in the King James Bible.

may itself return further related queries. Continuing in this way, enables the user to iteratively explore a large space of potential queries that are more variable than single character differences identified by the *Type I related queries* approach (refer to Chapter 5).

### 6.3.3 User history and popular queries and documents

The user history is displayed as secondary information, and acts as one of the main shortcuts to previous searches and document views. The recommended queries and documents are refreshed after each new query or document view is recorded in the log data, which triggers an action in the *controller* component of the architecture, which is responsible for updating the user interface with new recommendations. If the user has logged on for the first time, there is no

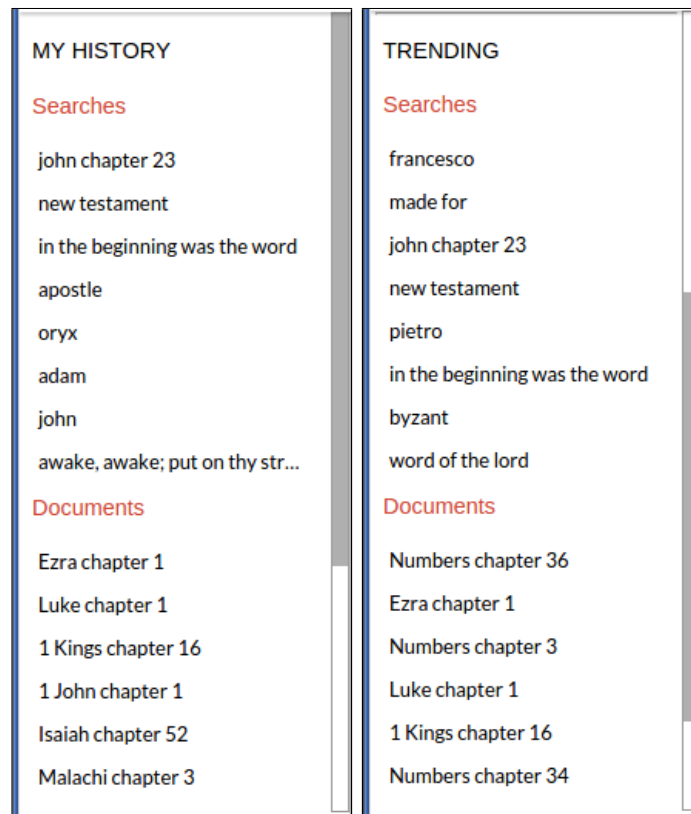


Figure 6.5: The left sidebar showing the user query and document history, and the most popular queries and documents in the whole community of users.

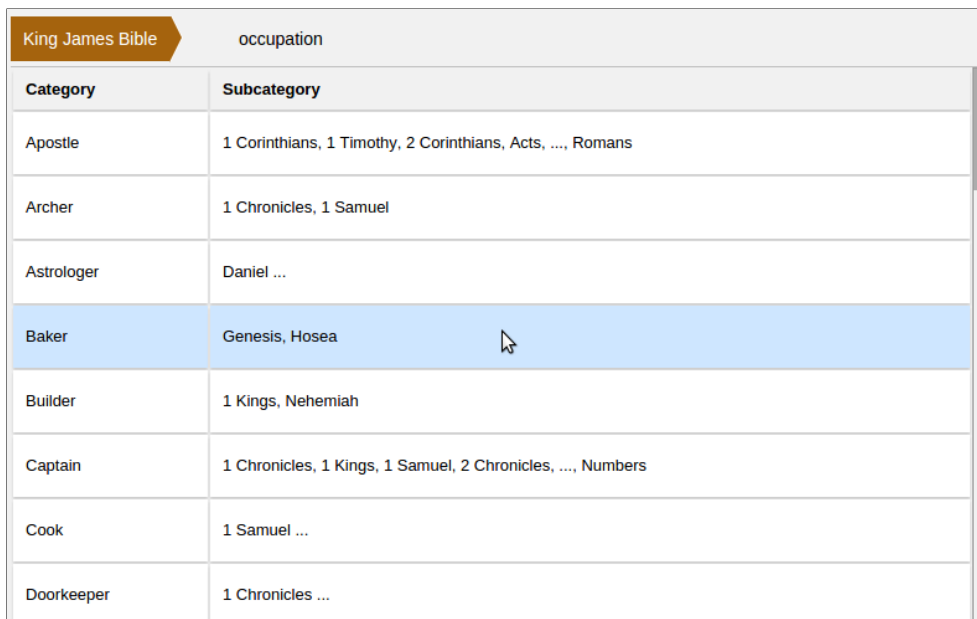
personal search or browsing history due to a lack of user activity. Therefore, the popular queries and documents provide the main starting point for users to begin exploring the archive. The ranked lists are updated asynchronously after each new interaction received from the users by the server, which provides up-to-date information on the research community search and browsing activity.

## 6.4 BROWSING

The browsing architecture in *Samtla* provides users with a number of different routes to the documents based on the structure of the digital archive, the metadata, and the named entities extracted from the document text, see Chapter 5. To recap, the paths to the documents are represented as nodes in a hierarchical graph structure, where each node is labeled with a category representing a field or value from the metadata, a named entity type, or named entity label. The nodes of the graph are then visualised in one of two ways, a vertical list view, discussed in Section 6.4.1, and a treemap view presented in Section 6.4.2. If the available display area is too small and the number of items to display is large, then there is a chance that the treemap will not be able to generate an appropriate layout. When an error occurs, then the browsing tool fallbacks to the vertical list view by default.

### 6.4.1 The vertical list view

The default browsing view is represented by a vertical list. The list is divided in to two columns, with the first column displaying the current categories available to the user, which they can select in order to continue traversing the hierarchical navigation structure to the document level.



King James Bible	
occupation	
Category	Subcategory
Apostle	1 Corinthians, 1 Timothy, 2 Corinthians, Acts, ..., Romans
Archer	1 Chronicles, 1 Samuel
Astrologer	Daniel ...
Baker	Genesis, Hosea
Builder	1 Kings, Nehemiah
Captain	1 Chronicles, 1 Kings, 1 Samuel, 2 Chronicles, ..., Numbers
Cook	1 Samuel ...
Doorkeeper	1 Chronicles ...

Figure 6.6: Browsing the Bible corpus using the list view.

The second column provides a summary of each category in terms of any

additional categories, a list of documents if the next level down represents the document level, or a summary for each document grouped under the given category at the point in the tree preceding the document level.

### 6.4.2 The treemap view

The size and colour of the cells of the treemap can be altered to reflect certain attributes of the collection, for example, according to the category size, by increasing the size of the cells explicitly or introducing colour, as shown in Figure 6.7. This enables users to locate sub-groups of documents with very little effort due to the visual cues that can be embedded in the treemap. Image

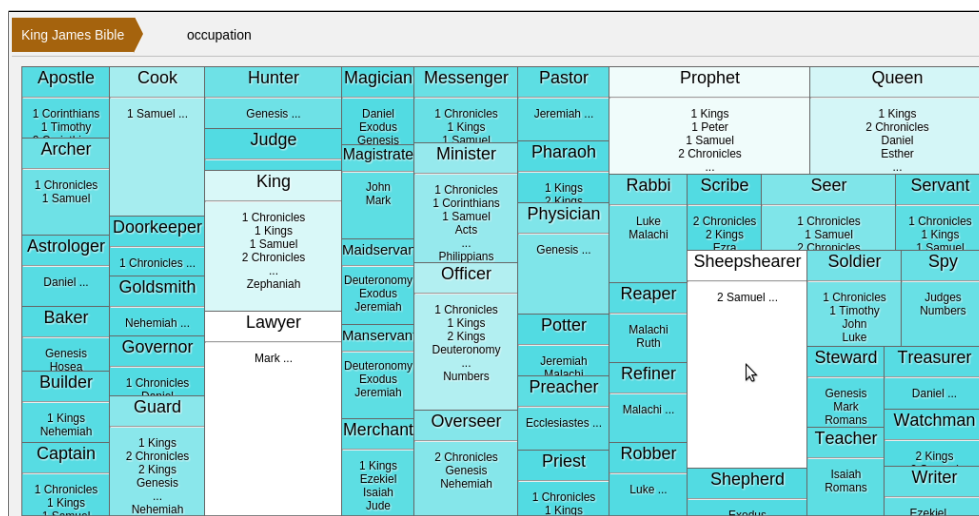


Figure 6.7: Browsing the *King James Bible* through the treemap view generated from the document metadata.

data is also supported by the treemap view, where the images are scaled according to the dimensions of the treemap cells, and labelled according to the documents related to the image. Images are either provided as links to external sites, such as the *Aramaic Magic Bowl archive*, which references the images available at the British Museum photographic archive (see Figure 6.8), or images that accompany the text, such as the paintings referenced in the *Giorgio Vasari archive* (in Figure 6.9). The advantage of the treemap view over the vertical list view is that the breadth and type of information available is displayed all at once, allowing the user to gain a comprehensive overview of the information stored in the digital archive. It also offers a unified approach for different media, such as browsing the images that accompany the documents. Unlike text labels, images are still recognisable at small scales. The

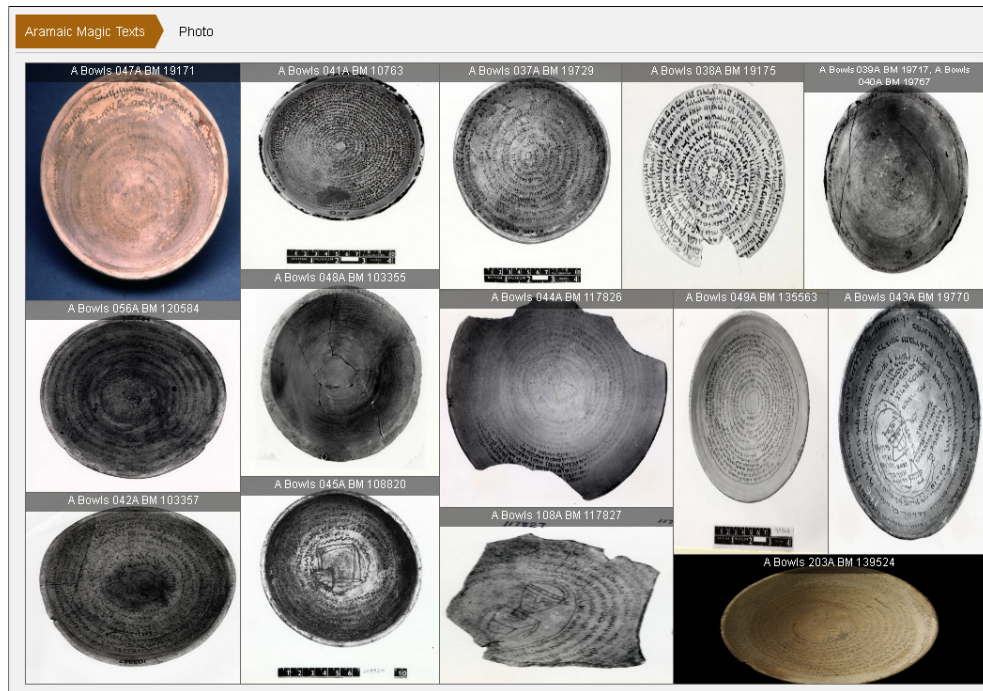


Figure 6.8: Browsing the photograph category of the *Aramaic Magic Bowl* archive.

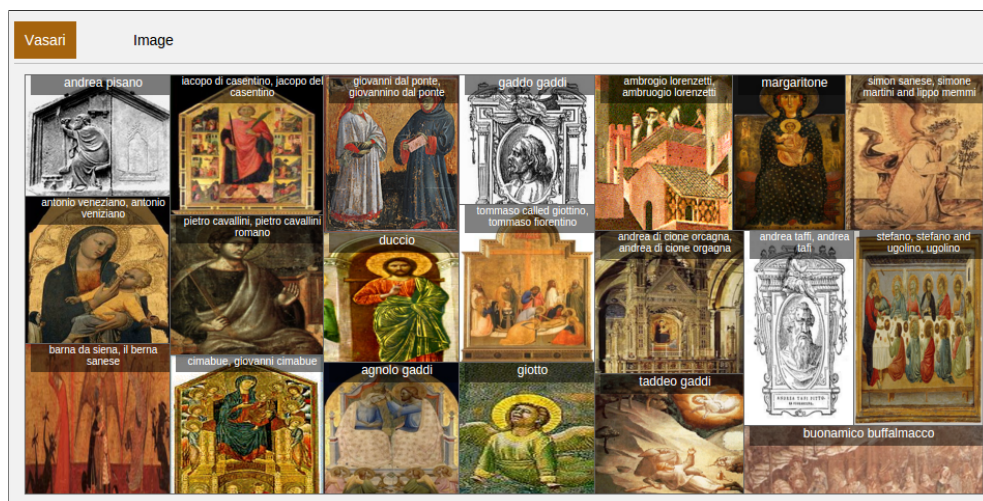


Figure 6.9: Browsing the images cited in the text of the *Giorgio Vasari* archive.

vertical list view, on the other hand, is more familiar to users, is intuitive and easy to use, and appropriate to browsing large groups of items that may be common when clustering documents in large-scale digital archives. Feedback from our users indicated that not everyone appreciated the treemap view, and so the vertical list view is provided as the default view when entering the system.

## 6.5 DOCUMENT VIEW

The document view is accessed through browsing or searching when the user selects a document from the search results, or the document-level category of the browsing tool, represented by the leaf nodes of the hierarchical graph structure. Depending on the method used to access the document, the system will display the document text, or original image of the scanned document, with additional highlighting of the query for search, or named entities when arriving at the document through browsing (see Figure 6.10). Some archives,

The screenshot shows a document view for 'Aramaic Magic Bows'. The main area displays OCR text with numbered annotations (1-9) highlighting specific words or phrases. The sidebar on the right shows the metadata for the document, including title, category, DocID, Parent, Language, Author, Textid, Sokoloffid, Etymology, Patronym, Sex, Length, and Questions.

**Metadata:**

- Title: A Bows 042A BM 103357
- Category: Bows
- DocID: 56
- Parent: קודקאדור
- Language: Jewish Aramaic
- Author:
- Textid: 042A (BM 103357)
- Sokoloffid: 144
- Etymology: The name Gahbiš is a pseudonym meaning "seize evil". The matronym is probably a hybrid, using the Semitic word qadqad, "skull", followed by the Iranian duk, "daughter" (CAMIB, p. 86).
- Patronym:
- Matronym:
- Sex: M
- Length: 627
- Questions: Could the pseudo-matronym קודקאדור relate in fact to the concept of chicken as evil agents in these incantations?

Figure 6.10: The document level for the *Aramaic Magic Bows* archive, which shows the default metadata view for the document.

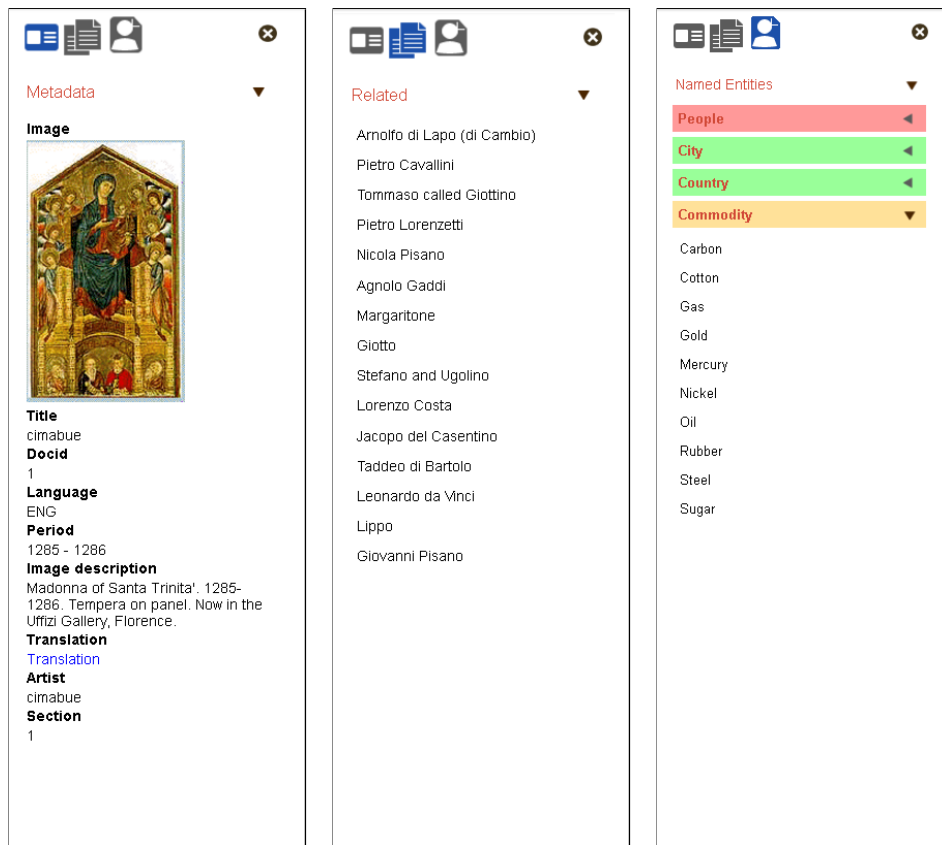
such as the *FT newspaper archive* store the original scanned image together with the *OCR* text, which required some adaption of the document view to support navigation of the text and the full-size image of the newspaper. This also presented a challenge, where additional data layers, represented by the extracted named entities, required rendering in both the raw text, and the image of the original document. Due to the quality of some of the documents, users may find it difficult to extract the information they need, and so presenting the option to view the original image helps to compensate for errors that may have occurred during the scanning process. The document level view contains

further tools for navigating around the image, as well as switching between the alternative formats of the document, as illustrated in Figure 6.11). At



Figure 6.11: The document level for the FT newspaper archive, which displays the original scanned image of the document.

the document level, supporting tools are provided that display the metadata, named entities, and related documents, which provide secondary information and actions related to the document text. For instance, the document text can be filtered for named entities of different types to provide additional context to the document text, or alternatively, the related documents provide users with access to the document comparison tool for when they wish to compare the content of the current document with other representative examples in the archive (see Figure 6.12). The default information displayed to the user at the document level is the metadata, which may also come in a range of formats; including third-party sources of information such as links to external web pages, images, or alternative translations, and published research related to the context of the document (see Figure 6.10).



(a) The metadata view tool.

(b) The related documents tool.

(c) The named entity tool.

Figure 6.12: The document-level tools.

The related documents tool displays the top-twenty most similar documents to the target document being viewed by the user (introduced in Section 5.3.2, and illustrated in Figure 6.12b). The top document represents the document with the highest *JSD* score, and consequently the document with the most similar *n*-gram probability distribution given the currently viewed document. The main role of the related document tool is to provide access to the document comparison tool (described in Section 6.6, below), where users can compare the content of the current document with semantically related documents by selecting them from the related documents list.

The named entity tool is accessed from the right-hand sidebar at the document-level (see Figure 6.12c). When active, the server updates the right-hand sidebar with the named entities extracted from the document (see Section 5.5). Each entity is ordered according to entity type, with people and locations appearing at the top. The default view displays all the entities for the document, whilst selecting the label representing the entity type displays



entities of that type only e.g. all people entities. Furthermore, users can select specific entities, such as the name of a person or country, to show only those results in the document.

The entities are highlighted in the text as colour-coded links. Hovering over a link displays further information stored about the named entities, whilst clicking on a named entity, submits it as a query. Depending on the type of entity, the metadata about each entity ranges from encyclopedic knowledge, etymologies, bibliographic data, and notes. When the named entity relates to a location, an additional window comes in to view, which displays a Google map with green markers rendered for each location referenced in the text, and the currently selected location is highlighted in red. Visualising the location entities on a Google map is a natural choice as it provides additional context about the spatial relationship between the location mentioned in the text. Many of the tools and systems discussed in Chapter 2 incorporate some form of support for named entities, and additional encyclopedic, and bibliographic information. However, these tools are usually provided as separate components. In *Samtla*, the display of named entities and the related metadata is incorporated as part of the same view.

## 6.6 DOCUMENT COMPARISON

Sequence comparison is a difficult task to perform manually, especially over several documents and particularly when some of the sequences may be approximate, or overlapping. The interface for the document comparison has been designed to emulate the process of comparison performed by the researchers. Typically a the researcher would layout the two documents side-by-side, and highlight the sequences that they consider the same or similar. The document comparison is presented as two viewports one representing the target document for comparison, and the other is the document selected from the related document tool. When the tool is first instantiated, the default behaviour of the tool is to display the largest shared-sequence identified, which is highlighted to the user.

The tool is equipped with a control to choose the length of the shared-sequence to view, with the minimum being 3-gram and the default setting displaying the longest sequence found between the two documents. This en-

ables users to investigate both large shared-sequences spanning several lines of text to smaller sequences representing a word. Appearing above each document is a small horizontal map, which summarises the sequences shared between the two documents to provide feedback on the location and extent of the shared-sequence. The map also helps the user navigate through the document when shared-sequences appear outside of the viewport. For example, shared-sequences may be limited to the introduction or conclusion of the text. When a user selects a shared-sequence, all sequences with the same identifier are highlighted across the two documents (see Figure 6.13), and the viewport scrolls both document windows to align the sequences in order to facilitate their comparison.

The design of the document comparison tool is the result of an iterative design process, based on feedback from our users, and the orientation of the document windows attempts to emulate the manual process of document comparison where the user would layout two documents side-by-side and attempt to locate similarities by marking up the text. Some of the tools and systems cited in Chapter 2, support document comparison, but the approach tends to display the “parallel passages” as interlinear text, with one sequence under another. However, the layout selected for the document comparison is more efficient, since the researcher can quickly scan the text of both documents vertically, rather than horizontally, as required by interlinear text [102].

## 6.7 DISCUSSION

To summarise, the *Samtla* user interface was developed under a participatory design approach, where the users provided feedback on a series of prototypes until the desired look and functionality was attained. *Samtla*'s user interface is designed on the basis of “progressive disclosure” or “context-dependency”, where secondary information and actions are only presented to the user when they make sense in the context of the analysis. For instance, the user will only require document-specific tools when they are viewing a document, otherwise they are hidden from view. Users also have control over the configuration of the “work area”, where the majority of the interface can be reduced to focus only on primary information and actions. This helps to reduce visual distractions from the interface during cognitively intensive tasks such as query formulation and close-reading of the document text, as well as to facilitate the navigation of large images, such as those stored in the *FT newspaper archive*.

The interface communicates state changes to the system in a number of different ways. First, whenever the user interface communicates with the server (the *model* component in Chapter 3), a small animation shows that the user's request has been acknowledged and is being processed. Secondly, each user interaction is updated in the breadcrumb of the display responsible for status updates. The breadcrumb also permits the reversal of actions by allowing users to return to a previous state of the analysis such as browsing or the search results. Lastly, each element of the interface displays a short description to indicate their purpose or use, when users hover over them with the mouse pointer. The interface also provides several shortcuts for commonly performed actions, such as favourite queries or documents through the secondary information displayed as the user's search and browsing history, which allows users to return to relevant documents that they analyse regularly or wish to return to after each session.

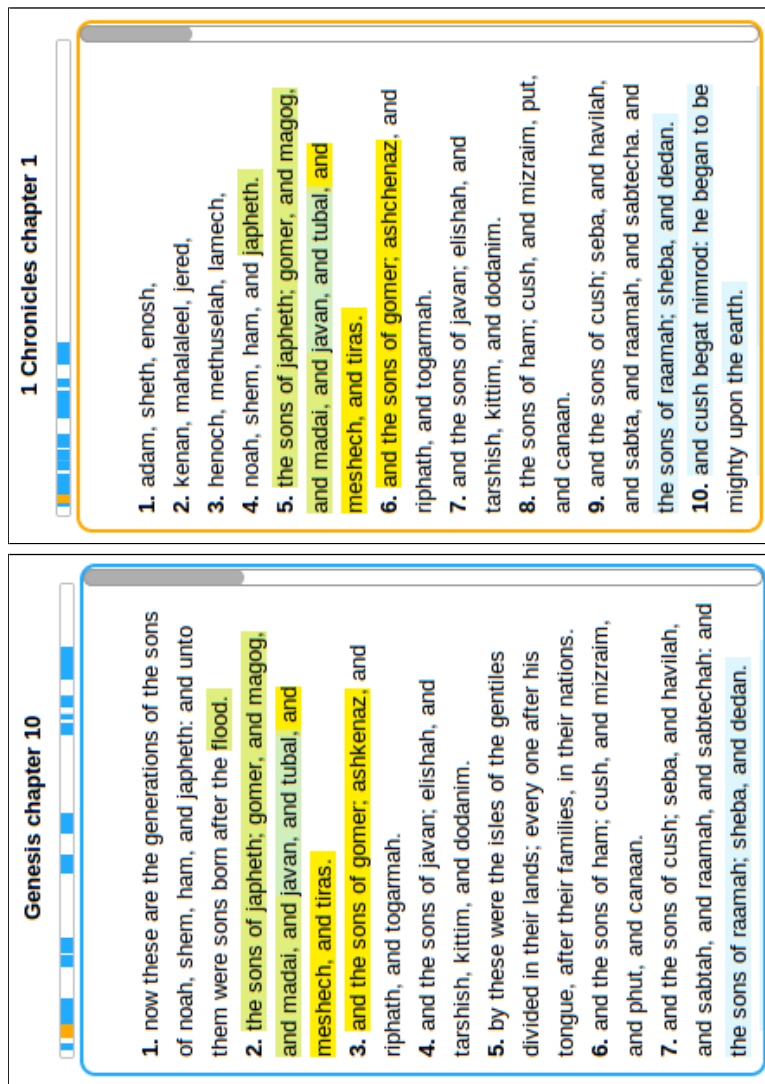


Figure 6.13: An example of Samtla document comparison. The document comparison interface shows a pairwise comparison of the target document (left) and a document selected from the list of related documents (right). Sequences highlighted in yellow reflect the currently selected sequence, and blue represents all sequences shared between the two documents.

# CHAPTER 7

## EVALUATION

This chapter presents the details of a novel approach adopted for evaluating the performance of the infrastructure represented by the data model of the infrastructure, based on a character-level  $n$ -gram *SLM* stored in a space-optimised  $k$ -truncated suffix tree data structure. Section 7.1 introduces the approach adopted for the evaluation of the information retrieval component of the *Samtla* system, and Section 7.2 gives a brief description of the crowdsourcing model with details on the crowdsourcing platform selected for the evaluation.

The experimental design of the evaluation is described in Section 7.3, which discusses the preparation of the queries, the choice of relevance scale, and filtering criteria adopted for selecting the human assessors. In addition, the section introduces the statistical measures used for evaluating the system performance through a series of non-parametric measures. These include the *Normalised Discounted Cumulative Gain (NDCG)* [110], which is adopted as a system-based assessment of the ranking quality of the search results. A further set of non-parametric correlation measures [60,90] describe the level of user satisfaction with the search results by comparing the system ranking generated by the *SLM*, with a user ranking generated from the overall “wisdom” of the crowd for each query. Furthermore, the significance of the results are measured with the *bootstrap* approach, which is flexible to the test statistic, making it compatible with the non-parametric performance measures adopted for assessing the results.

The chapter concludes, in Section 7.4, with a discussion of the evaluation results, and an assessment of the *SLM* performance, which supports the

information retrieval and mining tools developed for the *Samtla* system.

## 7.1 OVERVIEW

The evaluation presented in this chapter adopts crowdsourcing as a method for employing the skills of a group of users for the purpose of evaluating the system performance. Crowdsourcing is an attractive platform for researchers, since the pool of users is large and diverse with respect to their individual social, economic, and educational backgrounds. This enables the researcher to obtain a more representative sample of the population. Retrieval models are not perfect, and this is because the notion of relevance is a subjective concept, and users may have different criteria for assessing the relevance of documents in search results [162]. Despite this fact, research focuses on developing algorithms and approaches that attempt to rank documents as close to a human assessor as possible, by comparing the algorithms output to a ranking of the documents representing the “ground truth”, obtained from the relevance judgements of human assessors [89].

Evaluation of the underlying retrieval model is an important aspect of search engine development. Researchers use the data gathered to assess the effectiveness of parameter tuning, and choice of ranking algorithm. Search engines are commonly evaluated using the *Cranfield* paradigm [186], which measures system performance on the basis of a standard test collection, with a set of topics (represented by queries), and statistical measures that permit the comparison of performance across various systems. The data provided by an evaluation provides researchers with feedback on how changes to the system affect the system’s ranking performance. System evaluation can be divided in to two main types:

1. **System-based:** Measures the ranking quality of the search results.
2. **User-based:** Assesses the users’ level of satisfaction with the search results.

A user-based evaluation is often preferred as it provides a way to directly assess the main objective of any information retrieval system, which is to address the users information need in response to the topic described by the terms of their query [186]. The experimental setup for measuring the performance of an

information retrieval system requires three components; a set of queries, a set of relevance grades for users to express their judgements of relevance, and some statistical measure for assessing the similarity of the user generated ranking with the system ranking:

1. **Queries:** a set of queries and the corresponding top- $n$  search results. The search results commonly contain a small snippet for each document to provide to user with some context for the query to enable them to assess the relevance of each document.
2. **Relevance scale:** a set of two or more relevance grades that the user can assign to the documents in the search results. Multiple relevance grades enable the evaluation to assess relevance at different degrees.
3. **Statistical measures:** one or more statistical measures for comparing the system ranking to the crowd opinion or “wisdom”, generated from the relevance grades assigned by the human assessors.

The evaluation presented in this chapter, adopts a novel approach to assessing system performance by adopting crowdsourcing as a platform for enlisting users to act as human assessors, which provides quick access to a large pool of diverse and globally distributed users.

## 7.2 CROWDSOURCING

Crowdsourcing is a web-based business model [66] that enables companies and individuals to employ the skills of people from a distributed community, in order to perform some task in return for a small reward. These tasks are often large in scale or complex, and therefore time consuming as a result. Crowdsourcing in information retrieval involved outsourcing manual tasks such as data-annotation, labeled-data collection for training models, and system evaluation. This process was often completed in-house with a limited workforce, which could be a slow process involving several days of work, depending on the size of the task [117]. The fact that a large group of people can be enlisted to form a crowd of users for a specific task, who are globally dispersed, means that tasks can be completed at any hour of the day. There is also the potential for reducing bias in aggregated results, compared to in-house evaluations, due to the diversity and representativeness of the workers in terms of

demographic [129]. There are a number of crowdsourcing platforms available to researchers for enlisting a crowd of users for the purposes of classification and evaluation:

- *Amazon Mechanical Turk*<sup>1</sup> (*MTurk*) is one of the better known ones [52, 122, 193]. Workers complete tasks, and are then presented with a URL, which activates a payment for their completed submission. The advantage of this platform is the level of flexibility in how tasks can be defined. *MTurk* provides task creators with a templating and editing tool for designing the layout of the task. Furthermore, the task can be hosted on an external server to the platform, which enables the task creator to design more complex and dynamic tasks that can respond to the input from the user in real time. The main limitation of the platform, however, is its restricted availability to residents in the United States of America.
- *Crowdfunder* is a popular large-scale crowdsourcing platform, with nearly two million workers [24]. *Crowdfunder* is mainly adopted by business and data scientists for the purpose of cleaning and labelling large data collections for business applications. Unlike *MTurk*, tasks are created using the *Crowdfunder Markup Language (CML)* to define the layout of the elements, which consequently requires task creators to learn a new mark up language in order to post their tasks to the crowd.
- *Prolific Academic* is a crowdsourcing platform for researchers and startup companies with approximately 35,810 workers<sup>2</sup>. The platform is much smaller than other well known platforms, but it provides a similar level of flexibility as offered by *MTurk*, where workers are directed to an externally hosted survey, or dynamic web application.

The majority of crowdsourcing platforms that were investigated only supported static surveys, where the evaluation survey is represented by a series of static web pages constructed using a template web form editing tool. Depending on the requirements of the evaluation task, then a platform that offers a way for the researcher to link to a URL hosting an external web application

---

<sup>1</sup><https://www.mturk.com/>

<sup>2</sup>As of 14 May 2016, see <http://prolific.ac/demographics>



provides more flexibility, by allowing the researcher to develop a more appropriate evaluation tool for their specific needs. *Prolific Academic* provides a unique URL that researchers can present at the end of the task to enable users to collect a payment for their completed submission, which makes it a suitable platform for the evaluation present in this chapter, since an evaluation application can be developed to serve dynamic content to the users, and monitor the quality of the submissions during the evaluation. A well known issue with crowdsourcing is the difficulty in identifying or controlling the overall quality and reliability of the results obtained from the workers [97]. Unlike traditional lab-based evaluations, it is not possible to provide feedback or guidance to users on how the task should be completed. Consequently, identifying poor quality submissions is an important challenge, which is discussed in the next section (Section 7.3).

### 7.3 METHODOLOGY

The motivation for the evaluation is to determine whether the search results generated from the probabilities assigned by the underlying *SLM* to the documents is consistently providing users with a ranking where the top documents address their information need as expressed by a particular query specified in advance. The evaluation consists of a set of fifty individual tasks, where each task is represented by the top- $n$  documents selected from the search results to create a ranked list for each query submitted to the *Samtla* system. Fifty queries are generally considered to be sufficient for obtaining a stable measure of system performance [186]. The queries are representative of the different query types that a researcher of the Bible might submit to the system, including short keyword queries often reflected by the names of individuals, locations, time, and events, to long phrase-like queries representing short “parallel-passages” such as set-phrases and liturgical quotations.

Each assessor assigns a relevance score to each document in the ranked list, according to how well the snippet for each document addresses the topic of the query. Some of the well-known information retrieval evaluations, such as those of the *Text REtrieval Conference (TREC)* [42], assess relevance on the basis of binary relevance judgements, where documents are either relevant or irrelevant. For the purpose of the evaluation presented in the subsequent

sections, a four-point graded relevance scale was adopted. However, by expanding the range of relevance grades available to the assessors, then the resulting evaluation can measure relevance at different degrees, by recording relevance judgements for documents that are marginally relevant to the users information need [119].

### 7.3.1 Data Preparation

The evaluation consists of a ranked list of the top-ten documents for each query submitted to the Samtla system. Users are asked to assign a graded relevance score to each document according to the four relevance grades “not relevant”, “somewhat relevant”, “quite relevant”, or “highly relevant”, with respect to a query displayed at the top of the task.

The queries and documents are generated from the King James Bible version of Samtla, since many of the participants will be familiar with the content of the Bible, to some degree. The set of fifty queries are of variable length, ranging from single word queries (e.g. “Moses”, and “Jesus Christ”), to longer verbose queries representing common phrases (e.g. “the Lord hath spoken”, and “blessed be the Lord”). Two test queries were constructed to control the quality of the users. Each ranking for the query is processed to create two permutations on the ordering of the documents in the ranked list for the query, as follows:

- *Set 1*: represents the *system ranking* for the documents, which are sorted according to the probability for the query, inferred from the *SLM*, as described in Chapter 4.
- *Set 2*: the documents are randomly shuffled to obscure the original *system ranking* so as to reduce the influence of the ranked position of a document on the assessment of relevance by the user, known as a “presentation bias” [59].

For each set of queries we can obtain the original *system ranking*, which is sorted by the probability inferred from the *SLM* for each document, and the *display* order of the documents according to their display position in the ranked list. The *Set 1* queries only have a single order as the *SLM* and the *display* order are equal as they represent the ranking of the documents ordered by

*SLM*. In short, there was no shuffling of the document positions as per the *Set 2* queries. The documents in the *Set 2* queries can be ordered to retrieve the original *system ranking* by ordering the documents according to the probability assigned by the *SLM*, and the *display* order of the documents, by sorting the documents according to the randomly generated rank assigned by the shuffling process. Each user completes ten queries from *Set 1*, and forty queries from *Set 2*.

The evaluation application assigns each new user to one of five bins, and distributes an appropriate set of fifty queries to ensure an even distribution of completed queries from *Set 1*. For example, the users in the first bin will receive their first ten queries from *Set 1*, and the remaining queries from *Set 2*, whereas a user from the second bin will receive their first ten queries from *Set 2*, a set of ten queries from *Set 1*, and their last thirty queries will be from *Set 2*; the process is illustrated in Figure 7.1.

Bin id	q					Display order	
	1-10	11-20	21-30	31-40	41-50	Set 1 (SLM order).	Set 2 (random order).
1	■	■	■	■	■	■	■
2	■	■	■	■	■	■	■
3	■	■	■	■	■	■	■
4	■	■	■	■	■	■	■
5	■	■	■	■	■	■	■

Figure 7.1: Users are assigned to one of five bins, each with their own distribution of queries from *Set 1*, and *Set 2*.

When assessing the system performance, the measures are applied to the *Set 2* queries exclusively, as these results provide a more objective measure due to the fact that the *system ranking* was obscured from the users. If the documents for a query from *Set 1* received a higher relevance score, on average, compared to the *Set 2* equivalent, then the assessors may have been influenced by a presentation bias, resulting in them assigning a higher relevance score to the document when the document appears near the top of the search results for the query.

### Assessing the results

At the end of the evaluation each user will have provided a ranking of the documents according to each query, composed of their relevance judgements, which is referred to as a *user ranking*. All user rankings are aggregated to

create a single *consensus ranking* for each query, which represents the overall “wisdom of the crowd”. The *consensus ranking* is constructed from each query by, first, summing up the relevance score assigned by each user to the documents for each query, where “not relevant” = 1, ..., “very relevant” = 4. The documents are then reranked by sorting them according to their total relevance score, in descending order. The *system ranking*, represented by the probabilities inferred from the *SLM*, is compared to the *consensus ranking* in the following two ways:

1. A system-based evaluation is performed by measuring the quality of the ranking algorithm directly, using the common *NDCG* measure applied to the user relevance scores.
2. A user-based evaluation measures the correlation between the system ranking and the *consensus ranking* to assess users’ satisfaction with the search results through the non-parametric *Spearman’s Footrule* and *M*-measure. If the correlation between both the system ranking, and the individual *user ranking* agrees, on average, with the *consensus ranking*, then the ranking of the documents is on par with human-level performance.

Throughout the remainder of this chapter, when discussing the specific performance measures, let  $r_1$  denote the *system ranking* (either the *SLM* order, or the *display* order of the ranked lists), and  $r_2$ , the *consensus ranking*.

An important part of an evaluation based on human assessments, is to detect the presence of any bias introduced by the design or assumptions of the evaluation. One particular issue that arises when presenting users with ranked lists for assessment, is that users can be influenced by the ordering of the documents, known as a “presentation bias”. This issue can result in users assigning greater relevance to items at the top of a ranked list, regardless of the provided context, since intuitively, our experience of search engines, tells us that the results at the top of a ranked list are generally more relevant. Detecting a presentation bias is achieved by comparing the system ranking, and each *user ranking*, to the *consensus ranking* for both sets of queries (*Set 1* and *Set 2*), when the documents were sorted according to the *display* order). If users are influenced by the presentation order of the documents, then the

users may assign greater relevance scores to the documents contained in the ranked lists of the *Set 1* queries. This could be caused by the fact that the documents are ordered by the *system ranking* generated by the *SLM*, unlike the *Set 2* queries, which are displayed in random order. A presentation bias will be apparent if there is a notable difference between the average scores between the two query sets.

The user interface represents a cut-down version of the Samtla system, where the main search result window has been isolated. Each task is represented by a query displayed at the top of the page, and a ranked list of search results for the query. The documents are displayed with a title and short snippet showing the highlighted terms of the query for the document. Next to each entry in the search results, a drop-down box is displayed with the available relevance grades, which the user selects from, to assign a relevance grade. The user interface performs some basic validation of the results, including updating the timestamp of each response, and identifying missing responses. Once the user has assigned a relevance grade to all documents, a button appears at the bottom of the page to allow the user to progress to the next query, or to the payment screen. The server side of the application validates the results received from the test queries and redirects the user depending on whether they pass or fail. The server assigns successful participants to one of four bins. Each bin represents a different permutation on the order of the query sets presented to the user. The next section discusses the criteria for selecting users for assessing the performance of the system.

### Selecting participants

A participant represents a member of the public, who is not necessarily concerned with the motivation behind the evaluation, and we can not assume that they have had previous experience or competence at the task being presented to them by the researcher. Therefore it is important to prepare for this fact and attempt to filter the crowd of individuals for those who possess the required skills for completing the task. *Prolific Academic* provide a number of filters that enable Samtla system, was that users had to be fluent English speakers.

In addition, not all users will have read, or understood the instructions

detailing the requirements of the evaluation [97]. Furthermore, a select few users may attempt to "game" the system [122], by simply assigning relevance grades at random in order to speed through the tasks to get to the payment screen. It is important to plan for this type of user behaviour, since it is generally not feasible to monitor the performance of users in real time during the evaluation (although see [193] for a description of machine learning approaches for improving the quality of crowdsourced translations). One way to identify these types of users is to incorporate tests at the start of the evaluation. The tests should reflect the tasks that the user will be required to perform. The Samtla system evaluation uses two test queries to filter users, with each test designed to capture one of the two types of user behaviour described.

The first test query contained the top-five ranked documents for the single word query "Satan", which was the basis on which the users had to assess the relevance of each document. The last five ranks contained the top-5 ranked documents from a different much longer query, "chief priests and scribes". To pass the test, the user had to assign "Not Relevant" to the last five documents since they do not match the query "Satan" displayed at the top of the task. This test aims to identify users who have not understood the instructions at the beginning of the evaluation, and users who are assigning relevance scores at random. These types of user behaviour will be apparent from higher relevance scores assigned to the last five documents, which should be marked as "not relevant".

The second test query "Christ Jesus" was composed of the top-ten documents ranked in reverse order of relevance. In order to continue on to the evaluation, the user must assign higher relevance grades to documents as the rank position increases, in otherwords, the user had to assign greater relevance scores to the bottom ranks. This test captures users who are speeding through the task by simply assigning relevance to the documents in decreasing order of relevance, as again, users who are assigning relevance at random.

### 7.3.2 Evaluation Measures

We adopt two sets of non-parametric measures for calculating the system performance. The first set of non-parametric measures evaluates the ranking quality of the system using the *Normalised Discounted Cumulative Gain mea-*

sure (*NDCG*), which is a measure commonly adopted for evaluations based on multiple relevance grades [110]. The second set of non-parametric measures, assesses user satisfaction with the search results by measuring the correlation between the *system ranking* and the *consensus ranking*. If the correlation between the *system ranking* is positive and highly correlated with the *consensus ranking* then we can conclude that the ranking quality of the system is on a par with human-level performance. In the following sections, each of the non-parametric measures is formally defined, before a summary of the final results is presented in Section 7.4.

### Normalised Discounted Cumulative Gain (NDCG)

A popular measure for assessing the quality of ranked search results is the *Normalised Discounted Cumulative Gain (NDCG)*. The *NDCG* is a normalised version of the *Discounted Cumulative Gain (DCG)*, which measures the ranking quality of a system according to the position of the document, and the relevance score assigned by the users. The measure operates with a ranked list of any size, and is one of the few evaluation metrics suited to graded relevance judgements of three or more relevance grades. Unlike other measures, such as those based on binary-relevance, the *NDCG* provides a means for measuring relevance at different degrees, as opposed to binary-relevance judgements, which only describe whether the document is relevant, or not relevant. These measures consequently ignore the fact that some documents, in the search results, may be partially relevant to the user, which is a more realistic assumption [110]. The *DCG* uses a discounting function to model the users interaction with the retrieval system, in the form of a user persistence model. User persistence describes whether the user will continue to look for more documents further down the search results after having seen a certain number of relevant documents. User persistence is often ignored by other performance measures, but can be important for a good performing evaluation metric [77]. There are two commonly adopted discounting functions for the *NDCG*; the first, reduces the contribution of each relevance score according to the document's rank position  $i$ . This models a very impatient user, who will quickly stop searching for relevant documents after the first handful of documents. The second discounting function, which is more popular, discounts the rele-

vance score using the logarithm of the rank  $\log_2 i$ . This user model reflects a more persistent user as the effect of the discounting function is more gradual. This type of user will continue to scan the search results for potentially relevant documents further down the ranked list. Both models of persistence are included for completeness, since they reflect different types of search behaviour. The *DCG* for a ranked list  $r$ , of size  $k$ , is defined as follows:

$$DCG_k(r) = rel_1 + \sum_{i=2}^k \frac{rel_i}{\log_2 i} \quad (7.1)$$

where  $rel_i$  is the relevance score at position  $i$ , and  $\log_2 i$  is the discounting function, which may also be substituted for  $i$  to discount the relevance score by rank position.

To illustrate, assume a ranking of five documents  $D_1, D_2, D_3, D_4, D_5$ , then the *DCG* for a ranked list of size  $k=5$ , results in the following table of gains for each rank: If we apply (7.1) to the result of the logarithmic discounting

$i$	$rel_i$	$\log 2_i$	$\frac{rel_i}{i}$	$\frac{rel_i}{\log 2_i}$
1	4	0	4	0
2	3	1	1.500	3
3	4	1.585	1.333	2.523
4	2	2.000	0.500	1
5	1	2.322	0.200	0.430

Table 7.1: Calculating the *DCG@5* for each rank.

function for the *DCG*, in Table 7.1, then we have:

$$DCG_5(r) = rel_1 + \sum_{i=2}^5 \frac{rel_i}{\log 2_i} = 4 + (3 + 2.523 + 1 + 0.430) = 10.953 \quad (7.2)$$

The resulting *DCG* for this ranking is then normalised to obtain the normalised *DCG* (*NDCG*) through:

$$NDCG_k = \frac{DCG_k}{IDCG_k} \quad (7.3)$$

where the value of the *DCG* is divided by the *Ideal Discounted Cumulative Gain* (*IDCG*), representing the best possible ranking for the documents according to the relevance scores assigned by the human assessors. The *IDCG* is calculated by first sorting the documents in descending order by relevance score, for example, assuming the same set of five documents  $D_1, D_2, D_3, D_4, D_5$ ,



and a list of corresponding relevance scores e.g. 4, 3, 4, 2, 1, then the best possible ranking of the documents is  $D_1, D_3, D_2, D_4, D_5$ . Next, we calculate the  $DCG$  for this ranking to obtain the maximum possible value of the  $DCG$ . The  $IDCG$  is then used to normalise the  $DCG$ , which enables the measure to be applied to ranked lists of variable size [110] and across multiple queries and users. In our case the ranked lists are the same size, but the full formulation of the  $NDCG$  is adopted since this is the commonly adopted by the information retrieval community. The  $IDCG$  for this ranking is then:

$i$	$rel_i$	$\log 2_i$	$\frac{rel_i}{i}$	$\frac{rel_i}{\log 2_i}$
1	4	0	1	0
2	4	1	2.666	4
3	3	1.584	2.250	1.892
4	2	2.000	4	1
5	1	2.321	5	0.430

Table 7.2: Calculating the  $IDCG@5$  for each rank.

$$IDCG_5(r) = rel_1 + \sum_{i=2}^5 \frac{rel_i}{\log 2_i} = 4 + (1.892 + 1 + 0.430) = 11.322 \quad (7.4)$$

The  $NDCG$  for this ranking is then derived by dividing the  $DCG$  by the  $IDCG$  as follows:

$$NDCG_5(r) = \frac{10.953}{11.322} = 0.967 \quad (7.5)$$

The resulting  $NDCG$  is quite close to the  $IDCG$ , and so we can say that the obtained ranking is very close to that of human-level performance. All that was required to improve the  $NDCG$  score, was to swap the positions of the documents at rank two and three to achieve a perfect ranking. A measure of the ranking quality of the system is obtained by computing an average  $NDCG$  by query and user. The query average is obtained by summing the  $NDCG$  for each *user ranking*, for the given query, and then dividing by the number of users who completed that particular the query. Next, we sum up the average  $NDCG$  score for each query, and divide by the total number of queries. The process is repeated for the average user  $NDCG$  by summing up the  $NDCG$  score for each query according to the given user, and then dividing the resulting sum by the number of queries completed. If the final query average is high, then the quality of the ranking generated by the system is close to the  $IDCG$ , which suggests that the ranking quality of the system

is close to what would be considered human-level performance. A baseline for the *NDCG* is computed for the purpose of comparison with the final average. This is achieved by simulating the random input of 1000 users for the set of forty queries. With each iteration, we generate a ranked list of results for each user and query by assigning a random value representing a graded relevance score, for each of the top-ten rank positions. In effect, the process simulates a very poor performing user who simply assigns random relevance scores, in other words we obtain a *pseudo-user ranking* for each query. The baseline figure is then calculated by averaging over the *NDCG* scores for each *pseudo-user ranking* for the given query, and then a final average is calculated from the average *NDCG* for each query.

### Non-parametric correlation measures

The *NDCG* measures the ranking quality of the system, however, we also wish to know whether the ranking also satisfies the users opinion of which documents are relevant to the query. This is achieved by measuring the correlation between the system ranking and a ranking generated from the users relevance scores. The non-parametric correlation measures adopted for this purpose are the *Spearman's footrule* [90] and the *M-measure* variant [60]. These non-parametric measures describe the degree of correlation between two ranked lists, and provide similar results to other non-parametric correlation measures including Spearman's  $\rho$  and Kendall's  $\tau$  [92]. In our case the two ranked lists are represented by the *system ranking*  $r_1$ , and the user *consensus ranking*  $r_2$ . We discuss each of the non-parametric correlation measures in more detail, where *Spearman's footrule* is abbreviated to simply *Footrule* throughout the rest of the discussion.

The *Footrule* is calculated by summing the result of the absolute differences between the rank positions of the documents for each individual ranked list. The *Footrule*, denoted by  $Fr$ , is more formally defined as follows:

$$Fr(r_1, r_2) = \sum_{i=1}^k |(r_1(i) - r_2(i))| \quad (7.6)$$

where  $r_1$  and  $r_2$  are two ranked lists assumed to contain the same set of documents, and  $k$  is the size of the ranked list, in our case  $k = 10$ , which

represents the top-10 ranked documents. In order to use the *Footrule* as a metric, we need to normalise the result by calculating the maximum possible value, through:

$$F = 1 - \frac{Fr(r_1, r_2)}{\max Fr(k)} \quad (7.7)$$

where  $\max Fr$  represents the maximum value, which when  $k$  is an even number  $\max Fr = \frac{1}{2}k^2$ , and if  $k$  is an odd number then  $\max Fr = \frac{1}{2}(k+1)(k-1)$ . This ensures the resulting *Footrule* falls in the range of 0 and 1 where a value close to 1 means that the two ranked lists are highly similar.

When evaluating search results, however, we may wish to consider the fact that documents in the top ranks are often considered the most relevant to the users information need than documents appearing in lower ranks [60]. For this purpose, the *M*-measure was adopted, which is designed to assign more weight to ranked lists containing identical, or near-identical, sets of documents at the very top of the ranked lists. The original measure defined in [60], accounts for the situation where the ranked lists are different sizes, or where the set of documents in  $r_1$  and  $r_2$  are different. For the purposes of the evaluation presented in this chapter, the ranked lists are the same size and contain the same set of documents. As a result, the *M*-measure is reduced to the following form:

$$m(r_1, r_2) = \sum_{i=1}^k \left| \frac{1}{r_1(i)} - \frac{1}{r_2(i)} \right|, \quad (7.8)$$

where we calculate the sum of the absolute difference between the rank position of the document in the given ranked list. Next, we calculate the maximum possible value,  $\max M$ , given a ranked list of size  $k$ :

$$\max M = \sum_{i=1}^k \left| \frac{1}{i} - \frac{1}{k-i+1} \right| \quad (7.9)$$

The maximum value for  $m(r_1, r_2)$  is used for the normalisation step, in order to obtain a metric, with values falling in the range of 0 and 1. The normalisation involves dividing the value of  $m(r_1, r_2)$ , by the maximum value  $\max M$  in (7.9), and subtracting 1 from the result.

$$M = 1 - \frac{m(r_1, r_2)}{\max M(k)} \quad (7.10)$$

To illustrate, the difference between the *Footrule* and the *M*-measure, consider the following simple example, where we assume two ranked lists of documents  $r_1$  and  $r_2$  illustrated in Table 7.3: The *Footrule* correlation for these

$i$	$r_1 i$	$r_2 i$
1	$D_4$	$D_3$
2	$D_3$	$D_4$
3	$D_1$	$D_1$
4	$D_2$	$D_2$
5	$D_5$	$D_5$

Table 7.3: Two ranked lists  $r_1$  and  $r_2$ , with a document  $D_4$  at rank 1 of  $r_1$ , and rank 2 of  $r_2$ .

two ranked lists is  $F = 0.833$ . The resulting *Footrule* for the ranked lists is positive and highly correlated, which shows that the order of the documents in the ranked lists are very similar. However, the *M*-measure for these ranked lists is  $M = 0.655$ , which is much lower than the *Footrule* as it penalises highly relevant documents appearing in later ranks, specifically the drop in rank of  $D_4$  to rank 2 in  $r_2$ . In order to obtain a measure of the users' satisfaction with the search results, we calculate an average for each of the measures over all the obtained *user rankings*. First, for each query, we sum up the correlation scores (*Footrule* or *M*-measure) for each of the *user rankings* compared to the *system ranking*, which is then divided by the number of users for the query. Next, we sum over the average query correlation scores, and divide the result by the total number of queries.

A further process measures the correlation between each individual *user ranking* and the *consensus ranking*, which describes the average agreement between each user and the opinion of the crowd. This is achieved, by first, iterating over each user and extracting the *user ranking* for the specified query, and then calculating the correlation between the *user ranking* and the *consensus ranking* generated from all *user rankings* for the query. The average correlation is then the sum of the correlation scores for each query divided by the number of queries. A baseline for each measure is also established for comparison, by calculating the *Footrule* and *M*-measure between the *SLM* order and the *display* order for each query, before taking an average over all

queries. The baseline describes how much correlation already exists between the two query sets, in other words how successful the shuffling process was in randomising the order of the documents for each query in the *Set 2* queries.

### Significance testing

Measuring the statistical significance of the results is achieved by adopting the *bootstrap* method [87, 169, 176], which attempts to approximate the original underlying distribution of the population, by selecting a series of random samples of size  $N$  with replacement from the observed population data. An advantage of the bootstrap method is that it is compatible with any statistical measure [176], meaning we can use the correlation and *NDCG* scores as our test statistics. Under the bootstrap method, we assume that the null hypothesis is that there is no difference between the ranking generated by the system and the ranking generated by the user evaluations. The difference is considered significant, with respect to the stated significance level, if the confidence intervals do not overlap. In order to obtain the confidence intervals, a series of samples are generated by selecting a value at random from the original distribution of correlation or *NDCG* scores for the query (e.g. *Footrule* or *M-measure*, and *NDCG*). Each sample is equal in size to the number of queries (or users depending on the analysis) in the original evaluation.

The sampling process can be thought of as extracting values from the rows and columns of a  $n$  by  $m$  matrix, where the rows contain the correlation or *NDCG* scores by query, and the columns represent the per user scores. Each random sample  $b$ , where  $b = 1, \dots, B$ , is composed of values selected with replacement. We perform this operation for a total sample size of  $B = 1000$  and calculate the average of the test statistic for each sample. Calculating the final confidence interval then involves sorting the averages in ascending order, and selecting the values that fall at the  $B(1 - (\alpha/2))$  and  $B(\alpha/2)$  percentile, where  $\alpha$  is the required significance level and  $\alpha = 0.05$  represents a 95% confidence interval. We take an average over the lower and upper bounds of the confidence intervals and partition the results by query, *user agreement* (see Section 7.4).

## 7.4 EVALUATION RESULTS

The evaluation was attempted by a total of 65 participants. Ten users were excluded from the results due to incomplete submissions resulting from connection timeout issues. A further thirty-one users were removed due to failing the test queries, which is almost half of the submissions received. Out of the thirty-one who failed, ten users failed to pass the first test query, which means that 33% of the users were unable to identify that the search results of the first test query contained documents from two completely different queries (one keyword query versus a long verbose query). These results highlight the importance of designing tests as part of an evaluation in order to filter potentially poor performing users upfront. In the end, a total of twenty-four participants successfully completed all evaluation tasks. The successful participants are summarised in Figure 7.2 according to information provided by the users at the start of the evaluation: The majority of the submissions were

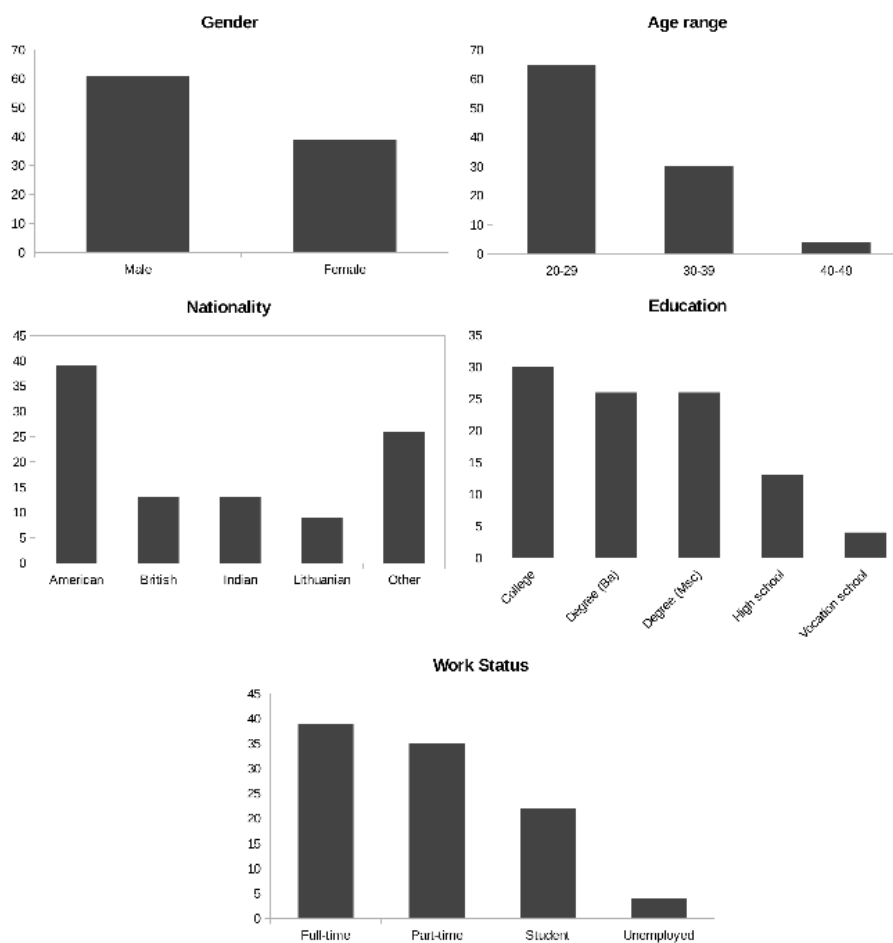


Figure 7.2: Summary statistics on the successful participants by demographic.

received from men between the ages of 20-29 years, and resident or born in North America. Furthermore, the participants were mainly degree educated, with an even split between Bachelor's degree and postgraduate level. The majority of the users were also working either part-time or full-time. This suggests that for these users, crowdsourcing provides a way to supplement their income, or for pure interest.

### Normalised Discounted Cumulative Gain (NDCG)

The average baseline figures for each discounting function are presented in Table 7.4 below. The average *NDCG* for the baseline is quite close to the ideal ranking (*IDCG*). The logarithmic discounting function  $\log_2 i$  being slightly less aggressive than the discounting by rank position  $i$ . The average the *NDCG* for the *SLM* ranking and the *display* ranking are presented below (see Table 7.5 and Table 7.6). We make a distinction between the *display* ranking

Baseline <i>NDCG</i>	
$i$	$\log_2 i$
0.8514	0.8686

Table 7.4: Baseline *NDCG*

for the *Set 1* queries and the *Set 2* queries and report the average *NDCG* by query and user with their 95% confidence intervals presented alongside in square brackets.

From the results, it appears that the users tended to assign higher relevance

NDCG@10	SLM rank order	
<i>Set 1</i> queries (10)	$n$	$\log_2 i$
Query	0.9852 [0.9849, 0.9855]	0.9876 [0.9873, 0.9878]
<i>Set 2</i> queries (40)	$n$	$\log_2 i$
Query	0.9814 [0.9811, 0.9818]	0.9838 [0.9835, 0.9841]

Table 7.5: Average query *NDCG* for each set of queries according to the *SLM* rank order of the document with the 95% confidence intervals reported in square brackets.

to the top documents in the search results, as shown by the average *NDCG* score being very close to the *IDCG*. This is also supported by the higher average relevance scores that were assigned by users at the top ranks of the search results, as illustrated in Figure 7.3. Here we see that users assigned slightly higher relevance, on average, to the top documents in query set *Set*

1, which were displayed according to the score assigned by the *SLM*, versus the documents in query set *Set 2*, which were displayed in random order. The *NDCG* is higher for the *Set 1* queries than the *Set 2* queries, suggesting that users assigned higher scores to the top documents due to the presence of a presentation bias. Consequently, *Set 1* queries are removed from any further analysis.

Concentrating now on the *Set 2* queries, and the average *NDCG* scores for the *SLM* rank order for the documents, the average *NDCG* is quite close to the *IDCG* with  $n=0.9814$ , and  $\log_2 i=0.9838$ . Comparing these figures to the the baseline *NDCG* (Table 7.4), and the *NDCG* for the *display* rank order of the *Set 2* queries (Table 7.6), there was less of a presentation bias, as indicated by the relatively low *NDCG* of  $n=0.8819$ , and  $\log_2 i=0.8951$ . This suggests that the users were not influenced by the presentation order of the documents as much as they were when viewing the *system* ranking of the documents, represented by the *SLM* rank order of the documents for the query. The non-overlapping 95% confidence intervals from the *bootstrap* means that the results are significant at the  $\alpha = 0.05$  level.

A further observation is that the users assigned higher relevance grades to the documents appearing in the top ranks of the *Set 2* queries, which is reflected by the higher average *NDCG* score, across the discounting functions, for the *SLM* rank order in Table 7.6). However, the baselines figures for the *NDCG* are quite high ( $n=0.8514$ , and  $\log_2 i=0.8686$ ), and if we were to discount the baseline *NDCG* scores from the average *NDCG* for the *display* order of the queries, the effect of the presentation order on the judgement of relevance would actually be less pronounced.

NDCG@10	Display rank order	
<i>Set 2</i> queries (40)	$n$	$\log_2 i$
Query	0.8819 [0.8811, 0.8828]	0.8951 [0.8943, 0.8960]

Table 7.6: Average *NDCG* for the *Set 2* queries according to the *display* rank order of the documents, with the 95% confidence intervals reported in square brackets.

### Non-parametric Correlation measures

To recap, the non-parametric correlation measures assess the level of satisfaction with the search results generated by the system. In this section, the



average correlation scores are presented, which report how close on average the *system ranking* is to the “gold-standard”, represented by the *consensus ranking*, and how each *user ranking* compared to the “wisdom” of the crowd. Firstly, the baseline for each measure is established and reported in Table 7.7, below. The baseline correlation for the *Set 1* queries is 1.000 since the *dis-*

Baseline <i>Correlation</i>		
<i>Type</i>	<i>Footrule</i>	<i>M-measure</i>
<i>Query</i>	0.4000	0.3687

Table 7.7: Average query baseline correlation for each measure, which compares the *SLM* rank order of the documents to the *display* rank order for *Set 2* queries only.

*play* and *SLM* rank order are equivalent, and therefore it is not included in Table 7.7. In terms of the baseline for the *random* order queries, we can see that it is quite low across the two measures, but does show that there is some correlation despite the shuffling process. The final results for each measure are displayed in Table 7.8 and Table 7.9 where we present the average *Footrule* and *M-measure* for the *SLM* and *display* ranking compared to the user *consensus ranking*, respectively. For each form of analysis, we divide the results into query, user, and user consensus averages, and report the 95% confidence interval in square brackets, obtained from the bootstrap (see Section 7.3.2). The results of Table 7.8 show that the user relevance judgments for the *Set 1*

	SLM rank order	
<i>Set 1</i> queries (10)	<i>Footrule</i>	<i>M-measure</i>
Query	0.7739 [0.7719 - 0.7758]	0.8382 [0.8363 - 0.8401]
User consensus	0.8003 [0.795 - 0.798]	0.8474 [0.845 - 0.848]
<i>Set 2</i> queries (40)	<i>Footrule</i>	<i>M-measure</i>
Query	0.7558 [0.7542 - 0.7573]	0.7613 [0.7594 - 0.7632]
User consensus	0.7173 [0.7159 - 0.7188]	0.7370 [0.7353 - 0.7386]

Table 7.8: Average query and user consensus correlation scores for the *SLM* rank order of the documents, divided by query set.

queries are positive and highly correlated with the *SLM* rank order of the *Set 1* queries (*Footrule*=0.7739, *M-measure*=0.8382). The average query correlation for the *SLM* rank order of the *Set 2* queries is also positive and highly correlated (*Footrule*=0.7558, *M-measure*=0.7613), but slightly less so than the *Set 1* queries, which implies that users were influenced by the presentation order of the documents when they were viewing the *SLM* rank order of the

query results, compared to the equivalent query in the *Set 2* queries, where the documents were assigned a random rank position. This is supported by the average *M*-measure (*Set 1*=0.8382, *Set 2*=0.7613), which is higher for the *Set 1* queries, suggesting that the users assigned higher relevance grades to the top documents in the search results, when judging relevance according to the *Set 1* queries in *SLM* rank order. Consequently, *Set 1* queries are discarded from the analysis due to the existence of a presentation bias, and the rest of the discussion will focus on the relevance judgements obtained for the *Set 2* queries. Returning to the results in Table 7.8, the average query correlation scores for the *Set 2* queries, are positively correlated with the *consensus ranking* (*Footrule*=0.7558, *M*-measure=0.7613). Furthermore, the slightly higher score for the *M*-measure, suggests that the users were assigning higher relevance grades to a small selection of documents appearing in the top ranks of the search results.

In addition, looking at the average correlation between the *user ranking* and the *consensus ranking* for each query (*Footrule*=0.7173, *M*-measure=0.7370), suggests that the users agreed on average with crowd opinion with respect to the order and relevance of the documents for each query. To determine whether the results are reliable, the *Set 2* queries are reordered according to their *display* rank order, in order to assess whether there is any presentation bias in the results. The correlation scores for the *display* ranking are presented in Table 7.9, which summarises the average for the *Footrule* and *M*-measure for the *Set 2* queries by query and the average agreement between each *user ranking* and the *consensus ranking*. The correlation scores in Table 7.9 are

	Display rank order	
<i>Set 2</i> queries (40)	<i>Footrule</i>	<i>M</i> -measure
Query	0.3992 [0.3971 - 0.4014]	0.4722 [0.4699 - 0.4744]
User consensus	0.4374 [0.4360 - 0.4388]	0.4177 [0.4165 - 0.4189]

Table 7.9: Average query and user agreement correlation scores for the *Set 2* queries in *display* rank order.

much lower than the average correlation for the *SLM* rank order presented in Table 7.8, across the measures (*Footrule*=0.3992, *M*-measure=0.4722). This suggests that there was less of a presentation bias than that present in the *Set 1* queries, which is understandable given the random presentation order of the documents.

It would appear that the users were attempting to do a good job by judging relevance according to the information provided by the snippet text of each document, as opposed to assigning relevance as a function of the document position. The average correlation scores are still rather high, suggesting that there is a presentation bias even for the *display* order of the *Set 2* queries. If we consider the degree of correlation that already existed between the two permutations on the order of documents in the two query sets ((*Footrule*=0.7558, *M-measure*=0.7613) in Table 7.7), then it could be argued that the correlation is less pronounced.

On the basis of the performance measures, presented above, we can say that users were highly correlated with the *SLM* order of the queries than the *display* order when analysing the results of the *Set 2* order queries independently of the *Set 1* queries. The users were more influenced by the presentation order of the *Set 1* queries, in the sense that they were slightly more generous with their relevance grades; where users tended to assign higher relevance scores to a select few documents at the very top of the search results. This was observed from the high positive correlation scores for the *M-measure*, and the *NDCG*.

To assess whether this is the case, we can compare the average relevance grade assigned by the human assessors to each rank position over both query sets according to the *SLM* ranking for the documents. Figure 7.3, shows the average relevance score assigned to each rank of the search results, the plot shows that the human assessors on average gave higher relevance scores to the top five ranks when they completed the *Set 1* queries, which were in the order ranked by the system. To summarise, there is an observable difference in the non-parametric correlation measures with respect to the scores assigned to the *SLM* rank order and *display* rank order of the *Set 2* queries. The non-parametric correlation measures are positive and highly correlated to the *SLM* rank order of the documents, which represents the *system ranking* of the queries generated by the infrastructure supporting the *Samtla* search engine. This suggests that crowd of users agreed with the ranking generated by the *SLM*, as indicated by the high and positively correlated scores between each individual user ranking and the consensus ranking generated from all *user rankings*. Therefore, the null hypothesis can be rejected, as there is a strong correlation between the ranking generated by the *SLM* and the user generated

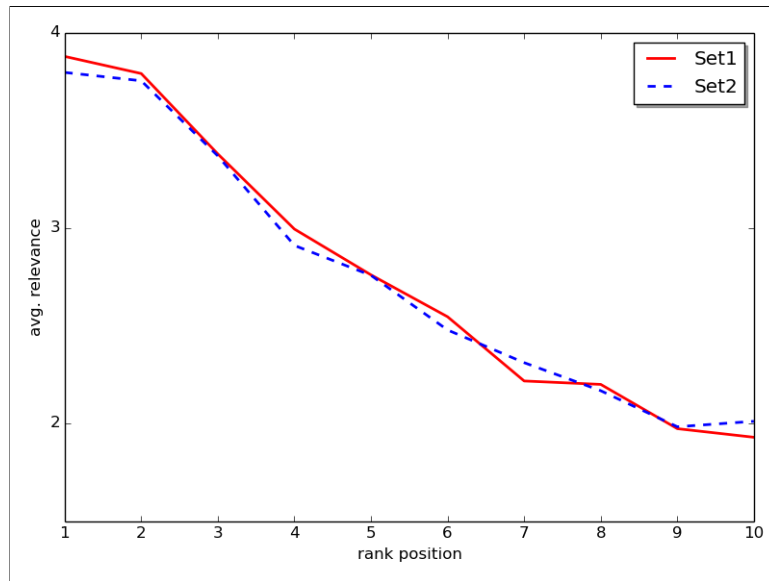


Figure 7.3: Average relevance score by rank position divided by query set.

ranking reflected by the *consensus ranking*.

Out of the fifty queries completed by each user, 80% of them were presented in random order, yet the users consistently assigned more relevance to the documents that received the highest document score according to the underlying *SLM*, and we can see that these scores are not the result of users assigning relevance at random, or "gaming" the system, in part due to the role played by the quality assessment represented by the test queries. Users also revisited their earlier relevance assignments, when they encountered highly relevant documents at the bottom of the result page, caused by the random shuffle process, which was evident from the time stamps assigned to the document in the search results after each relevance grade was assigned. Therefore, there is significant evidence to suggest that users were attempting to do a good job and were not assigning relevance grades purely at random, but based on what they considered to be relevant given the provided query context.

### Query performance

The query averages for each measure provide a good basis for exploring how the choice of query has an impact on the relevance judgments assigned by the users. Although the queries were rigorously selected to ensure a balance between content, length, and number of highly relevant documents in the search results, some queries may have performed better than others. Furthermore, identifying the poor performing queries may help to provide more insight in

to how the non-parametric measures were arrived at, and to inform future evaluations of system performance. Looking at the distribution of relevance grades, it appears that users did not make full use of the grades when assigning relevance to the documents (see Figure 7.4). For example, some users preferred to adopt a binary relevance approach where they only assigned a relevance according to the grades “very relevant” and “not relevant”.

In terms of query length, the shorter queries tended to have more relevant documents in the top-ten results. Under this scenario, users appear to judge relevance on the basis of the number of terms highlighted in the snippets. On the other hand, the longer verbose queries contained an average of three to four “very relevant” documents, with the remaining results presenting partial matches to the query. For example, given a query “As the Lord commanded”, the documents containing a full match, naturally received higher relevance grades than the partial match “...As thy Lord commanded”. Likewise, the query “the angel of the Lord appeared” also contained partial matches for “the angel of the Lord appeareth”. The users tended to assign less relevance to partial matches, but this is user-dependent. For instance, the users represented a “general” user, that is, one who does not necessarily have any interest or knowledge of the archive underlying the evaluation. However, a humanities researcher interested in the Bible may find the the partial matches for the query partially relevant to their information need, or at the very least, may find these partial matches an interesting example worth including in their summary of the research topic, as these partial matches represent a rare form of the phrase or passage. Plotting the average correlation and *NDCG* scores for each query shows that some queries did perform poorly compared to others. From the resulting plots (Figure 7.5 and Figure 7.6) we can see that there were approximately nine queries that performed particularly poorly, where the *Footrule*, or *M-measure* was  $\leq 50$ .

The worst performing query for the *NDCG* measure (with discounting by  $\log_2 i = 0.86$ ) was number 30 “the ark of the god of israel”. The search results contained one known-item for this query “1 Samuel chapter 5”, which contained four full matches for the query in the snippet text. There was then one further document with a full match, and the remaining eight documents contained partial matches of variable length. This query received the most

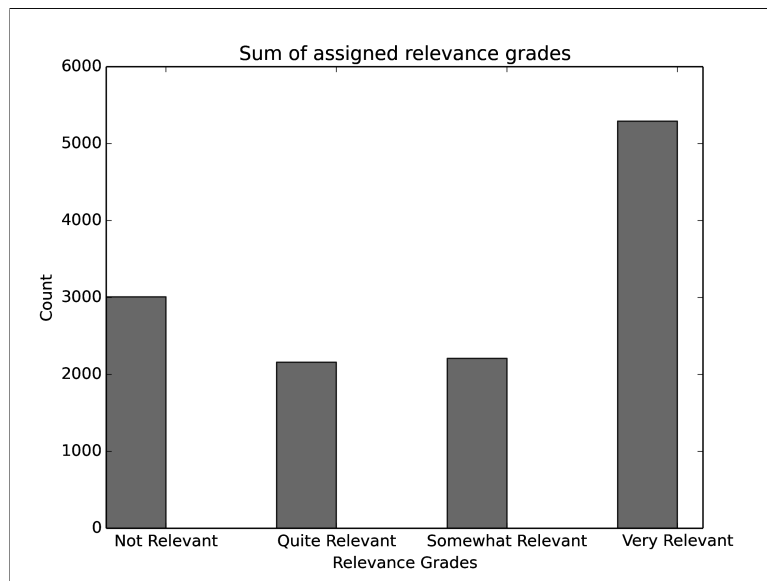


Figure 7.4: Distribution of relevance grades used in the evaluation.

“very relevant” scores from users of all the queries, with the remaining relevances distributed in descending order, ending with only two “not relevant” documents.

Looking at the average scores across the correlation measures, the worst performing query according to the *Footrule* was number 40 “the name of the Lord” ( $Footrule=0.24$ ). The search results for this query contained one document with three full matches for the query. This was followed by a further five documents with two full matches, and the remaining documents contained only a single match, with some partial matches distributed in the same snippet text. This query received the second most “very relevant” scores from users for the single document containing three full matches for the query. Users appear to have disagreed over the best relevance grade to assign the five documents containing two full matches, since aside from the context provided by the snippet, the documents could all be considered more or less relevant when considered in isolation.

For the  $M$ -measure query number 42 “on the seventh day” performed quite poorly ( $M$ -measure=0.46). The users were presented a choice between one document with four full matches, two documents with three full matches, a further two documents with two full matches, and the remaining documents contained only a single full match for the query. Once again, the users disagreed with the best choice of relevance for the five results containing only a single match for the query, which represented half of the search results. The

poor performing queries were distributed evenly across different query lengths, but the pattern that emerges is that users do well at assigning relevance to the documents which contain a large number of full matches. However, when the user is faced with a choice between a select few documents containing the same number of full matches for the query, for instance, those documents with only one full match in the examples, there was more disagreement between the users.

Furthermore, when it came to the documents with only partial matches, the users tended to adopt one of two strategies. The first, assigned partial relevance to the documents according to the total number of partial matches or breadth of the match in the snippet text. And the second approach involved assigning “not relevant” to any document with a partial match for the query. With no other context aside from the provided snippets, it is understandable that there is little agreement between the users for these types of query matches. However, it is possible, that a researcher who is familiar with the Bible might be able to choose between the documents containing similar matches, based on any previous knowledge they may have acquired about the document content. The document prior introduced in Chapter 5, may be useful in this context, by providing additional information about the documents when producing the search results.

The subjective notion of relevance is supported by anecdotal evidence in the form of email enquiries received from the users after completion of the evaluation. It appears that some users adopted quite unexpected strategies for assessing relevance, as illustrated by the following enquiry:

Should passages which refer to a person, but that person is not named in the snippet, be marked as very relevant? For example, the topic is *Moses spake unto the children* and one of the results is *...came out. And he came out, and spake unto the children of Israel that which he was. He* refers to Moses. Am I correct in assuming that should be considered very relevant?”

This is indeed a fair question, since the snippet is obviously semantically similar to a number of other snippets in the search results that do explicitly mention the name “Moses”, which is part of the query. In this instance, the pronoun makes reference to “Moses”, however, from an information retrieval

perspective this document would be ranked lower as it represents only a partial match to the full query “Moses spake unto the children”.

## 7.5 DISCUSSION

Crowdsourcing has its challenges, in particular, the researcher has little control over the evaluation process once it is launched and available online. Therefore, it is necessary to consider the use of test queries in order to filter out bad users upfront e.g. those who have not understood the task, or do not have the correct attitude.

This increases the quality of the submissions, and mitigates against issues that can arise, such as an unhappy user as a result of a rejected submission, or withholding payment due to a suspect submission. These issues can be difficult to resolve and may have an impact on your reputation, and consequently on whether you will be able to submit future evaluations with the same crowd sourcing platform.

The design of the evaluation should record data that permits the testing of a display bias, since some users may assign relevance to document in the top ranks without necessarily digesting the snippets fully. This is easily achievable by randomising the order of the queries. It is also worth recording a timestamp for each response. This enables the researcher to check for users who are speeding through the evaluation at a rate that exceeds the ability to comfortably digest the information related to the task. We found that users assigned relevance at an average rate of three seconds per rank position. The minimum time taken was one second, which we could argue is not enough time to digest the snippet and then navigate to the drop-down box to select a relevance grade. The maximum time to select a relevance grade was thirteen minutes, but this is likely the result of users being interrupted or distracted from the task.

To summarise, the chapter presented a novel approach to system evaluation in IR, by adopting a crowdsourcing approach. The chapter briefly described the concept behind crowdsourcing (Section 7.2), before discussing the methods and non-parametric measures selected for evaluating the system’s performance. The non-parametric correlation and *NDCG* measures provide a good basis for assessing the performance of an information retrieval system.



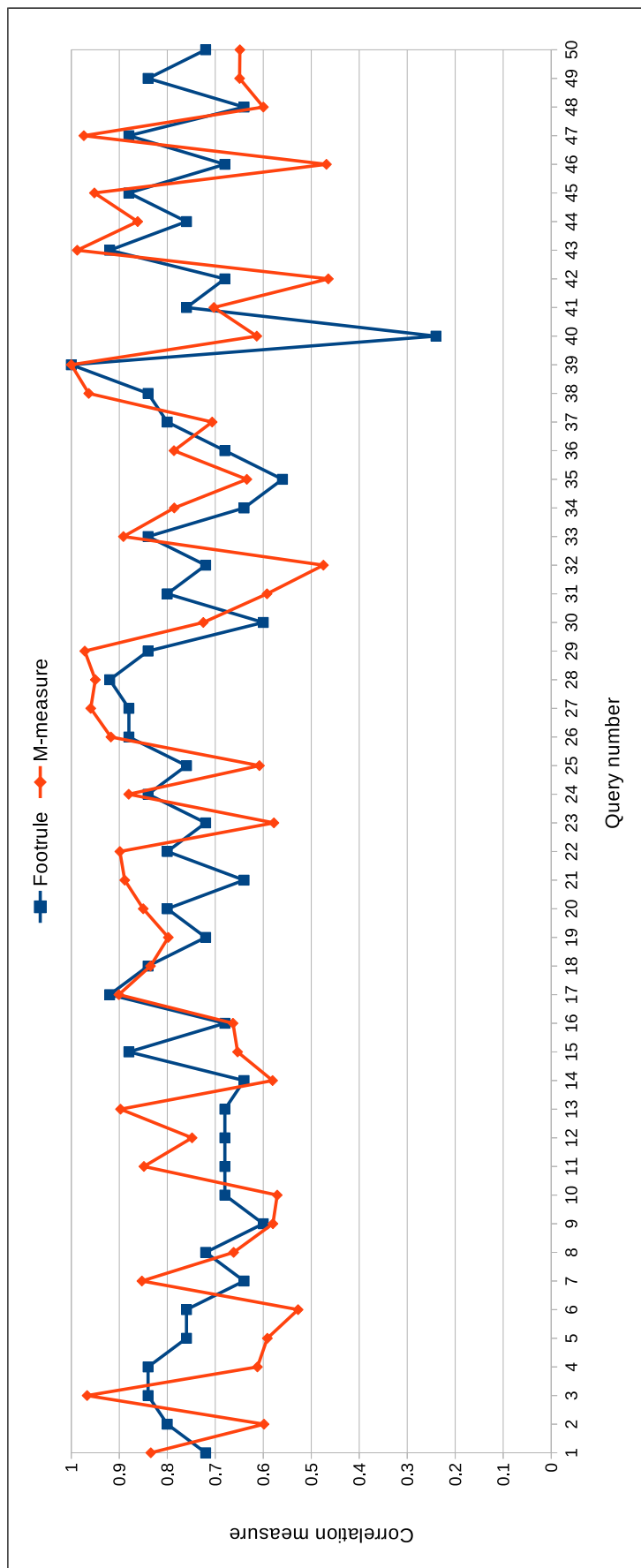
The non-parametric correlation measures represented by the *Footrule* and the *M*-measure (described in Section 7.3.2), measured how close the *system ranking*, represented by the *SLM*, was very close to the *consensus ranking*, or crowd opinion, obtained from the user relevance scores.

The *NDCG* measure, discussed in Section 7.3.2, assessed the ranking quality of the search results, which demonstrated that the *SLM* consistently produces a ranking where the top-ranks of the search results are occupied by the most relevant documents. Furthermore, these results were also supported by the level of agreement between the individual *user ranking* and the *consensus ranking* representing the crowd opinion. The significance of the results was measured through the *bootstrap* method presented in Section 7.3.2, which is a non-parametric approach for measuring significance of the results, and has been found to be competitive with other significance tests [169]. The advantage of the *bootstrap* approach is that it is relatively simple to implement, and flexible to the test statistic. Section 7.4 presented an analysis of the results obtained from the evaluation. The results demonstrate that the *SLM* consistently places the most relevant documents at the top of the search results, for a range of query types represented by both short keyword queries containing one or two terms, to long phrase-like queries of three or more terms.

Crowdsourcing as a platform for system evaluation provides a unique opportunity for researchers to gain access to a large group of participants from a diverse range of social and economic backgrounds. This is one of the things that can also make crowdsourcing a challenge as it becomes necessary to put filters and controls in place in order to guarantee the a good standard of quality in the human assessments. Therefore, the evaluation should be designed to include a minimal and accessible user interface to increase usability and enable the user to complete the task with the minimum of distractions. Furthermore, to minimise poor quality results, log data should be recorded on the users interaction with the evaluation application, which can be as simple as a time stamp, in order to spot suspect users who may be attempting to “game” the system.

In conclusion, the performance of the infrastructure, represented by the *SLM* stored in the suffix tree data structure, is very close to that of human-level performance when applied to information retrieval tasks. Quantitatively

evaluating the text mining tools is not as straight forward as that adopted for information retrieval. First, there does not appear to be any standard performance measures or benchmarking processes for text mining tools. Furthermore, each tool would require an independent assessment with its own set of assessment criteria, user interface design, and evaluation measures. Nevertheless, the text mining tools are constructed from the same *SLM* component as the one evaluated in this chapter, and the tools were designed with alongside researchers presented as case studies (see Chapter 3), who empirically assessed the output of the tools as part of a collaborative development process.

Figure 7.5: Average correlation by query partitioned into *Footrule* and *M-measure*.

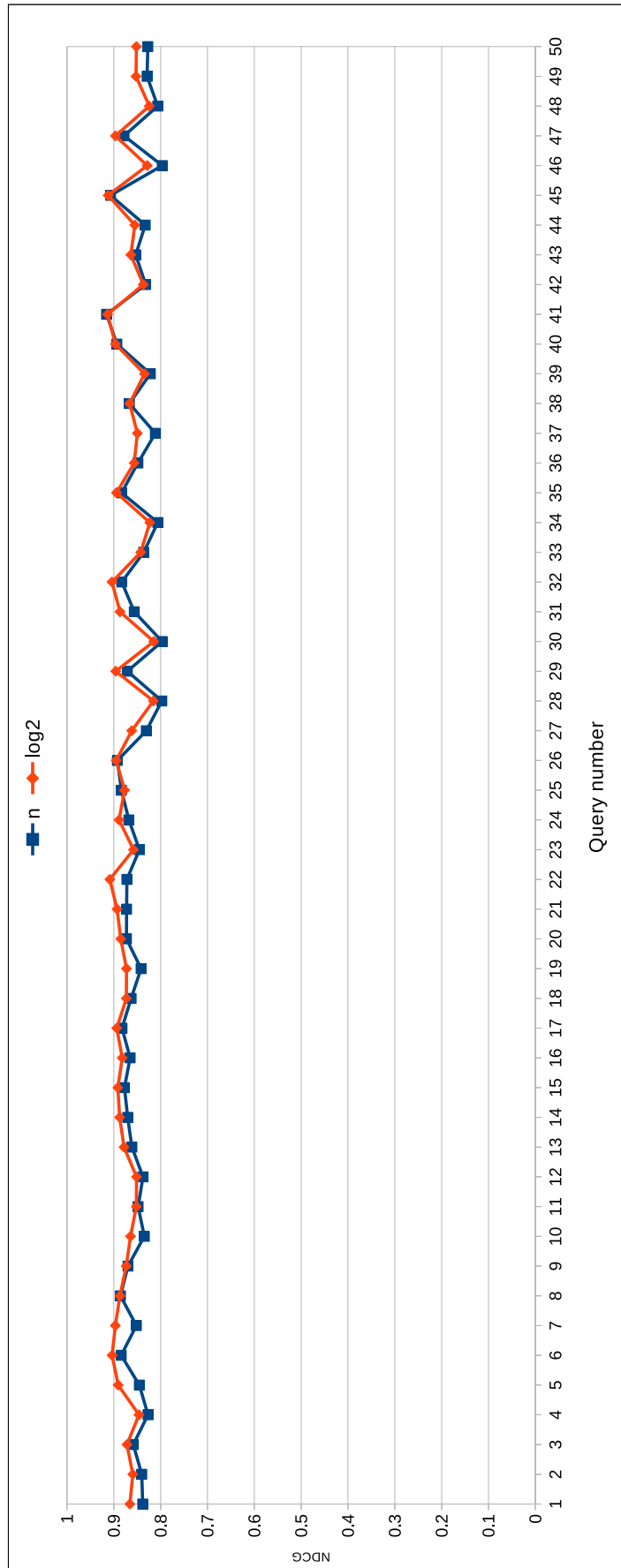


Figure 7.6: Average *NDCG* by query divided according to discounting function.

## CHAPTER 8

# CONCLUSIONS AND FUTURE WORK

This chapter, summarises the research contributions presented in the thesis. A summary of the thesis is presented in Section 8.1, followed by an overview of the main contributions in Section 8.2. The thesis concludes with a discussion of the prospects for future research and development of the proposed infrastructure and the *Samtla* system in Section 8.3.

### 8.1 SUMMARY OF THE THESIS

Chapter 1 described the current issues faced by humanities researchers who wish to make use of the increasing availability of digital archives, but lack the tools to perform the required analysis. The volume of digitised material available to researchers is now on such a scale that it is becoming increasingly difficult to comprehensively analyse the material using manual or traditional approaches such as close-reading, and annotation of the texts. There is now a genuine need for digital tools to support the analysis of digitised documents, but also the much larger volume of “born-digital” material published online everyday. Tools and systems have been developed to support humanities researchers in accessing and analysing material, but despite the current availability of tools, there does not appear to be a wider adoption of these tools. This means that researchers are not making full use of digital archives as a key resource for discovering re-occurring cultural contexts that can help to address new questions, and revisit old research ones through a much larger

body of evidence provided by these digital repositories.

There are several barriers to computer-assisted forms of analysis in the humanities, some barriers are related to the incompatibility of the tools with the research topic, methodology, or digital archive of interest. This is attributed to the fact that the tools are often developed as part of specific research projects, and were not designed to be interoperable, making them inflexible to other data standards and formats, domains, and languages.

A further barrier relates to the way in which the tools are implemented, which are often based on domain and language-specific approaches using the commonly adopted word-level model for the terms of the documents and user query. A word-level model requires a certain degree of preprocessing, involving language-specific segmentation of the text in to words, which are then normalised to identify all occurrences of the same term regardless of the syntax of the language. However, the word-level model is not a suitable representation when the language contains no explicit word-delimiter, or when the morphology of the language is more complex than that of the commonly studied languages for which a wealth of natural language processing resources are available. Furthermore, the objects of study may be represented by historic documents, which contain non-standard spelling and formatting, and are also not commonly supported by natural language processing tools that are optimised for modern forms of the language. The current word-level representation will become difficult to sustain due to the increasing diversity and large-scale nature of digital archives.

The literature suggests that the current tools are not directly supporting the needs of researchers as they are often incompatible with the research approach. For example, summarisation tools such as word-clouds and word-frequency statistics enable researchers to identify interesting correlations between the vocabulary of the documents that are potentially relevant, but they reduce the texts to a new intellectual product that is distanced from the original. Researchers in the humanities are interested in locating and comparing “parallel passages”, which represent repeated structural text patterns that could describe, among other things, eye witness accounts of the same or similar events of interest to historians, or a repeated grammatical structure relevant to a linguist. Humanities researchers also have difficulty in relating to the

approaches adopted by tool developers, which are often poorly documented or tend to obscure how the tool interfaces with the data. When researchers are unable to identify how the tool might be relevant to their research, the tools experience little adoption and are often abandoned as a result, making it difficult for the discipline to move forward.

A review of the current digital tools available to researchers, revealed a common set of tools and functionality that are often provided to support the humanities. The majority of research and commercial tools and systems support full and partial matching of queries, browsing the documents through metadata and ontologies, collaborative search and annotation tools, natural language processing tools for part-of-speech tagging and named entity extraction, and visual mining tools such as word clouds, social network graphs, and time lines for producing visual summaries of the document content.

Chapter 3 introduced the architecture supporting the development of the *Samtla* system, which was developed in response to the need for a generic and flexible set of digital tools for the search and text mining of any domain, language, format, and media. The *Samtla* system is a web application supported by a Model-View-Controller architecture, where the *model* component reads and writes the data used by the system, including that of the document text, metadata, and images. The *view* component supports the deployment of the user interface of the system, and the *controller* component represents the program logic, and provides the bridge between the data and the researchers' interaction with the *view*. The architecture is easily maintained as a result of the separation of concerns provided by the *MVC* design pattern, which means that new tools and features can be introduced without interfering with previously established components. The resulting architecture also enables the *Samtla* system to be deployed very quickly with a consistent set of generic tools, which is demonstrated by several case studies represented by several diverse digital archives, which are different with respect to the morphological complexity of the languages, time periods, a range of media other than textual data, and of varying quality with respect to the resulting digital objects.

The details of the novel infrastructure proposed in the thesis were presented in Chapter 4, which described a relatively new approach reflected by a character-level  $n$ -gram Statistical Language Model *SLM* stored in a space-

optimised  $k$ -truncated suffix tree data structure. Traditional approaches, such as the common boolean retrieval model and Vector Space Model, are often language-dependent, ad-hoc or based on heuristics, and tend to adopt separate data models for search and mining. Furthermore, the commonly adopted word-level model for the terms of the query and document text is responsible for limiting the generalisability of the tools and systems to other domains and languages.

However, the character-level  $n$ -gram *SLM* model has not been widely adopted due to the storage requirements and added complexity of the approach, but it has several advantages over word-level models, such as being domain and language-independent. Furthermore, the character-level  $n$ -gram model provides innate support for erroneous query specification and spelling errors in the document text through the generation of partial matches to the query.

The *SLM* provides a probabilistic framework, where the terms of the document are assigned a probability proportional to their frequency in the document and the collection as a whole. The approach is based on statistically well-founded estimation techniques that have been long established under the context of natural language research in speech recognition. Furthermore, the *SLM* approach has been shown to perform as well as traditional methods in a range of tasks including information retrieval, recommendation, machine translation, and email-spam filtering.

The text mining tools were detailed in Chapter 5, which introduced each of the text mining tools developed from components of the *SLM* used for search. The tools support the specific needs of several research groups introduced in Chapter 3, through query and document recommendation, which highlight the interesting aspects of the archive through the search and browsing activity of the community of researchers. The recommendation tools were straight-forwardly extended address language-specific issues by recommending alternative forms of the query based on a small set of string permutation methods that identify similar items on the basis of the statistics of the language stored in the *SLM*.

Chapter 6 introduced the *Samtla* user interface, which was developed through collaboration with our research groups. The minimal design of the *UI*,



allows researchers to focus on the content of the documents achieved through a modular and context dependent approach to tool deployment. Whilst many of the digital archives supported by *Samtla* are deployed with the same set of generic tools for search and text mining, some tools were developed to address specific needs, such as the *FT newspaper archive*, which required additional tools to operate with both the document text, and scanned images of the original.

The infrastructure was formally evaluated through a crowdsourcing platform, which enlisted a group of general users to assess the relevance of documents in the search results generated by the *SLM* (in Chapter 7). The results were evaluated by adopting a number of well-known non-parametric measures, and a novel approach based on the non-parametric *bootstrap* method to assess the significance of the results. The results demonstrated that the search component of the *Samtla* system consistently provides a ranking of the documents, where the most relevant documents are ranked at the top of the search results.

Although *Samtla* is still in development it currently operates with a range of digital archives, supporting the discovery of documents in languages including Aramaic, Syriac, Mandaic, Hebrew, English, German, French, Hungarian, Italian, and Russian, and different domains reflected by biblical scripture, magic incantation texts, monographs, reports, and news articles. In addition, *Samtla* is not necessarily restricted to historic document collections, but can be extended straightforwardly to other domains that require search and mining of text patterns, such as medical and legal text collections.

To conclude, the field of humanities will not be able to advance in a direction that is adaptable to the recent emergence of large-scale digital archives, unless tools are developed to be generalisable and easily extensible to permit their adoption by a wide-range of researchers in the humanities. Ignoring the current issues faced by humanities researchers, with respect to current tool provision may mean that the human record will be described, and communicated through the perspectives of the dominant cultures. In short, without flexible tools to analyse cultural contexts recorded across multilingual document collections, a cultural-bias may inevitably be forced upon the research conducted by humanities researchers, due to the limitations of the approaches.

## 8.2 SUMMARY OF THE CONTRIBUTIONS

The main contribution of the thesis is a novel language and domain-independent infrastructure that provides a consistent and unified approach to the development of search and text mining tools to support a range of research and disciplines in the humanities, where electronic source material is increasingly being published through digital archives. The contributions presented in the thesis, are summarised as follows:

- **A unified and consistent approach to domain-independent and language-independent search and text mining of digital archives.** The *SLM* can be extended to other domains and languages with very little preprocessing of the documents. Furthermore, the research on *SLMs* has demonstrated that they are relatively simple to implement, provide a basis for a unified approach to tool development, through their straightforward extension to other domains other than speech recognition, and information retrieval for which they have been traditionally adopted.
- **The first practical implementation of a character-level  $n$ -gram *SLM* stored in a suffix tree data structure, as the underlying data model of a system with real users.** There has been a lot of research on *SLMs*, including tool kits for implementing a *SLM* as an integral part of a system, for example, the *Lemur Project* [31]. However, as far as the author is aware, the proposed infrastructure supporting the *Samtla* system is the first practical example of a character-level  $n$ -gram *SLM* stored as a  $k$ -truncated suffix tree structure, which supports the development of generic, flexible, and easily extensible tools that address the specific needs of real users, as represented by the case studies introduced in Chapter 3.
- **An innovative set of text mining algorithms to support the discovery of “parallel passages” represented by variable length character-sequences.** The algorithms developed for text mining are generic and flexible to the domain and language of the digital archive, and have been developed from many successful approaches in disciplines such as bioinformatics. The text mining tools support the identification and comparison of related structural text patterns that are often variable

with respect to their coverage of the topic and extent, as a result of the vocabulary, morphology of the language, or choices made by the author.

- **A novel approach for assessing the performance of the infrastructure through a crowdsourcing platform.** The *SLM* was assessed through a comparison of a user ranking of the documents generated from the users' relevance judgements in response to a query. The evaluation results demonstrated that the infrastructure represented by the character-level  $n$ -gram *SLM* is generating a ranking of the documents that is similar to one generated by a human assessor, meaning that the current implementation provides human-level performance applied to information retrieval tasks.

## 8.3 FUTURE WORK

There are several directions for future work, including the development of the infrastructure to support new forms of analysis based on feature requests by the research groups, and more efficient algorithms for indexing and storing the data model of the infrastructure to enable the infrastructure to support much larger digital archives than it does currently.

### 8.3.1 Developing the infrastructure

#### Scaling the architecture

Chapter 4 discussed the low-level implementation and storage of the *SLM* in a character-based  $k$ -truncated space optimised suffix tree data structure. The current approach involves constructing the suffix tree in memory to support the fast indexing and storage of the terms of the documents. This may not be efficient for very large collections containing hundreds of thousands of documents. One way to scale the approach is to build to disk, where the tree is periodically flushed to disk. With the introduction of Solid State Drives (*SSD*), it is now possible to leverage the random access memory-like performance of the *SSD* to construct the index on disk without the problems faced by previous hardware, where accessing the data requires an unavoidable amount of latency due to the seeking mechanism relied on by traditional hard drive storage, and desecration of the media through constant reading and writing of the data.

Additional scaling of the infrastructure could be achieved by partitioning the suffix tree data structure across several different servers.

### 8.3.2 Developing new tools

#### Data analytics

Data analytics is a common component of the current set of tools identified in Chapter 2. Analytics range from statistical summaries on the distribution of terms in the documents, which are often visualised as word-clouds and social network graphs. The data-analytic tools developed for *Samtla* could provide support for a much more extended analysis through the exploration of topics, genres, and authorship to provide new forms of analysis and ways to identify information relevant to the researcher. The resulting analytics could be incorporated in to the existing search results, or as additional facets in the browsing tool represented by the treemap view.

#### Annotation tools

Many of the tools reviewed in Chapter 2 allow researchers to make corrections to the digitised text or annotations, or permit tagging of the text and images. The development of an annotation tool would require some enhancements to the user interface, and a method for recording the annotations made by individual users, and whether they should be available to the wider community, which would require some form of rights-management. The resulting data collected from the users annotation would provide a further platform for developing additional tools that make use of the data provided by the annotated texts.

#### Event detection and identification

New tools could be developed to help identify important events in the documents that would then be presented to researchers as time lines as per the existing set of tools. The task of event detection is often referred to as event tracking and identification [170]. An event can be described as having a timestamp, and one or more individuals and locations associated with it. The existing set of tools that provide visual tools based on time lines would appear to be based on manually tagged data, whereas the *Samtla* system would require

an approach that is at least semi-automatic and both domain and language independent due to the often multilingual nature of digital archives.

### **Supporting new research**

The *Samtla* system was developed for researchers in the humanities, and correspondingly requires the feedback and feature requests from real users in order to develop the system further. Otherwise, there is a tendency to develop tools that will not be relevant to research that is actually being undertaken by researchers. The infrastructure is flexible to any digital archive, but there may exist some digital archives that represent interesting opportunities for future research and tool development. For example, the *Samtla* system has not yet been applied to a language represented by a syllabic or ideographic script.

# APPENDIX A

## ARCHITECTURE

The infrastructure, architecture and the *Samtla* system was developed using the following technologies and tools:

### A.1 THE MODEL

- Python programming language: with some adoption of the *Numpy* and *Scipy* statistical libraries for computing the prior in Chapter 5, and the correlation measures in Chapter 7.
- SQL database: all data is stored in a SQL database according to function e.g. the *SLM* document model, metadata, pair-wise JSD scores for the related documents, and user activity log data.
- The character-level  $n$ -gram *SLM* for the collection model is stored in *JSON* format to promote portability. The loading function used by the standard JSON library is overridden with a custom loader, which casts strings representing numbers to integer format, which considerably reduces the memory requirements of the data structure when loaded in to memory after construction.

### A.2 THE CONTROLLER

- Python 2.7 programming language: version 2.7 was selected due to its compatibility with some of the more advanced Python packages including *numpy* and *scipy*, which provide powerful matrix computation and implementation of common statistical measures. Python 3.2 may be

more appropriate in future due to the unifying approach adopted for the encoding of strings e.g. unicode versus ASCII.

- Django web framework: provides the main mechanism for communication between the server and the front-end of the *Samtla* system. Also responsible for the storage, retrieval, and validation of user login credentials.

### A.3 THE VIEW

- Javascript programming language: used for developing the functionality of the tools in the user interface.
- jQuery: for addressing the disparity between web browsers to ensure the interactive elements of the interface are consistent cross-browser.
- HTML5: The development of the HTML5 standard provided support for a number of new features including Local Web storage for recording users preferences. The Canvas element of the page is used for rendering and interacting with the original image for the document.

Where possible, native tools and standard-libraries have been preferred due to issues arising from third-party API updates, which can sometimes break a system due to the depreciation of certain features. The advantage is that the system requires little maintenance, aside from development revolving around the system itself e.g. new features and text mining tool development.

# APPENDIX B

## EVALUATION

### B.1 THE TEST QUERIES

The test queries for the formal evaluation in Chapter 7 are presented below. Each test displays the query at the top of the page, and the task involves assigning a relevance grade according to a set of specific criteria. The first test query contained the top-five ranked documents for the query “Satan”, with five additional entries containing the top-five search results for the query “chief priests and scribes”. The user must assign a very low relevance to the last five documents due to the fact that these search results do not contain a match for the query. The second test query “Christ Jesus”, was composed of the top-ten documents ranked in reverse order by their probability inferred from the *SLM*. To pass this test and continue to the evaluation, the user was required to assign relevance in reverse order, with the lower ranked documents receiving the highest relevance.

#### B.1.1 Test query 1

**Test query 1:** “Satan”

##### 1. Job chapter 1

before the LORD, and Satan came also among them. And the LORD said unto Satan, Whence comest thou? Then Satan answered the LORD, and ... feareth God, and escheweth evil? Then Satan answered the LORD, ... down in it. And the LORD said unto Satan, Hast thou considered ...



**2. Jude chapter 20**

which is the Devil, and Satan, and bound him a thousand ... the thousand years are expired, Satan shall be loosed out of ...

**3. Zechariah chapter 3**

the angel of the LORD, and Satan standing at his right hand to resist him. And the LORD said unto Satan, The LORD rebuke thee, O ... Satan, The LORD rebuke thee, O Satan; even the LORD that ...

**4. Job chapter 2**

before the LORD, and Satan came also among them to present himself before the LORD. And the LORD said unto Satan, From whence comest thou? ... to destroy him without cause. And Satan answered the LORD,... down in it. And the LORD said unto Satan, Hast thou considered

**5. Jude chapter 12**

serpent, called the Devil, and Satan, which deceiveth the whole ...

**6. Malachi chapter 16**

many things of the elders and chief priests and scribes, and be killed, and be ...

**7. Mark chapter 23**

answered him nothing. And the chief priests and scribes stood and vehemently ..., when he had called together the chief priests and the rulers ... Then said Pilate to the chief priests and to the people, I find ...

**8. Malachi chapter 21**

he healed them. And when the chief priests and scribes saw the wonderful things ... grind him to powder. And when the chief priests and Pharisees had heard his ... was come into the temple, the chief priests and the elders of the people ...

**9. Malachi chapter 2**

when he had gathered all the chief priests and scribes of the people together...

**10. Mark chapter 9**

be rejected of the elders and chief priests and scribes, and be slain, and be raised ...

**B.1.2 Test query 2****Test query 2: "Christ Jesus"****1. 2 Thessalonians chapter 3**

in the faith which is in Christ Jesus. These things write I unto ...

**2. 2 Thessalonians chapter 2**

between God and men, the man Christ Jesus; Who gave himself a ransom ...

**3. 2 Corinthians chapter 6**

and I unto the world. For in Christ Jesus neither circumcision avail ...

**4. 1 Timothy chapter 3**

all that will live godly in Christ Jesus shall suffer persecution... through faith which is in Christ Jesus. All scripture is given ...

**5. James chapter 5**

unto his eternal glory by Christ Jesus, after that ye have suffered ... be with you all that are in Christ Jesus. Amen. The Second General

**6. Ephesians chapter 4**

your hearts and minds through Christ Jesus. Finally, brethren, ... Amen. Salute every saint in Christ Jesus. The brethren which are ... to his riches in glory by Christ Jesus. Now unto God and our Father ...

**7. 1 Timothy chapter 1**

promise of life which is in Christ Jesus, To Timothy, my dearly beloved ... and grace, which was given us in Christ Jesus before the world began,... peace, from God the Father and Christ Jesus our Lord. I thank God,...

**8. Galatians chapter 2**

together in heavenly places in Christ Jesus: That in the ages to come ... are his workmanship, created in Christ Jesus unto good works, which ...

his kindness toward us through Christ Jesus. For by grace are ye saved  
...

9. **Ephesians chapter 3**

in the spirit, and rejoice in Christ Jesus, and have no confidence in ... or  
which also I am apprehended of Christ Jesus. Brethren, I count not...  
excellency of the knowledge of Christ Jesus my Lord: for whom I have  
...

10. **2 Thessalonians chapter 1**

to my trust. And I thank Christ Jesus our Lord, who hath enabled ...  
with faith and love which is in Christ Jesus. This is a faithful saying,  
and worthy of all acceptation, that Christ Jesus came into the world to  
...

## B.2 EVALUATION QUERIES

The fifty queries used for the formal evaluation of the *Samtla SLM* data model are presented in Table B.1, below.

query <i>id</i>	query text	query <i>id</i>	query text
1	tribe of manasseh	26	fight against the children of ammon
2	in the beginning god created the heaven and the earth	27	the children of joseph
3	moses commanded the children of israel	28	they buried him in the city of david
4	the grace of our lord jesus christ be with you all	29	and the spirit of god came upon him
5	are they not written in the book of the chronicles of the kings of israel	30	the ark of the god of israel
6	as the lord commanded mooses	31	the devil
7	blessed be the lord god of israel from everlasting to everlasting	32	build the house of the lord
8	the heads of the fathers of the levites	33	men of judah and the inhabitants of jerusalem
9	according to the commandment of mooses	34	then came the word of the lord unto jeremiah
10	bare the ark of the covenant of the lord	35	nebuchadrezzar king of babylon
11	they gathered themselves together against mooses	36	so rabshakeh returned and found the king of assyria
12	according to the house of their fathers	37	the servants of david
13	all the inhabitants of the land from before you	38	in the name of our lord jesus christ
14	we go up to jerusalem	39	moses spake unto the children
15	the beginning of the world	40	the name of the lord
16	house unto the name of the lord	41	sacrifice of peace offerings
17	and the lord said unto mooses	42	on the seventh day
18	the priest shall make an atonement for him	43	all the tribes of israel
19	a burnt offering unto the lord	44	isaiah the prophet the son of amoz
20	in the name of our lord jesus christ	45	caesarea
21	the angel of the lord appeared	46	the house of israel and the house of judah
22	moses did as the lord commanded him	47	throughout all the land of egypt
23	reigned in his stead	48	the voice of the lord our god
24	land of the children of ammon	49	they have sinned against thee
25	in the beginning of the barley harvest	50	god overthrew sodom and gomorrah

Table B.1: The fifty queries selected for the formal evaluation.

# BIBLIOGRAPHY

- [1] Ibm languageware. <https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=6adead21-9991-44f6-bdbb-baf0d2e8a673>. [Online; accessed 18 February 2015].
- [2] Msn book search and the british library. <http://blogs.bing.com/search/2005/11/04/msn-book-search-and-the-british-library>, 2005. [Online; accessed 18-February-2016].
- [3] Msn search announces msn book search. <http://news.microsoft.com/2005/10/25/msn-search-announces-msn-book-search/>, October 2005. [Online; accessed 15-May-2016].
- [4] Docuburst. <http://vialab.science.uoit.ca/docuburst/index.php>, 2011. [Online; accessed 15-May-2016].
- [5] Resource description framework (rdf). <https://www.w3.org/RDF/>, 2014. [Online; accessed 15-May-2016].
- [6] Vmba: Virtual magic bowl archive. <http://www.southampton.ac.uk/vmba/>, 2014. [Online; accessed 28-January-2014].
- [7] Xmlhttprequest level 1: W3c working draft 30 january 2014. <https://www.w3.org/TR/XMLHttpRequest/>, 2014. [Online; accessed 15-May-2016].
- [8] Accordance. <http://www.accordancebible.com>, 2015. [Online: accessed 01-October-2015].
- [9] Cultivating understanding through research and adaptivity. [http://1641.tcd.ie/project-related\\_projects.php](http://1641.tcd.ie/project-related_projects.php), 2015. [Online; accessed 20 February 2015].

- [10] Ecma-404 the json data interchange standard. <http://json.org/>, 2015. [Online; accessed 12-January-2015].
- [11] Elasticsearch. <https://www.elastic.co/products/elasticsearch>, 2015. [Online; accessed 3-November-2015].
- [12] List of biblical places. [https://en.wikipedia.org/wiki/List\\_of\\_biblical\\_places](https://en.wikipedia.org/wiki/List_of_biblical_places), 2015. [Online; accessed 02-May-2015].
- [13] List of major biblical figures. [https://en.wikipedia.org/wiki/List\\_of\\_major\\_biblical\\_figures](https://en.wikipedia.org/wiki/List_of_major_biblical_figures), 2015. [Online; accessed 02-May-2015].
- [14] Lists of lists on wikipedia. [https://en.m.wikipedia.org/wiki/List\\_of\\_lists\\_of\\_lists](https://en.m.wikipedia.org/wiki/List_of_lists_of_lists), 2015. [Online; accessed 01-October-2015].
- [15] Logos. <http://www.logos.com/about>, 2015. [Online; accessed 20-October-2015].
- [16] Openstreetmap api v0.6. [http://wiki.openstreetmap.org/wiki/API\\_v0.6](http://wiki.openstreetmap.org/wiki/API_v0.6), 2015. [Online; accessed 15-May-2016].
- [17] Responsa: Brief history. <http://www.biu.ac.il/JH/Responsa/history.htm>, 2015. [Online; accessed 27-October-2015].
- [18] Texcavator. <https://bitbucket.org/jvdzwaan/texcavator>, 2015. [Online; accessed 3-November-2015].
- [19] What is the 1641 depositions project? <http://www.1641.tcd.ie/project.php>, 2015. [Online; accessed 20 February 2015].
- [20] Bibleworks. [http://www.bibleworks.com/classroom/1\\_10/](http://www.bibleworks.com/classroom/1_10/), 2016. [Online; accessed 8-March-2016].
- [21] British museum images: Ceramic bowl with aramaic magic inscription. <http://www.bmimages.com/preview.asp?image=00036056001>, 2016. [Online; accessed 11-April-2016].
- [22] Compare documents side by side. <https://support.office.com/en-US/article/Compare-documents-side-by-side-52445547-7C07-475B-BB1D-22A98175EF04>, 2016. [Online; accessed 22-February-2016].

- [23] Creative commons: About the licenses. <https://creativecommons.org/licenses/>, 2016. [Online; accessed 21-March-2016].
- [24] Crowdfunder. <https://www.crowdfunder.com/>, 2016. [Online; accessed 16-May-2016].
- [25] Diff doc tool. <http://www.softinterface.com/MD/Document-Comparison-Software.htm>, 2016. [Online; accessed 8-February-2016].
- [26] Discussion on the meaning of the aramaic related queries identified by *samtla*. Personal communication, July 2016.
- [27] Gale cengage learning: Financial times historical archive. <http://gale.cengage.co.uk/financial-times-historical-archive.aspx>, 2016. [Online; accessed 09-March-2016].
- [28] Gmail. [https://www.google.com/intl/en\\_GB/mail/help/about.html](https://www.google.com/intl/en_GB/mail/help/about.html), 2016. [Online; accessed 19-May-2016].
- [29] Google photos. <https://photos.google.com>, 2016. [Online; accessed 19-May-2016].
- [30] Hebrew manuscripts digitisation project. <http://www.bl.uk/projects/hebrew-manuscripts-digitisation-project>, 2016. [Online; accessed 11-April-2016].
- [31] The lemur project. <http://www.lemurproject.org/>, 2016. [Online; accessed 20-June-2016].
- [32] The million book digital library project. <http://www.rr.cs.cmu.edu/mbdl.htm>, 2016. [Online; accessed 23-April-2016].
- [33] The national gallery – collection overview. <https://www.nationalgallery.org.uk/paintings/collection-overview>, December 2016.
- [34] The python standard library: Built-in types. <https://docs.python.org/2/library/stdtypes.html>, 2016. [Online; accessed 18-February-2016].

- [35] Search help: Autocomplete. <https://support.google.com/websearch/answer/106230?hl=en>, 2016. [Online; accessed 2-March-2016].
- [36] Shakespeare's globe. <http://library.shakespearesglobe.com:2000/#!dashboard>, 2016. [Online; accessed 11-April-2016].
- [37] The software environment for the advancement of scholarly research (sears). <http://www.seasr.org/>, 2016. [Online; accessed 15-May-2016].
- [38] Tapor 3: Discover research tools for studying texts. <http://tapor-test.artsrn.ualberta.ca/home>, 2016. [Online; accessed 09-April-2016].
- [39] Tei: Text encoding initiative. <http://www.tei-c.org/index.xml>, 2016. [Online; accessed 18-February-2016].
- [40] Text metadata services. <http://www.clearforest.com/>, December 2016.
- [41] Text mining the i vit s k ton th. <https://leminhkhai.wordpress.com/2012/09/30/text-mining-the-dai-viet-su-ky-toan-thu/>, 2016. [Online; accessed 14-March-2016].
- [42] The text retrieval conference (trec). <http://trec.nist.gov/overview.html>, 2016. [Online; accessed 21-February-2016].
- [43] Virtual magic bowl archive. <http://humanities.exeter.ac.uk/theology/research/projects/vmba/>, 2016. [Online; accessed 15-January-2016].
- [44] Voyant tools. <http://voyant-tools.org/>, 2016. [Online; accessed 11-March-2016].
- [45] What is xml? [http://www.w3schools.com/xml/xml\\_what\\_is.asp](http://www.w3schools.com/xml/xml_what_is.asp), 2016. [Online; accessed 11-April-2016].
- [46] William blake archive. <http://www.blakearchive.org/blake/>, 2016. [Online; accessed 19-May-2016].
- [47] Word length distribution in various languages. <https://reference.wolfram.com/language/example/>



- WordLengthDistributioninVariousLanguages.html, 2016. [Online; accessed 11-January-2016].
- [48] Wordseer: A text analysis environment for humanities scholars. <http://wordseer.berkeley.edu/>, 2016. [Online; accessed 10-April-2016].
- [49] Marc format for archival and manuscripts control. <http://www2.archivists.org/glossary/terms/m/marc-format-for-archival-and-manuscripts-control#.Vy2tsngrLVM>, [Online; accessed 28th June 2015].
- [50] Chadia Abras, Diane Maloney-krichmar, and Jenny Preece. User-centered design. In *In Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*. Publications, 2004.
- [51] C.C. Aggarwal and C. Zhai. *Mining Text Data*. Springer-Verlag New York Inc, 2012.
- [52] Omar Alonso and Ricardo A. Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings*, pages 153–164, 2011.
- [53] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, and Aristides Gionis. An optimization framework for query recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10*, pages 161–170, New York, NY, USA, 2010. ACM.
- [54] Howard Anton. *Elementary linear algebra*. Wiley, 2010.
- [55] Ebru Arisoy, Helin Dutağacı, and Levent M. Arslan. A unified language model for large vocabulary continuous speech recognition of turkish. *Signal Process.*, 86:2844–2862, October 2006.
- [56] W. Y. Arms. *Digital Libraries*. Digital libraries and electronic publishing. MIT Press, 2000.
- [57] Ricardo Baeza-Yates, Carlos Hurtado, Marcelo Mendoza, Marco Mesiti, Can Türker, Yannis Tzitzikas, and Athena I. Vakali. *Current Trends in*

- Database Technology - EDBT 2004 Workshops: EDBT 2004 Workshops PhD, DataX, PIM, P2P&DB, and ClustWeb, Heraklion, Crete, Greece, March 14-18, 2004. Revised Selected Papers*, chapter Query Recommendation Using Query Logs in Search Engines, pages 588–596. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [58] Lalit R. Bahl, F. Jelinek, and R. Mercer. A maximum likelihood approach to continuous speech recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5(2):179–190, March 1983.
- [59] Judit Bar-Ilan, Kevin Keenoy, Mark Levene, and Eti Yaari. Presentation bias is significant in determining user preference for search results a user study. *Journal of the American Society for Information Science and Technology*, 60(1):135–149, 2009.
- [60] Judit Bar-Ilan, Mazlita Mat-Hassan, and Mark Levene. Methods for comparing rankings of search engine results. *Comput. Netw.*, 50(10):1448–1463, July 2006.
- [61] M. Barsky, U. Stege, and A. Thomo. A survey of practical algorithms for suffix tree construction in external memory. *Softw. Pract. Exper.*, 40(11):965–988, October 2010.
- [62] Chris Biemann, Gregory R. Crane, Christiane D. Fellbaum, and Alexander Mehler. Computational Humanities - bridging the gap between Computer Science and Digital Humanities (Dagstuhl Seminar 14301). *Dagstuhl Reports*, 4(7):80–111, 2014.
- [63] Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what’s in a name. *Mach. Learn.*, 34(1-3):211–231, February 1999.
- [64] Phillip Bonacich. Power and Centrality: A Family of Measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.
- [65] Christine L. Borgman. The digital future is now: A call to action for the humanities. *Digital Humanities Quarterly*, 3(4), 2009.
- [66] D. C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75, 2008.

- [67] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, Jeffrey Dean, and Google Inc. Large language models in machine translation. In *In EMNLP*, pages 858–867, 2007.
- [68] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, April 1998.
- [69] Vanda Broughton. Faceted classification as a basis for knowledge organization in a digital environment: The bliss bibliographic classification as a model for vocabulary management and the creation of multidimensional knowledge structures. *New Rev. Hypermedia Multimedia*, 7(1):67–102, July 2002.
- [70] Mark Bruls, Kees Huizing, and Jarke van Wijk. Squarified treemaps. In *In Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42. Press, 1999.
- [71] George Buchanan, Sally Jo Cunningham, Ann Blandford, Jon Rimmer, Claire Warwick, Stavros Christodoulakis, and A. Min Tjoa. *Research and Advanced Technology for Digital Libraries: 9th European Conference, ECDL 2005, Vienna, Austria, September 18-23, 2005. Proceedings*, chapter Information Seeking by Humanities Scholars, pages 218–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [72] Monica E. Bulger, Eric T. Meyer, Grace De la Flor, Melissa Terras, Sally Wyatt, Marina Jirotko, Kathryn Eccles, and Christine McCarthy Madsen. Reinventing research? information practices in the humanities. *A Research Information Network Report*, April 2011.
- [73] Steve Burbeck. Applications programming in smalltalk-80(tm): How to use model-view-controller (mvc), 1987.
- [74] Anne (et.al) Burdick. *Digital Humanities*. Cambridge, MA:MIT Press, 2012.
- [75] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

- [76] William B. Cavnar and John M. Trenkle. N-gram-based text categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.
- [77] Olivier Chapelle, Donald Metzler, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09*, pages 621–630, New York, NY, USA, 2009. ACM.
- [78] S.F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–394, 1999.
- [79] Y. Choueka. Responsa: A full-text retrieval system with linguistic processing for a 65-million word corpus of jewish heritage in hebrew. *Data Eng.*, 12(4):22–31, November 1989.
- [80] D. Cohen, N. Fraistat, M. Kirschenbaum, and T. Scheinfeldt. Tools for data-driven scholarship: Past, present, future. a report on the workshop of 2224 october, 2008, turf valley resort, ellicott city, md: Maryland institute for technology and the humanities, 2009. Technical report, American Council of Learned Societies, 2009. [Online; accessed 2-May-2016].
- [81] Daniel J. Cohen. From Babel to Knowledge: Data Mining large Digital Collections. *D-Lib Magazine*, 12(3), 2006.
- [82] Daniel J. Cohen and Roy Rosenzweig. Web of lies? historical knowledge on the internet. *First Monday*, 10(12), 2005.
- [83] Christopher Collins, Sheelagh Carpendale, and Gerald Penn. Docuburst: Visualizing document content using language structure. *Computer Graphics Forum (Proc. of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis))*, 28(3):1039–1046, 2009.
- [84] Joëlle Coutaz. Pac: An object oriented model for implementing user interfaces. *SIGCHI Bull.*, 19(2):37–41, October 1987.
- [85] Steven P. Crain, Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Dimensionality reduction and topic modeling: From latent semantic index-

- ing to latent dirichlet allocation and beyond. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 129–161. Springer, 2012.
- [86] W. Bruce Croft, Robert Cook, and Dean Wilder. Providing government information on the internet: Experiences with thomas. In *In The Second International Conference on the Theory and Practice of Digital Libraries*, 1995.
- [87] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, 1997.
- [88] M.J. de Jong. *Isaiah Among the Ancient Near Eastern Prophets: A Comparative Study of the Earliest Stages of the Isaiah Tradition and the Neo-Assyrian Prophecies*. Supplements to Vetus Testamentum. Brill, 2007.
- [89] Thomas Demeester, Robin Aly, Djoerd Hiemstra, Dong Nguyen, and Chris Develder. Predicting relevance based on assessor disagreement: analysis and practical applications for search evaluation. *Information Retrieval Journal*, pages 1–29, 2015.
- [90] P. Diaconis and R. L. Graham. Spearman’s footrule as a measure of disarray. *Royal Statistical Society Series B*, 32(24):262–268, 1977.
- [91] Dominik Maria Endres and Johannes E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.
- [92] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. In *In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 28–36, 2003.
- [93] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.
- [94] Andreas Fickers. Towards a new digital historicism? doing history in the age of abundance. *VIEW Journal of European Television History and Culture*, 1(1), 2012.

- [95] W.O. Galitz. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. Wiley Desktop Editions. Wiley, 2007.
- [96] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [97] H. Garcia-Molina, M. Joglekar, A. Marcus, A.G. Parameswaran, and V. Verroios. Challenges in data crowdsourcing. *Knowledge and Data Engineering, IEEE Transactions on*, PP(99):1–1, 2016.
- [98] F. Gibbs and T. Owens. Building better digital humanities tools: Toward broader audiences and user-centered designs. *Digital Humanities Quarterly*, 6(2), 2012.
- [99] Roberto Grossi and Jeffrey Scott Vitter. Compressed suffix arrays and suffix trees with applications to text indexing and string matching (extended abstract). In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, STOC '00*, pages 397–406, New York, NY, USA, 2000. ACM.
- [100] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- [101] Ronald. Hagler. *The bibliographic record and information technology / Ronald Hagler*. American Library Association Chicago, 3rd ed. edition, 1997.
- [102] Marti A. Hearst. Uis for faceted navigation: Recent advances and remaining open problems. In *in the Workshop on Computer Interaction and Information Retrieval, HCIR 2008*, 2008.
- [103] Marti A. Hearst. *Search User Interfaces*. Cambridge University Press, 1 edition, 2009.
- [104] Marti A. Hearst and Chandu Karadi. Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of SIGIR-97, 20th ACM Interna-*

- tional Conference on Research and Development in Information Retrieval*, pages 246–255, Philadelphia, US, 1997.
- [105] Anthony T. Holdener, III. *Ajax: The Definitive Guide*. O’Reilly, first edition, 2008.
- [106] John Houvardas and Efstathios Stamatatos. *N-Gram Feature Selection for Authorship Identification*, pages 77–86. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [107] Samuel Huston and W. Bruce Croft. Evaluating verbose query processing techniques. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’10*, pages 291–298, New York, NY, USA, 2010. ACM.
- [108] Peter Ingwersen. *Information Retrieval Interaction*. Taylor Graham Publishing, London, UK, UK, 1992.
- [109] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.
- [110] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.
- [111] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *International Journal of Computer Technology and Applications*, 2(6):1930–1938, 2011.
- [112] Brian Johnson and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2Nd Conference on Visualization ’91, VIS ’91*, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.
- [113] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

- [114] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.
- [115] H. Juusola and Suomen Itämainen Seura. *Linguistic Peculiarities in the Aramaic Magic Bowl Texts*. Studia Orientalia. Finnish Oriental Society, 1999.
- [116] IOANNIS KANARIS, KONSTANTINOS KANARIS, IOANNIS HOUVARDAS, and EFSTATHIOS STAMATATOS. Words versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools*, 16(06):1047–1067, 2007.
- [117] Gabriella Kazai. In search of quality in crowdsourcing for search engine evaluation. In *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings*, pages 165–176, 2011.
- [118] Jun’ichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.
- [119] Jaana Kekäläinen and Kalervo Järvelin. Using graded relevance assessments in ir evaluation. *J. Am. Soc. Inf. Sci. Technol.*, 53(13):1120–1129, November 2002.
- [120] Max Kemman, Martijn Kleppe, and Stef Scagliola. Just google it - digital research practices of humanities scholars. *CoRR*, abs/1309.2434, 2013.
- [121] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *AAAI 2016*, 2015.
- [122] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’08*, pages 453–456, New York, NY, USA, 2008. ACM.



- [123] Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D. Manning. Named entity recognition with character-level models. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 180–183, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [124] Wessel Kraaij, Thijs Westerveld, and Djoerd Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02*, pages 27–34, New York, NY, USA, 2002. ACM.
- [125] Ralf Krestel and Peter Fankhauser. Language models and topic models for personalizing tag recommendation. In Jimmy Xiangji Huang, Irwin King, Vijay V. Raghavan, and Stefan Rueger, editors, *Web Intelligence*, pages 82–89. IEEE, 2010.
- [126] Oren Kurland and Lillian Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '04*, pages 194–201, New York, NY, USA, 2004. ACM.
- [127] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01*, pages 111–119, New York, NY, USA, 2001. ACM.
- [128] Victor Lavrenko, Matthew D. Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Language models for financial news recommendation. In *CIKM*, pages 389–396. ACM, 2000.
- [129] Matthew Lease and Emine Yilmaz. Crowdsourcing for information retrieval: introduction to the special issue. *Information Retrieval*, 16(2):91–100, 2013.
- [130] A. Leff and J.T. Rayfield. Web-application development using the model/view/controller design pattern. In *Enterprise Distributed Object Com-*

- puting Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International*, pages 118–127, 2001.
- [131] D. Levene. *Curse Or Blessing: What's in the Magic Bowl?* Parkes Institute pamphlet. University of Southampton, 2002.
- [132] M. Levene. *An Introduction to Search Engines and Web Navigation*. John Wiley & Sons, Hoboken, New Jersey, 2nd edition, 2010.
- [133] VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.
- [134] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning, ECML '98*, pages 4–15, London, UK, UK, 1998. Springer-Verlag.
- [135] Michael Liedtke. Google book-scanning efforts spark debate. [http://www.washingtonpost.com/wp-dyn/content/article/2006/12/20/AR2006122000213\\_pf.html](http://www.washingtonpost.com/wp-dyn/content/article/2006/12/20/AR2006122000213_pf.html), 2006. [Online; accessed 17-May-2016].
- [136] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37:145–151, 1991.
- [137] David E. Losada and Leif Azzopardi. Assessing multivariate bernoulli models for information retrieval. *ACM Trans. Inf. Syst.*, 26(3):17:1–17:46, June 2008.
- [138] J. Ma and L. Zhang. Modern BLAST programs. In *Problem Solving Handbook in Computational Biology and Bioinformatics*. Springer US, 2011.
- [139] J Makhoul and R Schwartz. State of the art in continuous speech recognition. *Proceedings of the National Academy of Sciences*, 92(22):9956–9963, 1995.
- [140] David Maltz and Kate Ehrlich. Pointing the way: Active collaborative filtering. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95*, pages 202–209, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

- [141] Thomas Mann. The peloponnesian war and the future of reference, cataloging, and scholarship in research libraries. *Journal of Library Metadata*, 8(1):53–100, 2008.
- [142] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [143] U.-V. MARTI and H. BUNKE. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):65–90, 2001.
- [144] Paul Mcnamee and James Mayfield. Character n-gram tokenization for european language text retrieval. *Inf. Retr.*, 7(1-2):73–97, January 2004.
- [145] Nimrod Megiddo and Dharmendra S. Modha. Outperforming lru with an adaptive replacement cache algorithm. *IEEE Computer*, 37(4):58–65, 2004.
- [146] Sean Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.
- [147] David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 214–221, New York, NY, USA, 1999. ACM.
- [148] Stefano Mizzaro. Relevance: The whole history. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 48:810–832, 1997.
- [149] J.A. Montgomery. *Aramaic Incantation Texts from Nippur*. Publications // University of Pennsylvania Philadelphia, Pa. / Museum. Univ. Museum, 1913.
- [150] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

- [151] Joong Chae Na, Alberto Apostolico, Costas S. Iliopoulos, and Kunsoo Park. Truncated suffix trees and their application to data compression. *Theor. Comput. Sci.*, 304(1-3):87–101, July 2003.
- [152] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- [153] JAKOB NIELSEN. Breadcrumb navigation increasingly useful. ”<http://www.useit.com/alertbox/breadcrumbs.html>”, April 2007.
- [154] Michael P. Oakes. *Literary Detective Work on the Computer*. John Benjamins Publishing Company, 2014.
- [155] Douglas W. Oard, Gina anne Levow, and Clara I. Cabezas. Clef experiments at the university of maryland: Statistical stemming and back-off translation strategies. In *In Working Notes of the First Cross-Language Evaluation Forum (CLEF-1)*, page <http://www.glue.umd>. Springer, 2000.
- [156] Rajesh Pampapathi, Boris Mirkin, and Mark Levene. A suffix tree approach to anti-spam email filtering. *Machine Learning*, 65(1):309–338, 2006.
- [157] Fuchun Peng and Dale Schuurmans. Combining naive bayes and n-gram language models for text classification. In *Advances in Information Retrieval*, pages 335–350. Springer Berlin Heidelberg, 2003.
- [158] Michael Piotrowski. Natural language processing for historical texts. *Synthesis Lectures on Human Language Technologies*, 5(2):1–157, 2012.
- [159] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’98, pages 275–281, New York, NY, USA, 1998. ACM.
- [160] M. F. Porter. An algorithm for suffix stripping. In Karen Sparck Jones and Peter Willett, editors, *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

- [161] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.
- [162] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.
- [163] Geoffrey Rockwell. What is text analysis, really? *Literary and Linguistic Computing*, 18(2):209–219, 2003.
- [164] T. Rommel. Literary studies. In *A Companion to Digital Humanities*, pages 88–96. Blackwell Publishing Ltd, 2007.
- [165] Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? In *Proceedings of the IEEE*, volume 88, pages 1270–1278, 2000.
- [166] Roy Rosenzweig. Scarcity or abundance? Preserving the past in a digital era. *American historical review*, 108(3):735–762, June 2003.
- [167] William H. Rossell. *A handbook of Aramaic magical texts*. Shelton Semitic series, no. 2. Ringwood Borough, N.J. Shelton College, Skylands, 1953.
- [168] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics, 2011.
- [169] Tetsuya Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06*, pages 525–532, New York, NY, USA, 2006. ACM.
- [170] Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. Event detection and tracking in social streams. In *In Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2009)*. AAAI, 2009.
- [171] Linda Schamber, Michael Eisenberg, and Michael S. Nilan. A re-examination of relevance: Toward a dynamic, situational definition. *Inf. Process. Manage.*, 26(6):755–776, November 1990.

- [172] Susan Schreibman and Ann M. Hanlon. Determining value for digital humanities tools: Report on a survey of tool developers. *Digital Humanities Quarterly*, 4(2), 2010.
- [173] Marcel H. Schulz, Sebastian Bauer, and Peter N. Robinson. The generalised k-truncated suffix tree for time-and space-efficient searches in multiple dna or protein sequences. *IJBRA*, 4(1):81–95, 2008.
- [174] Katie Shilton. Supporting digital tools for humanists: Investigating tool infrastructure. final report: May 15, 2009. <http://www.clir.org/pubs/archives/ShiltonToolsfinal.pdf>, May 2009. [Online; accessed 8-February-2016].
- [175] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.
- [176] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07*, pages 623–632, New York, NY, USA, 2007. ACM.
- [177] M. Sokoloff. *A Dictionary of Jewish Palestinian Aramaic of the ...* Dictionaries of Talmud, Midrash, and Targum. Bar Ilan University Press, 2002.
- [178] Fei Song and W. Bruce Croft. A general language model for information retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management, CIKM '99*, pages 316–321, New York, NY, USA, 1999. ACM.
- [179] Efstathios Stamatatos. Intrinsic plagiarism detection using character n-gram profiles. *threshold*, 2:1–500, 2009.
- [180] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *In KDD Workshop on Text Mining*, 2000.

- [181] Sanna Talja and Hanni Maula. Reasons for the use and non-use of electronic journals and databases: A domain analytic study in four scholarly disciplines. *Journal of Documentation*, 59(6):673–691, 2003.
- [182] Yuanyuan Tian, Sandeep Tata, Richard A. Hankins, and Jignesh M. Patel. Practical methods for constructing suffix trees. *The VLDB Journal*, 14(3):281–299, September 2005.
- [183] Helen R. Tibbo. Primarily history in america: How u.s. historians search for primary materials at the dawn of the digital age. *The American Archivist*, 66(1):950, 2003.
- [184] John Unsworth. Scholarly primitives: what methods do humanities researchers have in common, and how might our tools reflect this? <http://www3.isrl.illinois.edu/~unsworth/Kings.5-00/primitives.html>, 2000. [Online; accessed 8-April-2015].
- [185] John Unsworth. Tool-time, or ”haven’t we been here already?” ten years in humanities computing. *Transforming Disciplines: The Humanities and Computer Science*, 2003. [Online; accessed 8-April-2015].
- [186] Ellen M. Voorhees. The philosophy of information retrieval evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, CLEF ’01, pages 355–370, London, UK, UK, 2002. Springer-Verlag.
- [187] Marlo Welshons. Our Cultural Commonwealth: The report of the American Council of Learned Societies Commission on Cyberinfrastructure for the Humanities and Social Sciences. Technical report, American Council of Learned Societies, 2006. [Online; accessed 2-May-2016].
- [188] Stephen E. Wiberley and William G. Jones. Patterns of information seeking in the humanities. *College and Research Libraries*, 50(6):638–645, 1989.
- [189] Max L. Wilson. Search user interface design. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 3(3):1–143, 2011.

- [190] Philip C. Woodland, Sue E. Johnson, P. Jourlin, and Karen Sparck Jones. Effects of out of vocabulary words in spoken document retrieval. In *SIGIR*, pages 372–374, 2000.
- [191] Hengzhi Wu, Gabriella Kazai, and Michael Taylor. Book search experiments: Investigating ir methods for the indexing and retrieval of books. In *Advances in Information Retrieval, 30th European Conference on IR Research, ECIR 2008*, volume 4956 of *Lecture Notes in Computer Science*, pages 234–245. Springer, April 2008.
- [192] Zimin Wu and Gwyneth Tseng. Chinese text segmentation for text retrieval: Achievements and problems. *J. Am. Soc. Inf. Sci.*, 44(9):532–542, October 1993.
- [193] Omar F. Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11*, pages 1220–1229, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [194] C. Zhai. *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, San Francisco, 2009.
- [195] ChengXiang Zhai. Statistical language models for information retrieval: A critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, March 2008.
- [196] Chengxiang Zhai and John Lafferty. The dual role of smoothing in the language modeling approach. In *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR) 2001*, pages 31–36, 2001.
- [197] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, April 2004.
- [198] Diane Zorich. *A survey of digital humanities centers in the United States*. Council on Library and Information Resources, 2008.