HYBRID QUERY EXPANSION ASSISTED ADAPTIVE VISUAL INTERFACE FOR EXPLORATORY INFORMATION RETRIEVAL

by

Manav Sharma

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2021

Table of Contents

List of Tables . . . . **iv**

List of Figures . . . . **v**

Abstract . . . . **vii**

List of Abbreviations and Symbols Used . . . . **viii**

Acknowledgements . . . . **ix**

Chapter 1 Introduction **1**

1.1 Research Problem and Hypothesis 2

1.2 Contribution 3

1.3 Thesis outline 4

Chapter 2 Related Work **5**

2.1 Query Expansion in IR 5
2.1.1 Local Query Expansion 5
2.1.2 Global Query Expansion 7

2.2 Visual exploration for information retrieval 8

Chapter 3 Model Framework **12**

3.1 Overview 12

3.2 Design Guidelines 14

3.3 System Flow 15

3.4 Model Framework 18
3.4.1 Local Query Expansion Model 18
3.4.2 Word Embedding Expansion Model 21
3.4.3 Hybrid Expansion Model 21
3.4.4 Document Visualization 23

Chapter 4 Experiments **27**

4.1 Experimental Setup 27
4.1.1 Dataset 27

4.1.2 Development Environment 27
4.1.3 Text pre-processing 27
4.1.4 Word Embedding 28
4.1.5 Baseline and Performance Metrics 28
4.1.6 Evaluator for Query Expansion Model 28
4.1.7 Parameter settings 29

4.2 Application Use Cases 29

4.3 Use case Scenarios 30
4.3.1 Scenario 1 30
4.3.2 Scenario 2 31
4.3.3 Scenario 3 33

4.4 Discussion 35

Chapter 5 Conclusion 37

5.1 Limitations 37

5.2 Conclusion and Future Work 38

Bibliography 39

List of Tables

4.1 Statistics of collection used for experiments evaluations 27

4.2 Performance Measure of the baseline model BM25 and the interface proposed in this thesis on initial retrieval (VizIR) for the User Query in scenario 1. 30

4.3 Performance Measure of the baseline model BM25 and the interface proposed in this thesis on initial retrieval (QE-VizIR Iter-0) and after expansion term selection and user feedback provided in second iteration (QE-VizIR Iter-1) for the User Query in scenario 2. 33

4.4 Performance Measure of the baseline model BM25 and the interface proposed in this thesis on initial retrieval (QE-VizIR Iter-0) and after expansion term selection and user feedback provided in second iteration (QE-VizIR Iter-1) for the User Query in scenario 3. 35

List of Figures

2.1 VIBE system showing hundreds of documents. Documents are positioned based on the similarity ratio with the query terms known as Points of Interest. [26] 9

2.2 Visual Re-Ranking for Multi-Aspect Information Retrieval: The interactive relevance map is rendered based on three topics selected shown in the red label, and one topic disabled is greyed out. The user can select the exploration cursor (in blue) to select the region for re-ranking the documents. Documents rendered as red points are the top documents retrieved in rank list [18]. 10

2.3 TRIVIR interface: The scatterplot view (2) projects the documents based on TF-IDF or word-embeddings representation using the projection technique t-SNE (also supports least-square projection). The interface provides various views to filter the documents [9] 11

3.1 Framework of the Hybrid Query Expansion assisted Visual Interface 15

3.2 Overview of our framework. The numbers indicate the different sections of the visual interface. (1) Search Bar, (2) scatterplot of documents and query, (3) Document View displaying the document content of the current point select, (4) Ranked list of 20 documents positioned near to the query, (5) A pop-up to view the content of document from the Ranked List, (6) Query Term Occurrence shows the group of terms that occur in the documents from the query, (7) Filtered documents on scatterplot based on the selected terms from the query term occurrence and (8) Expansion term suggestion in order of highest to lowest probability based on our query expansion model. 16

4.1 MAP and NDCG score averaged over 3 runs for different number of feedback documents given for Query Expansion Models. 29

4.2 Initial search results for the given user query. Top-10 expansion terms are suggested based on the user query and feedback. . . 31

4.3 Second iteration, after submitting the select list of query expansion terms the document-query space re-renders. 32

4.4 Initial search results for the given user query. Top-10 expansion terms are suggested based on the user query and feedback. . . 34

4.5 Initial search results for the given user query. Top-10 expansion terms are suggested based on the user query and feedback. . . 35

Abstract

Query Expansion is an effective approach for improving the information retrieval (IR) system's performance as it addresses the vocabulary mismatch and distinct terminology issues. Traditional pseudo relevance feedback (PRF) based query expansion models assume that top-k retrieved documents to be positive feedback from which the expansion terms are selected. This approach might add terms out of context if the initial retrieved list contains a significant number of negative documents. Therefore, it is equally important to consider negative feedback along with positive feedback. Moreover, it has been observed that the terms suggested by the global expansion techniques, such as word-embeddings, are different from the local expansion technique. The proposed hybrid query expansion technique combines a word embedding model with the positive and negative feedback model based on the Expectation-Maximization algorithm. The experiment conducted on the CACM dataset demonstrates that integrating the global and local expansion techniques enhances the system's performance over the baselines. Subsequently, we provide an interactive visual interface assisted by the proposed hybrid query expansion techniques. Unlike static vector-space models like TF-IDF and Doc2Vec, this interface represents documents based on the relevance score with the other documents in the space. The document-query space is adaptive as query and expansion terms weights are added, based on whether they appear in the document. The other document terms are weighted according to their TF-IDF value. Moreover, the representation also adapts based on the user relevance feedback provided. The user scenario illustrates the visual interface's usefulness for navigating, analyzing, and providing feedback to the document in a large query-document space. The results confirmed that the system's adaptive nature, influenced by the expansion terms and user feedback, can improve the ranked list based on the documents closest to the query.

List of Abbreviations and Symbols Used

EM	Expectation-Maximization.
KL divergence	Kullback–Leibler divergence.
MAP	Mean Average Precision.
NDCG	Normalized Discounted Cumulative Gain.
PRF	Pseudo Relevance Feedback.
t-SNE	t-Distributed Stochastic Neighbor Embedding.
TF-IDF	Term Frequency-Inverse Document Frequency.
VSM	Vector Space Model.

Acknowledgements

In the process of writing this thesis, I received good support and assistance.

Firstly, I would like to thank my supervisor, Dr. Fernando Paulovich, whose expertise was invaluable in formulating the research questions and methodology. It helped me to see things differently and improved my work because of his insightful feedback and suggestions.

I would like to thank my co-supervisor, Dr. Elham Etemad, for her patient support and for all of the valuable guidance I was given to further my research.

I would also like to thank my friends and colleague at Waterford Energy Services Inc. and Dalhousie University for their support and motivation. This work was supported by Mitacs through the Mitacs Accelerate program.

Thanks to my family for giving me this opportunity to pursue my Masters and being supportive even though I was far away from home.

Chapter 1

Introduction

In domain-specific text retrieval systems, such as medical and law, where the decisions rely on the previous cases to be studied, retrieving all relevant documents for a given user query is essential [43]. However, often the user is not aware of the document space or terminology used in the collection. Consequently, this makes it hard for the user to formulate a proper query [2] or the user tends to search with short queries (the average size of the query was 2.4 words [33]), which makes it difficult for the system to find the relevant documents that the user expects. It is mainly due to the query lacking the important terms required for retrieving the relevant document or vocabulary mismatch, causing traditional information retrieval algorithms to fail [6]. Query expansion is a widely used approach to overcome this challenge and build a system that can retrieve the maximum relevant document to the given user query.

The query expansion models based on pseudo-relevance feedback (PRF) require an initial retrieved list. The model assumes the top-k documents as positive documents to the query and suggests expansion terms with their weights. In contrast, expansion techniques using the word-embedding model suggest terms based on semantically similar query terms and do not require initially retrieved documents. In a study by Saar Kuzi et al., it was observed that top query terms suggested based on the word-embedding model differ from the pseudo-relevance feedback model [21]. Hence, the relevance feedback model was integrated with the word-embedding model to leverage the performance [21]. However, the pseudo relevance query expansion technique might deviate from the original query if an initial retrieved list of documents contains significantly more negative documents than positive ones. In such cases, assuming top-k documents to be positive might suggest off-topic expansion terms.

For this reason, it is necessary to filter the positive and negative documents to a query in order to suggest better query expansion terms. An Expectation-Maximization (EM) algorithm based model for positive and negative feedback was used to weigh the expansion terms [14]. In our work, we integrate the word-embedding model with the positive and single negative feedback models. The single negative feedback model seems to suggest more efficient terms over multiple negative feedback models [38] [14]. Furthermore, the query must be enriched with relevant terms from the suggested term [6]. Our system gives the user access to select the feedback and expansion terms through a visual interface.

The visualization of information retrieval is a highly researched area; it helps the user comprehend the ranking of the documents [15]. Besides, it helps visualize large numbers of documents on a screen [26]. A user can navigate through documents based on different filters or an interactive visualization that assists the user to re-rank the documents [18] or to provide relevance feedback. According to a study by Amanda Spink et al. [33], half of the users only visited the first two results web pages of results consisting of 10 websites each. It is hard to conclude if the results were perfect or the users gave up exploring the content. However, based on the short queries, a retrieval system with better precision is required. Through this work, we have designed a visualization system that allows users to analyze retrieved results and provide positive feedback if the document is relevant and negative feedback if it is not. An effective query expansion model employing the interface allows complete user control over what expansion terms should be selected and a convenient way to analyze and provide feedback to the document.

1.1 Research Problem and Hypothesis

Traditional IR methods such as Term Frequency-Inverse Document Frequency (TF-IDF) and BM25 are based on terms matching; therefore, these algorithms face vocabulary mismatch and distinct terminology issues. Query expansion is a widely used approach to address this challenge. However, pseudo relevance feedback based on the assumption that top-k retrieved documents are positive might deviate the query if the retrieved list has significant irrelevant documents. Also, considering the negative and positive feedback might not be sufficient. As in the case when there are only negative feedback documents in the top-k retrieved list, the terms would only be penalized. In this scenario, including word-embedding would suggest terms semantically similar to the query terms.

Though the query expansion enhances overall system performance, there are still uncertainties and high variances across queries [7]. Moreover, the approach considering relevance feedback has a challenge of poison pills [35], where the expansion terms suggested from the positive documents still deviate the query and decrease precision. Hence, providing an interface which allows user to provide feedback and select the suggested expansion terms will address the current query expansion challenges.

For representing the documents, the commonly used approach is the vector space model such as TF-IDF and Doc2Vec. This representation has a static document space. Here, creating a visual interface representing documents in terms of relevance score will make the document-query space adaptive. As the expansion term(s) selected would update the relevance score representation, and relevance feedback provided will modify the feedback documents with respect to the query.

1.2 Contribution

This thesis focuses on hybrid query expansion model and visualization of document-query space. Our two main contributions are:

- We propose a query expansion technique that is a hybrid model of global and local query expansion techniques. We use the positive and negative feedback model for the local query expansion technique based on user relevance feedback. To generate the probabilities for the local query expansion models, we use the EM algorithm. The EM models assume that the context term distribution influences positive term distribution, whereas negative term distribution is impacted by both positive and context term distribution [14]. The local expansion models are then integrated with the global expansion technique formed using a word-embedding technique. The global expansion approach finds the centroid of the query terms in a word-embedding model for suggesting expansion terms. Finally, the probabilities of expansion terms generated from the hybrid query expansion model are merged with the query to formulate a new weighted query.

- Subsequently, we propose an adaptive visual interface assisted by the hybrid query expansion technique. The visual interface represents the documents containing one or more query terms based on the relevance score with other documents and the user query in the space. The user can view the document's content and interact with the system to analyze a particular document and give positive or negative feedback. The system will suggest the query expansion terms which the user can use to refine the query. In the proposed document-query space, the document vectors are adaptive based on the expansion terms selected by the user. Additionally, the document-query space also takes user feedback to re-position the documents.

1.3 Thesis outline

The remainder of this work is structured as follows. Chapter 2 provides the background about query expansion approaches and visualization in information retrieval. Chapter 3 gives an overview of the system and the model framework proposed. Chapter 4 describes the experiments to evaluate the effectiveness of the hybrid query expansion and discusses the use case and user scenario for the visual interface proposed. Finally, we present a summary of this work in Chapter 5, discuss the limitations, and propose some ideas for future work.

Chapter 2

Related Work

The goal of a text retrieval system is to retrieve relevant documents from the collection of documents for a given user query. Specifically, in the exploratory search process [25] the user usually starts with an open-ended query to seek information from the collection. It involves the user interaction with the system to refine the search results according to the user's specific needs by iterating over a task of finding maximum relevant documents from the collection. This approach helps the user to investigate and retrieve the desired documents from the collection. Every exploratory system has a user interface as a fundamental element to provide iterative user interaction with the system [16]. Our literature review focuses on query expansion assisted information retrieval models and visualization approaches for information retrieval.

2.1 Query Expansion in IR

Considering the original query lacks the information or a vocabulary mismatch to retrieve the relevant documents, selecting good terms for expanding or reformulating the query is necessary. It impacts the retrieval performance of the term matching IR algorithms significantly.

Query expansion is widely used with different information retrieval models such as vector-space model [6], probabilistic models, language model [41]. Query Expansion techniques add meaningful terms to the query, which helps remove the ambiguity, add more details to the original user query, and fetch the relevant documents that do not contain the exact terms from the original user query [20]. Query Expansion can be categorized into Global and Local Query Expansion techniques [39].

2.1.1 Local Query Expansion

The local query expansion technique selects the expansion terms from the documents listed in the initial search result. Expansion terms can either be selected based on relevance feedback or pseudo relevance feedback. Relevance Feedback [11] asks the user to assess the document's relevance to a given query. In contrast, in pseudo-relevance feedback [39], the top-k documents are considered as positive documents to the query for improving the performance of a text-based retrieval system.

Rocchio [30] highlighted the system performance improvement using query expansion technique through user relevance feedback. It reformulated the initial query vector based on the user feedback in the vector space model (VSM). In this approach, the new query vector might move away from the already nearby relevant documents if the other relevant documents are away from the initial query.

The pseudo relevance feedback (PRF) process, on the other hand, assumes the top-k retrieved documents as relevant documents for the given query [23]. It is a commonly used method for dealing with the problem of vocabulary mismatch. In this approach, the probabilities of the expansion terms are generated from the top-k documents that are assumed to be positive feedback documents. A query model is formed, which integrates the count of terms from the query and the expansion terms probabilities from the feedback document. KL divergence is used to rank the documents with respect to the query model generated in the second iteration. Other approaches to select the dominant relevant documents [1] or classifying positive and negative documents [14] for the PRF method have been proposed. The documents are clustered based on the frequent local words to find dominant PRF documents in a novel pseudo significant input documents collection based on clustering method [1]. Nevertheless, using PRF, the expansion terms were sometimes misleading, and very few terms were relevant to the user query [6]. As there are cases where for a particular query, there is minimal or no relevant document in the top-k documents out of the retrieved list.

Henceforth, it is equally essential to consider the negative feedback documents together with the positive feedback documents. The negative feedback model [38] is proposed to improve the efficiency in case of difficult queries. The model was evaluated only for difficult queries with at most three relevant documents in the top-10 initially retrieved result.

The positive and negative feedback models were integrated [14] and estimated using the Expectation-Maximization (EM) [42] algorithm. In this approach, the expansion term probabilities are computed by assuming that positive term distribution relies on the context model and negative term distribution relies on both positive and context models. Also, it includes Kullback–Leibler (KL) divergence [22] to identify whether the document is a positive or a negative feedback document. However, this approach requires more top-k feedback documents from the initially retrieved list to improve their feedback model's performance. Besides, identifying whether the document is positive or negative is dependent on the number of negative feedback documents in initial retrieval. Moreover, the KL divergence model is based on the assumption that negative documents would contain heterogeneous data. Hence it is necessary to evaluate the documents selected as positive and negative feedback.

Even though automatic query expansion techniques based on pseudo relevance feedback perform well on average, they have uncertainty and high variance across the queries [7]. Moreover, sometimes relevant feedback documents may act as "poison pills" [35] and impact the model's performance by automatically adding a certain number of expansion terms. Hence, in this thesis, we provide an interface for the user to select the expansion terms.

2.1.2 Global Query Expansion

Unlike local query expansion, the expansion terms in the global query expansion technique are selected from a knowledge resource such as a thesaurus or corpus of the entire collection or are semantically related to the query. Popular global query expansion approaches use methods such as thesaurus [4], ontology [3], WordNet [27] and word embeddings [10].

It was observed that relevance feedback or pseudo-relevance feedback produces more effective results compared to global analysis techniques based on linguistic resources, such as WordNet [37] [40]. In the case of word-embeddings, even though the expansion terms are semantically similar to the query, the terms might deviate the query. For example, the word embedding suggesting a query "programming language" might attract words such as "alphabets" and "English".

It is observed that the terms suggested by the word-embedding model (global) differ from the pseudo-relevance feedback model (local) [21]. A hybrid model which involves both global expansion technique based on word embedding and local query expansion techniques using the pseudo-relevance feedback [23] was proposed by Saar Kuzi et al. [21] to expand the initial user query. Another hybrid model based on the log-based expansion system integrates the query logs and the relevant documents chosen based on the user click on the documents in initial retrieval [8]. Nevertheless, these approaches do not take into account the negative feedback model. In our work, we integrate the word embedding model with positive and negative feedback models. This will help to enhance the suggested terms from both the global and local approaches as well as suggest terms when there is only negative feedback provided from the word-embedding model. Furthermore, we involve the user to give relevant feedback and select expansion terms from the suggested list.

2.2 Visual exploration for information retrieval

Interactive Intent Modeling [31] involves the user with the system. It helps in exploratory search where the system is not restricted just to the search textbox and button. Moreover, an efficient machine learning model with interaction could handle user noise and enhance the system's performance.

In an information retrieval system, visual interfaces analyze the initial set of results retrieved and give a user an insight into how the query terms impact the ranking of the documents. This would help the user give better relevance feedback, refine the query, and get a better set of desired results. Text-retrieval documents can be analyzed by highlighting the terms or features matched by the query and then displaying each word's intensity through visualization, such as TileBars [15]. For facet queries, TileBars helps visualize how query terms are distributed in documents in a compact manner. TileBars is based on frequency term matching and does not consider vocabulary mismatch, and it is scaleable for searching few terms but not for long queries. DeepTileBars [34] is a neural approach inspired by TileBars. In this approach, term features are visualized between query and document segment. The interaction matrix generated is then fed to the neural model to obtain the final ranking score.

VIBE [26] is another frequency-based visualization technique where the documents are positioned based on the similarity ratio with the query terms known as Points of

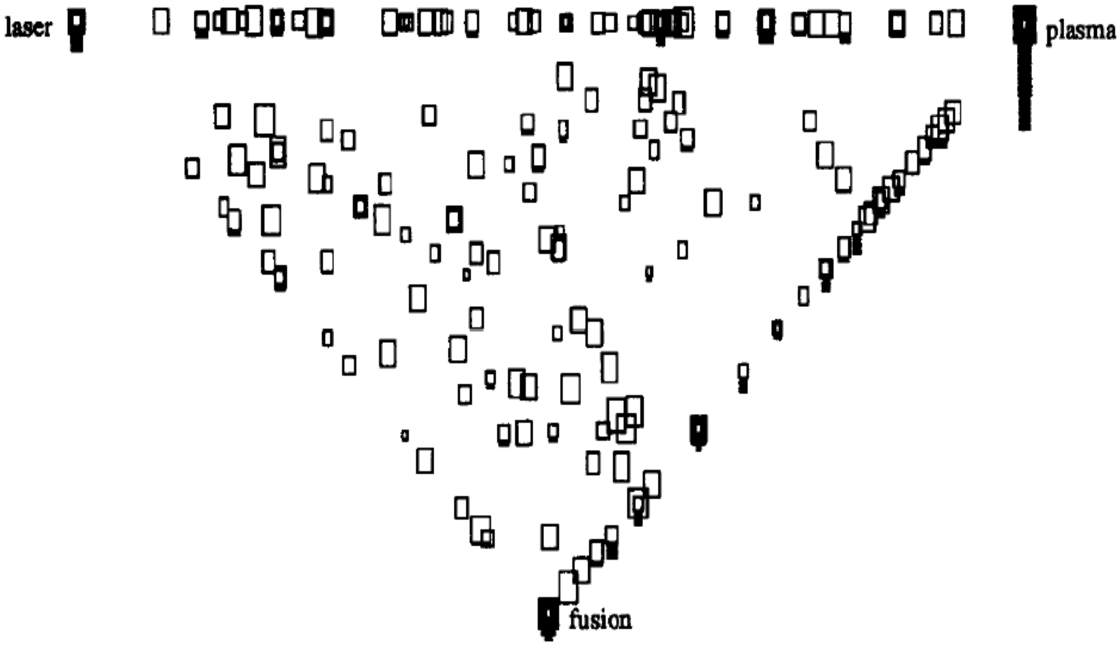
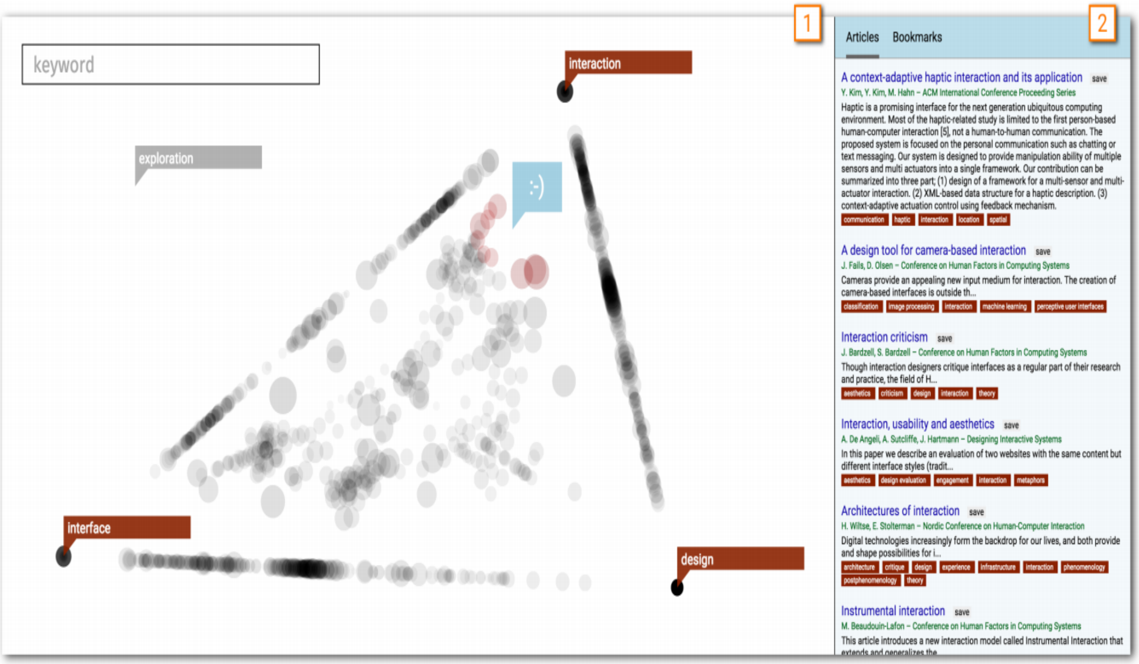


Figure 2.1: VIBE system showing hundreds of documents. Documents are positioned based on the similarity ratio with the query terms known as Points of Interest. [26]

Interest (POIs). VIBE is an interactive tool that can show hundreds of documents, as illustrated in Fig. 2.1 and re-position the POIs and add or remove POIs. However, it was hard to analyze the region if the documents were placed close by, and it did not support the re-ranking of the documents on the change of POIs position in the map.

VIBE's shortcomings have been addressed by Visual Re-Ranking for Multi-Aspect Information Retrieval [18], which enables a user to analyze the query terms and refine the document selection by selecting a point on an interactive relevance map interface. The user can set the position of the query terms in 2-D space, and documents are positioned accordingly with respect to the query terms, as shown in Fig. 2.2. In addition, the user can also activate or deactivate the query phrase to analyze the effect of different query terms.

WebRat [13] and VisElabor [19] allow and assist the user to refine the query. In WebRat Visualization [13], the user starts with a general query, and the document results are visualized using dynamic, interactive clustering. The labels are generated on the fly based on the concentration of the documents. Users can select the topic



A context-adaptive haptic interaction and its application

Y. Kim, Y. Kim, M. Hahn – ACM International Conference Proceeding Series

Haptic is a promising interface for the next generation ubiquitous computing environment. Most of the haptic-related study is limited to the first person-based human-computer interaction [5], not a human-to-human communication. The proposed system is focused on the personal communication such as chatting or text messaging. Our system is designed to provide manipulation ability of multiple sensors and multi actuators into a single framework. Our contribution can be summarized into three part; (1) design of a framework for a multi-sensor and multi-actuator interaction. (2) XML-based data structure for a haptic description. (3) context-adaptive actuation control using feedback mechanism.

A design tool for camera-based interaction

J. Fails, D. Olsen – Conference on Human Factors in Computing Systems

Cameras provide an appealing new input medium for interaction. The creation of camera-based interfaces is outside th...

Interaction criticism

J. Bardzell, S. Bardzell – Conference on Human Factors in Computing Systems

Though interaction designers critique interfaces as a regular part of their research and practice, the field of H...

Interaction, usability and aesthetics

A. De Angeli, A. Sutcliffe, J. Hartmann – Designing Interactive Systems

In this paper we describe an evaluation of two websites with the same content but different interface styles (tradit...

Architectures of interaction

H. Wiltse, E. Stolterman – Nordic Conference on Human-Computer Interaction

Digital technologies increasingly form the backdrop for our lives, and both provide and shape possibilities for i...

Instrumental interaction

M. Beaudouin-Lafon – Conference on Human Factors in Computing Systems

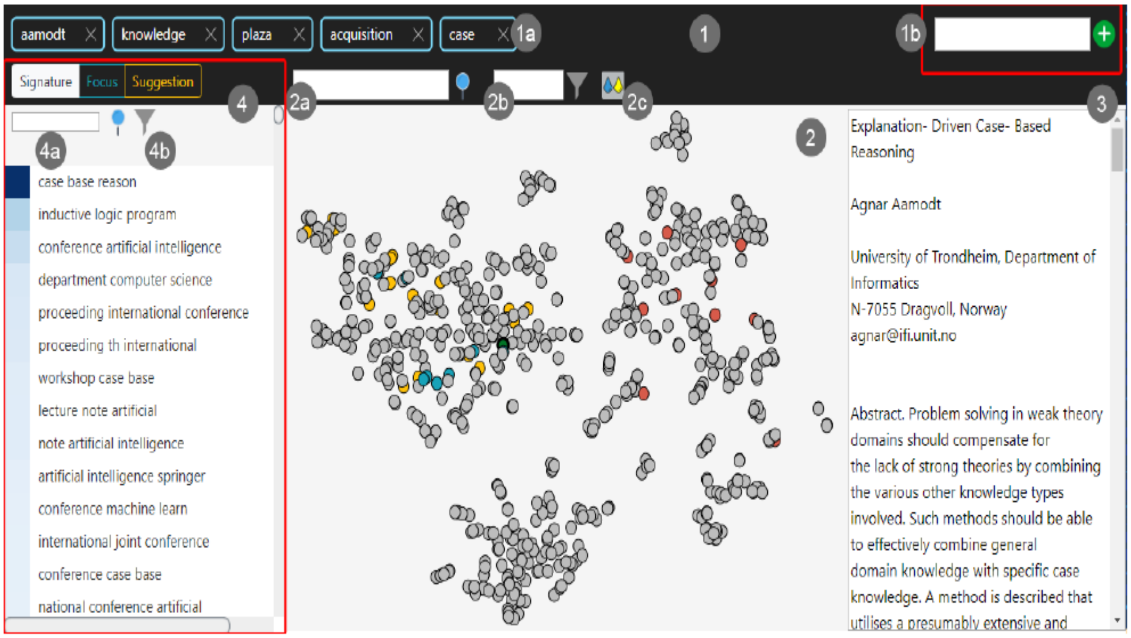
This article introduces a new interaction model called Instrumental Interaction that extends and generalizes the

Figure 2.2: Visual Re-Ranking for Multi-Aspect Information Retrieval: The interactive relevance map is rendered based on three topics selected shown in the red label, and one topic disabled is greyed out. The user can select the exploration cursor (in blue) to select the region for re-ranking the documents. Documents rendered as red points are the top documents retrieved in rank list [18].

labels from the visualization to refine the query. VisElabor [19] for the elaboration of search results consists of four views: list view (ordered list of hits based on the search engine ranking), category view (ordered list of categories (clusters) based on cluster size), graph view (represents the relationship among the search results) and full-text view (selected document in a browser window). Users can change the cluster's threshold value to refine the query and get the result of the specific topic of interest. This provides the various possible views of the same data.

The traditional approach for visualizing the document space is projection-based approaches that project the multi-dimensional documents represented by TF-IDF or Doc2Vec [24] using linear or non-linear dimensionality reduction approaches to represent the documents in 2-D or 3-D space.

TRIVIR uses techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) [36] and Least-Square Projection to plot the documents in a 2-dimensional space. The interface proposed in TRIVIR, where documents are projected on a


aamodt
knowledge
plaza
acquisition
case
1a
1
1b
Signature
Focus
Suggestion
4
2a
2b
2c
3
4a
4b
2
case base reason
inductive logic program
conference artificial intelligence
department computer science
proceeding international conference
proceeding th international
workshop case base
lecture note artificial
note artificial intelligence
artificial intelligence springer
conference machine learn
international joint conference
conference case base
national conference artificial

Explanation- Driven Case- Based Reasoning

Agnar Aamodt

University of Trondheim, Department of Informatics
N-7055 Dragvoll, Norway
agnar@ifi.unit.no

Abstract. Problem solving in weak theory domains should compensate for the lack of strong theories by combining the various other knowledge types involved. Such methods should be able to effectively combine general domain knowledge with specific case knowledge. A method is described that utilises a presumably extensive and

Figure 2.3: TRIVIR interface: The scatterplot view (2) projects the documents based on TF-IDF or word-embeddings representation using the projection technique t-SNE (also supports least-square projection). The interface provides various views to filter the documents [9]

scatterplot, is shown in Fig. 2.3. The circles plotted with green, blue, yellow, and red color represent query document, positive user feedback, negative user feedback, and positive feedback suggested by the system using a machine learning algorithm. This interface allows the user to interact with the machine learning algorithm for suggesting the relevant documents and had a static document collection space based on TF-IDF or word-embeddings.

In this thesis, we propose an adaptive visualization document-query relation space. The document is represented as the vector of relevance scores with the query and other documents containing one or more query terms. The visualization helps the user to explore the document and give feedback to the hybrid query expansion model. After that, the user can select or deselect the terms from the listed expansion terms; they are all selected by default. The visualization adapts according to the expansion terms selected and the user relevance feedback provided.

Chapter 3

Model Framework

This chapter describes the system flow and the model framework of the proposed visual interface, assisted by hybrid query expansion.

3.1 Overview

Given a user input query Q which consists of terms $t_Q = \{t_{Q1}, t_{Q2}, t_{Q3}, ..., t_{Qy}\}$ and a collection of documents $D = \{D_1, D_2, D_3, ..., D_M\}$, where a document D_i from collection D consists of terms $t_{Di} = \{t_{Di1}, t_{Di2}, t_{Di3}, ..., t_{Diz}\}$. Consider S a subset of documents from the document collection D which consists of only those documents that contain one or more terms from the query terms t_Q, $S = \{D_1, D_2, ..., D_N\}$.

BM25 (Best Matching) [29] is the standard retrieval algorithm used for retrieving ranked list of relevant documents from a collection of documents D using query Q. BM25 score of a document D_i for a given query Q is computed as:

$$\text{score}(Q, D_i) = \sum_{x=1}^{y} \text{term\_score}(t_{Qx}, D_i) \tag{3.1}$$

$$\text{term\_score}(t_{Qx}, D_i) = \text{IDF}(t_{Qx}) \cdot \frac{f(t_{Qx}, D_i) \cdot (k_1 + 1)}{f(t_{Qx}, D_i) + k_1 \cdot (1 - b + b \cdot \frac{|D_i|}{\text{avgdl}})} \tag{3.2}$$

where $f(t_x, D_i)$ is Q's term frequency in the document D_i from the collection D, $|D_i|$ is the length of the ith document in words, avgdl is the average document length in the text collection D from which documents are drawn. k_1 and b are free parameters. $IDF(q_i)$ is the IDF (inverse document frequency) weight of the query term t_{Qx}. It is computed as [28]:

$$IDF(t_{Qx}) = 1 + \log(\frac{N + 1}{n(t_{Qx}) + 1}) \tag{3.3}$$

where N is the total number of documents in the collection, and $n(t_{Qx})$ is the number of documents containing t_{Qx}

User queries are shorter and lack information, due to which term-matching algorithms are not able to retrieve relevant documents. Hence, query expansion is a popular approach to add more details to the query and overcome vocabulary mismatch problems.

Pseudo-Relevance feedback is a widely used approach with different IR algorithms to get the query expansion terms based on the initial retrieved results. It considers the top-k documents to be relevant to the query and suggests terms. However, in case of some queries, it might be possible that there is no relevant document in the initial retrieval [38]. Hence it is essential to consider the positive feedback model and negative feedback model combination to get the expansion terms. Pseudo-relevance feedback produces a more effective approach than global analysis techniques based on linguistic resources, such as WordNet techniques [40]. Top terms suggested by the word-embedding model differ from pseudo-relevance feedback [21]. Hence, integrating the word-embedding expansion model with the relevance model improved the query expansion's performance but did not account for the negative feedback model. In this thesis, we consider user relevance feedback, where $P = \{D_1, D_2, ..., D_K\}$ is a set of positive documents and $N = \{D_1, D_2, ..., D_L\}$ is a set of negative feedback documents assessed by the user. We integrate the positive and negative feedback model based on user relevance feedback which generates probabilities of the terms from the feedback documents using Expectation-Maximization (EM) algorithm with word embedding model for suggesting the most probable terms.

If the data is incomplete or the likelihood function has latent variables, the Expectation-Maximization algorithm is a general approach for maximum-likelihood estimation. In this thesis, we use the EM algorithm for estimating the probabilities of the terms. We have the known variable context model that is the distribution of the terms in the given collection. For the positive feedback model, we use an EM model to compute the latent variable: the distribution of positive terms given the positive feedback documents, and the known variable is the context model. In the negative feedback model, we use the EM model to compute the latent variable: the distribution of negative terms given the negative feedback documents. In this case, the known variables are the positive feedback model and the context model.

The weights generated by the positive and negative feedback model are used to re-weight the query terms and include the weights of the expansion terms to be suggested. These weights are then used in the Eq. 3.2 to get the weighted BM25 relevance score for each term in the query and expansion term list.

For the information retrieval interface, a standard approach to represent the documents is the vector space model (VSM) [32] of TF-IDF vectors or Doc2Vec on scatter plot using dimensionality reduction techniques such as t-SNE. This thesis proposes a visual interface where one document is represented in terms of the other documents from a collection containing one or more query terms and the query. The interface helps the user to analyze the documents and give feedback and select the suggested expansion terms. The document relation is re-weighted based on the expansion terms. Besides, it changes the query vector relevance score of positive and negative feedback documents. These factors make the document-query space adaptive to the expansion terms and user feedback to updates the representation.

The overview of the query expansion assisted visual interface is shown in Fig. 3.1. The proposed visualization matrix visualizes the document-query space that contains the user input query, Q, and documents from the S. The document-query space allows the user to interact with the interface and provide feedback documents. The hybrid expansion model suggests the expansion terms based on the user relevance feedback. Now, the user can select the expansion terms from the suggested list and re-search. The document-query space adapts based on the provided feedback documents and selected expansion terms.

3.2 Design Guidelines

In this section, we propose the design guidelines for our visual interface system based on the literature review for visual interface in information retrieval and query expansion models:

G1 - Representing documents from the collection that contains one or more query terms and the query in a document-query space and projecting it into a 2-Dimensional space. Here we project the documents as the vectors of relevance score with other documents.

G2 - To suggest expansion terms based on the initial retrieved list and global analysis techniques such as word-embedding. The positive and negative feedback

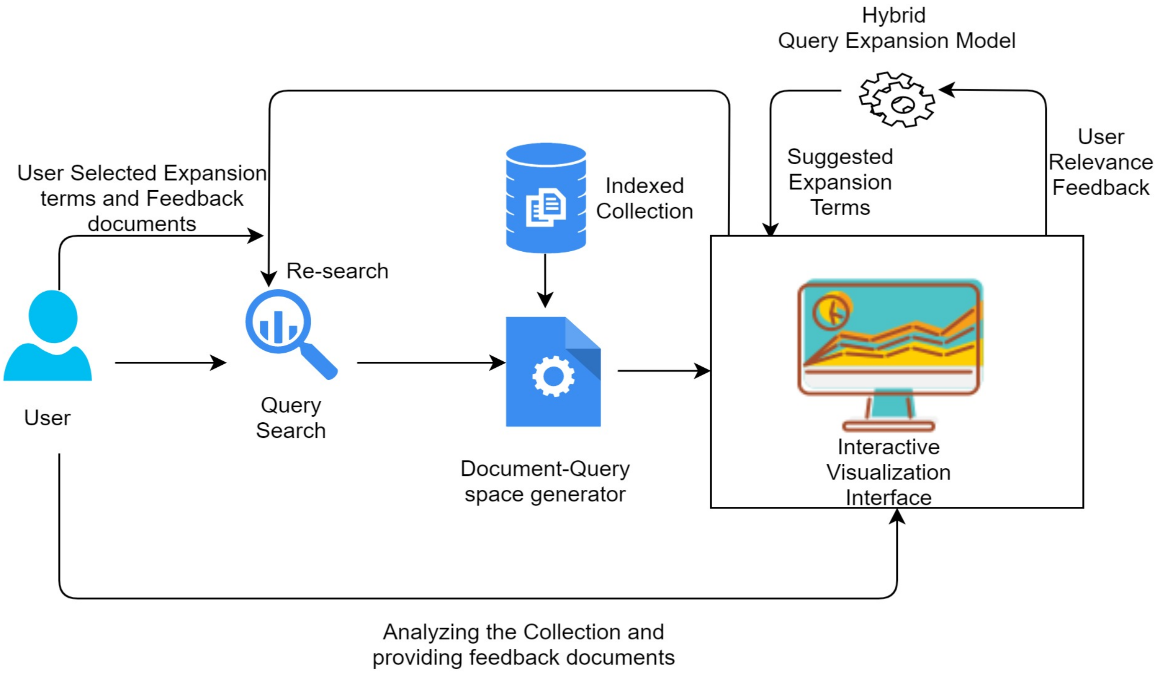

Hybrid
Query Expansion Model
User Selected Expansion terms and Feedback documents
Re-search
Indexed Collection
Suggested Expansion Terms
User Relevance Feedback
User
Query Search
Document-Query space generator
Interactive Visualization Interface
Analyzing the Collection and providing feedback documents

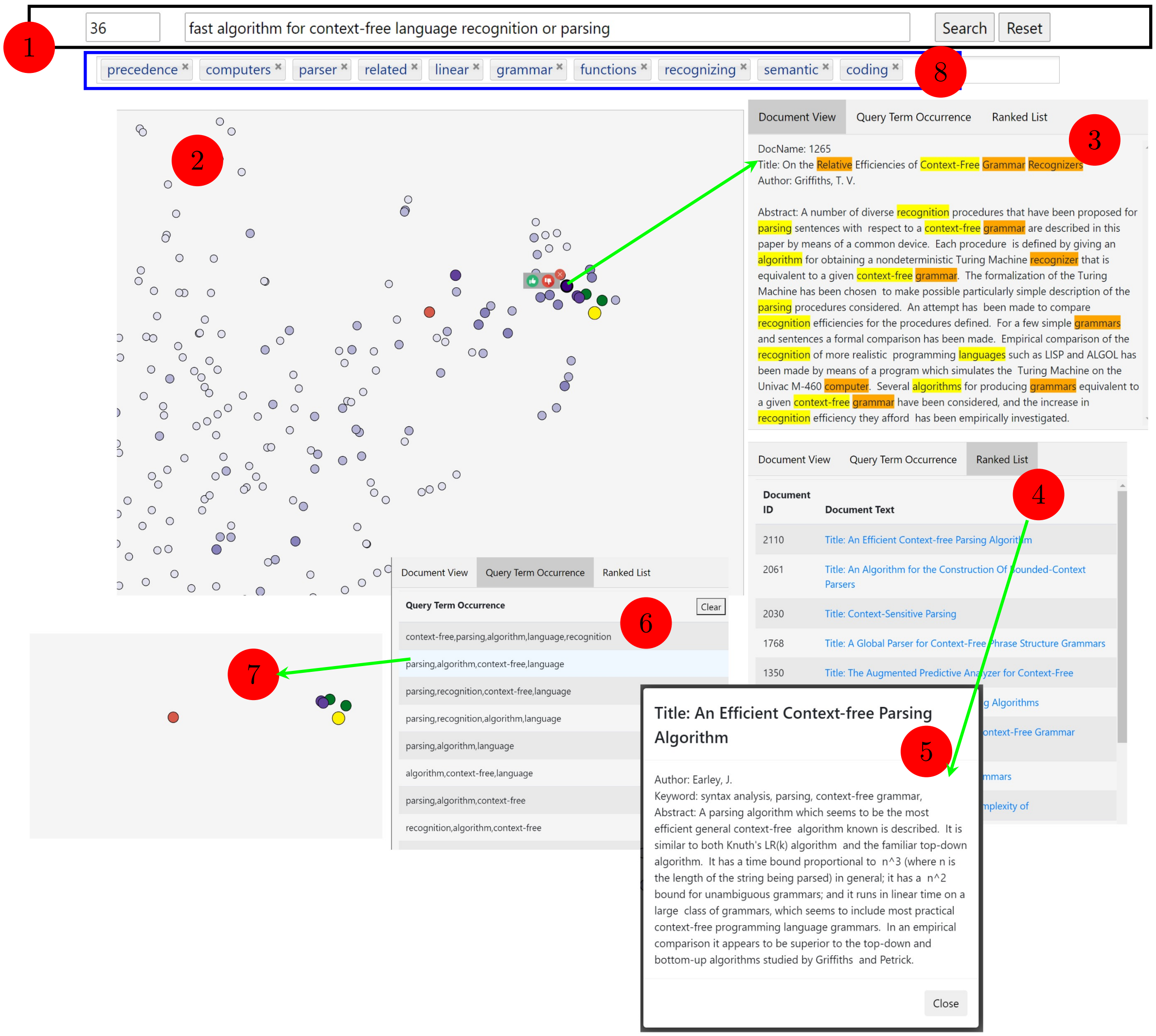
Figure 3.1: Framework of the Hybrid Query Expansion assisted Visual Interface

model integrated with the word-embedding model. Here, the positive and negative feedback model are based on EM algorithm. For word embedding model we compute the centroid of the query the query terms. The terms suggested by word embedding models are semantically similarly to the computed query centroid based on cosine distance.

G3 - A document-query space in G1 is adaptive to the expansion terms selected from G2 and the user relevance feedback. The document-query space is adaptive because the document's relevance score is determined as a function of its query terms and expansion terms, which are given higher weight than other document terms. Additionally, the user feedback impacts the representation by changing the relation of positive and negative feedback(s) with query. These two factors makes the document-query space adaptive to the user feedbacks and expansion terms.

3.3 System Flow

We propose an interface in this thesis with adaptive document-query space visualization and a hybrid query expansion model for implementing the design goals. The interface visualizes document collection S and query Q in a document-query space



DocName: 1265
Title: On the Relative Efficiencies of Context-Free Grammar Recognizers
Author: Griffiths, T. V.

Abstract: A number of diverse recognition procedures that have been proposed for parsing sentences with respect to a context-free grammar are described in this paper by means of a common device. Each procedure is defined by giving an algorithm for obtaining a nondeterministic Turing Machine recognizer that is equivalent to a given context-free grammar. The formalization of the Turing Machine has been chosen to make possible particularly simple description of the parsing procedures considered. An attempt has been made to compare recognition efficiencies for the procedures defined. For a few simple grammars and sentences a formal comparison has been made. Empirical comparison of the recognition of more realistic programming languages such as LISP and ALGOL has been made by means of a program which simulates the Turing Machine on the Univac M-460 computer. Several algorithms for producing grammars equivalent to a given context-free grammar have been considered, and the increase in recognition efficiency they afford has been empirically investigated.

Document ID	Document Text
2110	Title: An Efficient Context-free Parsing Algorithm
2061	Title: An Algorithm for the Construction Of Bounded-Context Parsers
2030	Title: Context-Sensitive Parsing
1768	Title: A Global Parser for Context-Free Phrase Structure Grammars
1350	Title: The Augmented Predictive Analyzer for Context-Free
	g Algorithms
	ontext-Free Grammar
	mmars
	mplexity of

Query Term Occurrence

context-free,parsing,algorithm,language,recognition
parsing,algorithm,context-free,language
parsing,recognition,context-free,language
parsing,recognition,algorithm,language
parsing,algorithm,language
algorithm,context-free,language
parsing,algorithm,context-free
recognition,algorithm,context-free

Title: An Efficient Context-free Parsing Algorithm

Author: Earley, J.
Keyword: syntax analysis, parsing, context-free grammar,
Abstract: A parsing algorithm which seems to be the most efficient general context-free algorithm known is described. It is similar to both Knuth's LR(k) algorithm and the familiar top-down algorithm. It has a time bound proportional to n^3 (where n is the length of the string being parsed) in general; it has a n^2 bound for unambiguous grammars; and it runs in linear time on a large class of grammars, which seems to include most practical context-free programming language grammars. In an empirical comparison it appears to be superior to the top-down and bottom-up algorithms studied by Griffiths and Petrick.

Figure 3.2: Overview of our framework. The numbers indicate the different sections of the visual interface. (1) Search Bar, (2) scatterplot of documents and query, (3) Document View displaying the document content of the current point select, (4) Ranked list of 20 documents positioned near to the query, (5) A pop-up to view the content of document from the Ranked List, (6) Query Term Occurrence shows the group of terms that occur in the documents from the query, (7) Filtered documents on scatterplot based on the selected terms from the query term occurrence and (8) Expansion term suggestion in order of highest to lowest probability based on our query expansion model.

based on Q and other documents in S (Design goal G1). The interface is adaptive since the vector presents a document as a relevance score with all the other documents. The relevance score is calculated by giving more weight to the terms in the query and the expansion terms selected in comparison to the other terms in the document (Design goal G3).

The interface view is presented in Fig. 3.2. The user starts with the desired query to find the documents of interest from the collection using the search bar (1). The user clicks the search button. The document-query space scatterplot renders all the documents from S containing one or more query terms, and the query Q (2). In this scatterplot, the query point is represented by a yellow color point, and the documents are colored with different shades of purple based on the number of terms matched from the query. The user can click on any point from the space and view the document content in the document view (3). In the document view, the query terms and expansion terms are highlighted in yellow and orange, respectively. The system has a ranked list of documents near the query point sorted based on Euclidean distance from the query (4). This ranked list shows the title of the documents, and the user can click the title to see the content of the document displayed in a pop-up window (5). The system provides the user an ability to filter specific documents to be rendered on the scatterplot based on the group of query terms that occurs in the document (6). These views will help the user analyze the documents.

The user can assess the documents by giving positive feedback if the document is relevant to the user query or negative feedback if the document is not relevant. A pop-up with an up-vote and down-vote button to provide feedback is shown when the user clicks on a particular document point displayed in a scatterplot (2).

The system suggests top-10 expansion terms to the user based on the feedback document(s) and word-embedding model (8). By default, all the ten expansion terms are selected. The user can remove the terms or add them again from the default suggested list of expansion terms and re-search based on the query and new set of expansion terms (Design goal G2). The document representation is updated as the relation of the document regarding the other documents. A dynamic representation is used for the document-query space instead of a static representation based on TF-IDF or Doc2Vec. The visual interface is adaptive as the matrix is influenced by the expansion terms selected and the feedback documents provided by the user.

3.4 Model Framework

3.4.1 Local Query Expansion Model

The feedback model, which is a combination of positive and negative feedback, improves the retrieved results compared to pseudo relevance feedback, considering only positive feedback [14]. Here, a document model is represented by the unigram language model or the term distribution. The positive term distribution or feedback model is based on context term distribution, whereas the negative term distribution is based on positive and context term distribution.

Expectation-Maximization (EM) Algorithm: To compute the term distribution for negative and positive feedback documents we use two feedback models based on EM algorithm: positive feedback model and a negative feedback model.

The general procedure of the EM algorithm is as follows [42]:

- Initializing the latent variable $\theta^{(0)}$ randomly or heuristically based on some prior knowledge about where the optimal parameter value might be. Here we use normalized term frequency for each term from the given set of positive or negative documents as a heuristic.

- Alternate between the following two steps to improve the estimated latent variable $\theta^{(i-1)}$:

  Step 1: The E-step (expectation): Compute term distribution for positive feedback model θ_P or negative feedback model θ_N.

  Step 2: The M-step (maximization): Re-estimate the term distribution for θ_P or θ_N by maximizing the likelihood function:

- Stop when the likelihood function converges.

Context Model: The context model θ_C illustrates the contextual information of documents by representing the distribution of common terms in the corpus. $p(t|\theta_C)$

which is formulated as:

$$p(t|\theta_C) = \frac{c(t,C)}{\sum_{t' \in V} c(t',C)} \tag{3.4}$$

where c(t,C) is frequency of the term in the C and V represents list of all the terms in corpus.

Positive Feedback Model: The EM model computes the probability distribution of terms with respect to the positive feedback documents by iterating over the expectation (E-step) and maximization (M-step) step until the log-likelihood function converges. Initially, the latent variables are set randomly or heuristically. Here we set the variables as the term frequency in the positive feedback documents normalized all the term frequency in the positive feedback documents as shown in Eq. 3.6.

Estimation step (E-step) computes the expected likelihood for the complete data where the expectation is taken with respect to the computed conditional distribution of the latent variables that is positive feedback model θ_P given the observed data θ_C.

E-Step (Expectation):

$$H_t^{(n)} = \frac{\lambda p^{(n)}(t|\theta_P)}{\lambda p^{(n)}(t|\theta_P) + (1-\lambda) p^{(n)}(t|\theta_C)} \tag{3.5}$$

Here λ is a constant between [0,1) for controlling the ratio of positive feedback model and context model.

To compute E-step we need to initialize the hidden variables. Here, $P^0(t|\theta_P)$ is initialized as:

$$P^0(t|\theta_P) = \frac{\sum_{j=1}^{n} c(t|d_j)}{\sum_i \sum_{j=1}^{n} c(t_i|d_j)} \tag{3.6}$$

Here $c(t|d_j)$ is the frequency of a term t in the document d_j from the set of documents in P.

In the maximization step (M-step) we re-estimate the term distribution for positive feedback model θ_P by maximizing the log-likelihood function.

M-Step (Maximization):

$$P^{(n+1)}(t|\theta_P) = \frac{\sum_{j=1}^{n} c(t|d_j) H^{(n)}(t)}{\sum_i \sum_{j=1}^{n} c(t_i|d_j) H^{(n)}(t_i)} \tag{3.7}$$

Log-Likelihood for the term distribution from the given positive feedback model is maximized over the iterations:

$$\log p(t|\theta_P) = \sum_{d \in P} \sum_{t \in V} c(t,d) \log[\lambda p(t|\theta_p) + (1-\lambda) p(t|\theta_c)] \tag{3.8}$$

Here θ_P is estimated by maximizing the log-likelihood function, and it focuses more on discriminative positive terms and removes the background noise or common terms generated by the context model.

Negative Feedback Model: There are two types of negative feedback model: single and multiple negative feedback models. In single negative feedback, the model generates the probability distribution over all the negative feedback document(s). On the other hand, there is one EM model for each negative feedback document in multiple negative feedback models, and maximum probability is considered for each term.

In our work, we select a single negative feedback model. As in the multiple negative models, the feedback documents might be of different topics and would capture negative terms from each feedback document individually [38] [14]. Also, in this thesis, we do not consider negative probabilities to penalize the relevance score.

Expectation-Maximization model is used to generate the probabilities $p(t|\theta_N)$ where t is the candidate term and θ_N is the negative feedback model.

Estimation step (E-step) computes the expected likelihood for the complete data where the expectation is taken with respect to the computed conditional distribution of the latent variables that is positive feedback model θ_N given the observed data θ_C and θ_P (obtained from the positive feedback model).

E-Step (expectation):

$$L^{(n)}(t) = \frac{\gamma_N p(t|\theta_N)}{\sum_x \gamma_x p(t|\theta_x)} \tag{3.9}$$

Here x is in [P, N, C].γ_x is the constant for adjusting the proportions of the positive model θ_p, negative model θ_n and context model θ_c contributing to the probability $\sum_x \gamma_x = 1$.

To compute E-step we need to initialize the hidden variable. Here, $P^0(t|\theta_N)$ is initialized as:

$$P^0(t|\theta_N) = \frac{\sum_{j=1}^{n} c(t|d_j)}{\sum_i \sum_{j=1}^{n} c(t_i|d_j)} \tag{3.10}$$

Here $c(t|d_j)$ is the frequency of a term t in the document d_j from the set of documents N.

In the maximization step (M-step) we re-estimate the term distribution for negative feedback model θ_P by maximizing the log-likelihood function.

M-Step (maximization):

$$P^{(n+1)}(t|\theta_N) = \frac{\sum_{j=1}^{n} c(t|d_j)L^{(n)}(t)}{\sum_i \sum_{j=1}^{n} c(t_i|d_j)L^{(n)}(t_i)} \tag{3.11}$$

Log-Likelihood for the term distribution from the given negative feedback model is maximized over the iterations:

$$\log p(t|\theta_N) = \sum_{d \in N} \sum_{t \in V} c(t,d) \log[\sum_x \gamma_x p(t|\theta_x)] \tag{3.12}$$

Here, the negative terms are generated, and background terms from the positive and context model are eliminated.

3.4.2 Word Embedding Expansion Model

In word embeddings, it is possible to perform arithmetic operations on word vectors to get semantically related word vectors. This property is leveraged to get the expansion term using word embeddings. We use the centroid vector $\vec{q_{cent}}$ obtained from the query terms vector by taking their mean and normalizing it to a unit vector. Top n terms nearest to the centroid vector obtained from the query terms are selected from the Word2Vec model [21]. The probability of these n terms are computed as :

$$p(t|\vec{q_{cent}}) = \exp(cos(q_i, t)) \tag{3.13}$$

Now, from the word embedding model in Eq. 3.13 top v terms are fetched and sum normalized denoted by $P(t|\mathcal{M})$.

3.4.3 Hybrid Expansion Model

The obtained word-embedding model is then combined with the positive and negative Feedback Model based on EM algorithm using an interpolation parameter β. The final query expansion terms probabilities are generated using:

$$p(t|\theta_F, \mathcal{M}) = \beta_P * p(t|\theta_p) + \beta_W * P(t|\mathcal{M}) - \beta_N * p(t|\theta_N) \tag{3.14}$$

Here $\beta_P, \beta_W, \beta_N$ are the constants for adjusting the proportions of the positive feedback model θ_p, word-embedding model θ_c and negative feedback model θ_N respectively, contributing to the probability. Here $\beta_P + \beta_W + \beta_N = 1$.

Algorithm 1 Query Expansion Model

```
Input: User Query, Positive and Negative Feedback document.
Output: Probability weighted expansion and query terms.
  $t_Q$: terms in query Q
  $t_p$: Terms from Positive Feedback document(s) P
  $t_n$: Terms from Negative Feedback document(s) N
  $W$: Word Embedding model
  $prob\_positive$: $P(t_p|\theta_P)$ computed using Eq. 3.7
  $prob\_negative$: $P(t_n|\theta_N)$: computed using Eq. 3.11
  $\vec{q_{cent}}$: compute centroid of all the query words
  $t_w$: Top n terms similar to $\vec{q_{cent}}$ in W
  $prob\_word\_em$: $p(t_w|\vec{q_{cent}})$ : compute probability as shown in Eq. 3.13
  {Now, the $prob\_positive, prob\_negativeandprob\_word\_em$ are sorted and top-x
  terms with their probabilities are selected. Here top-x is the $threshold\_limit$.}
  $threshold\_limit$: Number of terms to consider after sorting based on probabilities
  from highest to lowest.
  {sum_normalize_data: sum normalize the probabilities}
  prob_positive = sum_normalize_data(prob_positive[0 to threshold_limit])
  prob_negative = sum_normalize_data(prob_negative[0 to threshold_limit])
  prob_word_em = sum_normalize_data(prob_word_em[0 to threshold_limit])
  $final\_map$: computed as shown in Eq. 3.14
  $positive\_map$: sum_normalize_data($final\_map$ where value gt 0)
  $negative\_map$: $final\_map$ where value lt 0
  Initialize: $final\_query$
  for every $t_i\ in\ set(t_Q + positive\_map)$ do
    if $t_i\ in\ t_Q$ then
      $term\_weight = t_Q.count(t_i) + positive\_map[t_i, 0] + negative\_map[t_i, 0]$
    else
      $term\_weight = positive\_map[t_i]$
    end if
    $final\_map[t_i] = term\_weight$
  end for
```

The proposed query expansion model is summarised in Algorithm 1. Based on the user feedback documents P and N and trained word-embedding W, top-k terms are suggested to the user. The negative probability distribution *prob_negative* obtained considers only the terms present in the query, positive feedback document, and word-embedding distribution. All the distributions are sum normalized after selecting the top terms sorted in descending order at a given threshold limit. The final probability map for query expansion is generated using Eq. 3.14. The obtained final map is then divided into positive term map and negative term map; the positive term map is sum normalized. The query terms are weighted based on their weight in a query, positive map, and negative map (the negative map has weights less than zero). After re-weighting the query top-n expansion terms from the positive map are suggested to the user.

3.4.4 Document Visualization

The document vector is represented in terms of relevance score with other documents in S and query Q. The documents in collection S and the query Q are converted to vector and then used to generate a matrix M.

Computation of the document vector D_i containing a set of terms t_d is illustrated in Algorithm 2. The t_d consists of top-k TF-IDF terms from the documents. Here K is set to the average document length of the collection. To make the system adaptive, we weigh the document terms present in the query by adding the term's count in the query with the TF-IDF of the term in that document, whereas other document terms are weighted based on their TF-IDF value.

If the user selects query expansion terms, then the query terms and expansion terms are weighted by adding the weight generated from the query expansion model in Algorithm 1. After the documents are converted to the weighted terms, they are used to calculate the relevance score using the BM25 score function in Eq. 3.1 with respect to the other documents in S and Q.

$$\text{M} = \begin{bmatrix} score(S_1, S_1) & score(S_1, S_2) & ... & score(S_1, S_N) & score(S_1, Q) \\ score(S_2, S_1) & score(S_2, S_2) & ... & score(S_2, S_N) & score(S_2, Q) \\ ... & ... & ... & ... & \\ score(S_N, S_1) & score(S_N, S_2) & ... & score(S_N, S_N) & score(S_N, Q) \\ score(Q, S_1) & score(Q, S_2) & ... & score(Q, S_N) & score(Q, Q) \end{bmatrix}$$

Matrix M is then normalized by dividing each row by its diagonal element:

$$M_{ij} = \frac{M_{ij}}{M_{ii}} \tag{3.15}$$

where M_{ij} is the matrix element at i^{th} row and j^{th} column. Here M_{ii} will be the diagonal element with respect to which the i^{th} row is normalized. The normalized matrix obtained is then converted to a symmetric matrix by using the following equation:

$$M = \frac{M + M^T}{2} \tag{3.16}$$

The matrix element scores obtained are then converted to dissimilarity score using the following equation:

$$dis\_score(M_{ij}) = \frac{2}{1 + M_{ij}} - 1 \tag{3.17}$$

We use a non-linear dissimilarity equation to convert the similarity matrix of normalized BM25 scores. In matrix M, all the documents are similar to the query based on one or more terms. It might be the case that relevant document(s) have fewer query matching terms and have a low similarity score. Consider a document with a similarity score of 0.5. The linear formula $1 - M_{ij}$ would give us a 0.5 dissimilarity score. In contrast, using the Eq. 3.17, we get a dissimilarity score of 0.333. A matrix of dissimilarity scores are then generated using Eq. 3.17.

$$M_f = \begin{bmatrix} dis\_score(M_{11}) & dis\_score(M_{12}) & ... & dis\_score(M_{1N}) & dis\_score(M_{1Q}) \\ dis\_score(M_{21}) & dis\_score(M_{22}) & ... & dis\_score(M_{2N}) & dis\_score(M_{2Q}) \\ ... & ... & ... & ... & \\ dis\_score(M_{N1}) & dis\_score(M_{N2}) & ... & dis\_score(M_{NN}) & dis\_score(M_{NQ}) \\ dis\_score(M_{Q1}) & dis\_score(M_{Q2}) & ... & dis\_score(M_{QN}) & dis\_score(M_{QQ}) \end{bmatrix}$$

Here, the rows of the final matrix M_f are the truncated documents of length less than or equal to the average document length. The modified documents represented as the rows of dissimilarity matrix are then converted to 2-dimensional space using t-SNE. The obtained 2-dimensional matrix is visualized as a document-query space which consists of the query Q and the modified documents from S.

The document-query visualization space would be rendered based on the proposed visualization approach for the given user inputs the query. For ranking of the documents, Euclidean distance is computed between the t-SNE projection of query with rest of the documents in S and sorted in the order that has minimum Euclidean distance from the query.

The user can click on the document and view the content of the document in the document view. Based on the preference and content, the user can give up-vote (positive feedback) or down-vote (negative feedback) to a document. The hybrid query expansion model suggests expansion terms based on the submitted user feedback document(s) and word-embedding. The suggested expansion terms are selected by default; users may deselect terms from the initial expansion terms selected.

Based on the expansion terms selected, the similarity matrix M is updated as weights of the expansion terms are included in the similarity calculation shown in Algorithm 2. Additionally, the dissimilarity matrix weights in M_f are updated based on the feedback document(s). The dissimilarity matrix element for positive documents and query are updated to 0 whereas, negative feedback documents and the query are modified to maximum value from M_f.

Algorithm 2 Generating Matrix M

```
Input: User Query Q and Document collection S.
Output: Generating document vectors for matrix M
  rep_collection = $[S, Q]$
  $t_Q$: terms in query
  for every $d\ in\ rep\_collection$ do
    $t\_d$ = getTermsFromDocumet(d)
    {Here, function getTermsFromDocument fetches the top-k TF-IDF terms from the input document. Here top-k is the average document length of the collection.}
    $term\_prob\_map$ : obtained after user filtering terms from final_map obtained from algorithm 1
    $doc\_query = ""$
    {$doc\_query$ is the final weighted query which includes the expansion terms as well if selected. If expansion terms are selected, $term\_prob\_map$ will be true.}
    for every $t\ in\ t\_d$ do
      if $term\_prob\_map$ then
        if $w_t\ in\ term\_prob\_map.keys()$ then
          $doc\_query = doc\_query + "\ " + w_t * (term\_prob\_map.get(t) + tfidf(t, d))$
        end if
      else if $t\ in\ t_Q$ then
        $doc\_query = doc\_query + "\ " + w_t * (count(t, Q) + tfidf(t, d))$
      else
        $doc\_query = doc\_query + "\ " + w_t * (tfidf(t, d))$
      end if
    end for
    $document\_vector\_d = score(doc\_query, rep\_collection)$
    {score function computes the BM25 score with the weighted query formed and assigned to variable $doc\_query$.}
  end for
```

Chapter 4

Experiments

4.1 Experimental Setup

4.1.1 Dataset

To evaluate our system, we use the CACM collection, a collection of titles and abstracts from the journal CACM [12].

Collection	No. of Documents	Queries	Avg. Document Length
CACM	3204	64	36

Table 4.1: Statistics of collection used for experiments evaluations

We extract the title, author's, keyword, and abstract from the dataset. It is observed that one document does not contain a title, and out of 3204 documents, only 1587 documents had abstract content. Also, out of 64 queries given, only 52 queries have relevance judgment documents associated.

4.1.2 Development Environment

Our system is developed using HTML, CSS, javascript and D3 [5] for the front-end and Python[1]. for the back-end. The PyLucene Toolkit[2] is used to index the document collection and calculate similarity function BM25.

4.1.3 Text pre-processing

The content of the document is pre-processed by removing the stop words from the given list of 429 common words with the CACM dataset and the string punctuation symbols. We use the Snowball stemming algorithm to remove the more common morphological and inflexional endings from words.

[1] `https://www.python.org/`

[2] `https://lucene.apache.org/pylucene/`.

4.1.4 Word Embedding

Training word-embedding on domain-specific data helps to suggest the query expansion terms that are more frequent in relevant documents [10]. However, CACM collection was insufficient to generate the word2vec model that captures semantic relationships using different parameter settings. Hence we used the vocabulary of the CACM collection except for the common words and fetched the vectors from the pre-trained word2vec model trained on Google News documents[3].

4.1.5 Baseline and Performance Metrics

We consider the standard probabilistic model used for retrieving a ranked list of documents, BM25 (Eq. 3.1), as our baseline. The search results are evaluated over the top-20 ranked list of documents for the given user query. We use precision, recall, mean average precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) [17] as performance metrics for the system.

4.1.6 Evaluator for Query Expansion Model

For evaluating the performance of our query expansion model, we stimulate user feedback. The top 10 documents are selected from the initially retrieved list to select the feedback documents. According to the gold standard, the document is voted as a positive document if it is relevant to the query or as a negative document if it is not. Here the gold standard are the true labels of whether the document is relevant or not for the given query provided with the dataset. Top 10 expansion terms apart from query terms are selected from the suggested list with the initial query to get the second iteration performance metrics.

Performance of different query expansion models based on the stimulated number of relevance feedback is shown in Fig. 4.1. The MAP@20 and NDCG@20 scores are averaged over three runs, each with a different set of feedback documents provided in each run. Here, on the x-axis, the number of feedback documents includes the same set of positive and negative feedback documents for each model from the top-10 documents in the initially retrieved list of documents.

[3] `https://code.google.com/archive/p/word2vec/`.

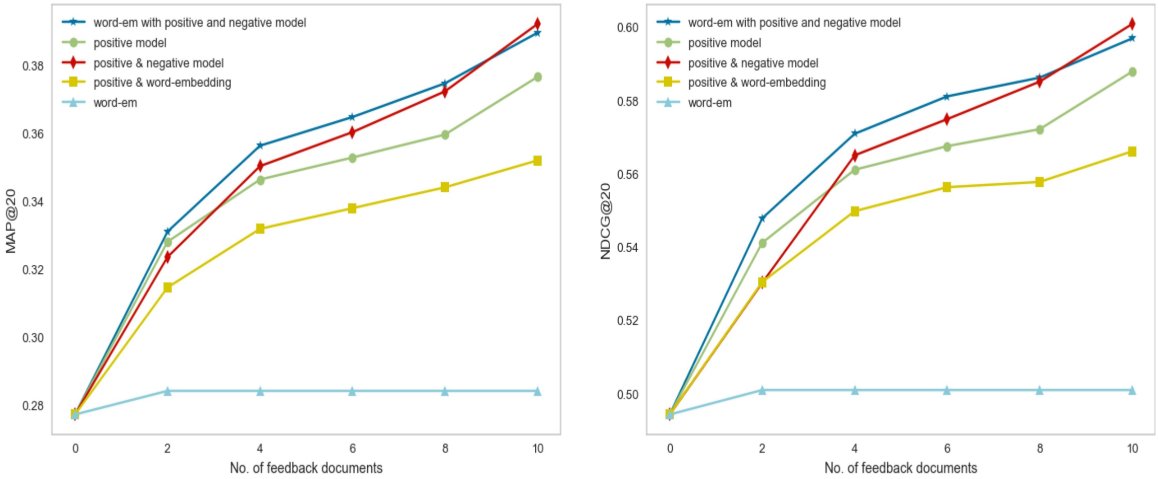


Figure 4.1: MAP and NDCG score averaged over 3 runs for different number of feedback documents given for Query Expansion Models.

It is observed that integrating the word-embedding model with the positive and negative feedback model increases the models' performance over the other models. The proposed model performs better in terms of MAP@20 and NDCG@20 scores than the other query expansion models. However, with more feedback documents (eight to ten), the combined positive and negative models perform better.

4.1.7 Parameter settings

Parameters for BM25 Algorithm in Eq. 3.1 are tuned to $k1 = 1.2$ and $b = 0.75$, empirically for CACM queries. Parameters for query expansion model $\lambda, \gamma_N, \gamma_P, \gamma_C, \beta_P, \beta_W, \beta_N$ are set as 0.5, 0.5, 0.2, 0.3, 0.5, 0.3, 0.2 respectively. The query expansion model parameters are set empirically based on the MAP and NDCG scores using the evaluator to stimulate relevance feedback. For converting document representation to 2-D, TSNE perplexity is set to 30 empirically for the CACM document collection.

4.2 Application Use Cases

Consider a scenario where a user wants to find all the relevant documents from the domain-specific collection data. For example, a doctor might want to go through all the case history regarding a particular disease or a lawyer searching for previous cases and evidence from a set of legal documents. In this thesis, we consider the

CACM dataset from which a student wants to study relevant papers from computer science regarding a particular topic of interest. This system helps the user to analyze documents visualized in a document-query space compare to the standard rank list. Users can also filter the documents based on the group of query terms occurrences. Moreover, the user can also provide positive or negative feedback to a document based on the document's content. As a result of the user's input, the system suggests the expansion terms using a hybrid expansion model. The user can eliminate terms that do not match the context of the query from the suggested list of terms. The visual document-query space would adapt based on the suggested expansion terms and the user feedback documents on re-search.

4.3 Use case Scenarios

Consider a scenario where a user wants to explore the collection of documents and find relevant documents for a given query. Here, we consider the CACM dataset as our collection of computer science papers with title, author, and abstract.

4.3.1 Scenario 1

Initial retrieval result using the proposed visual interface for the given user query: "code optimization for space efficiency". As per the gold standard, there are 11 relevant documents for the given user query.

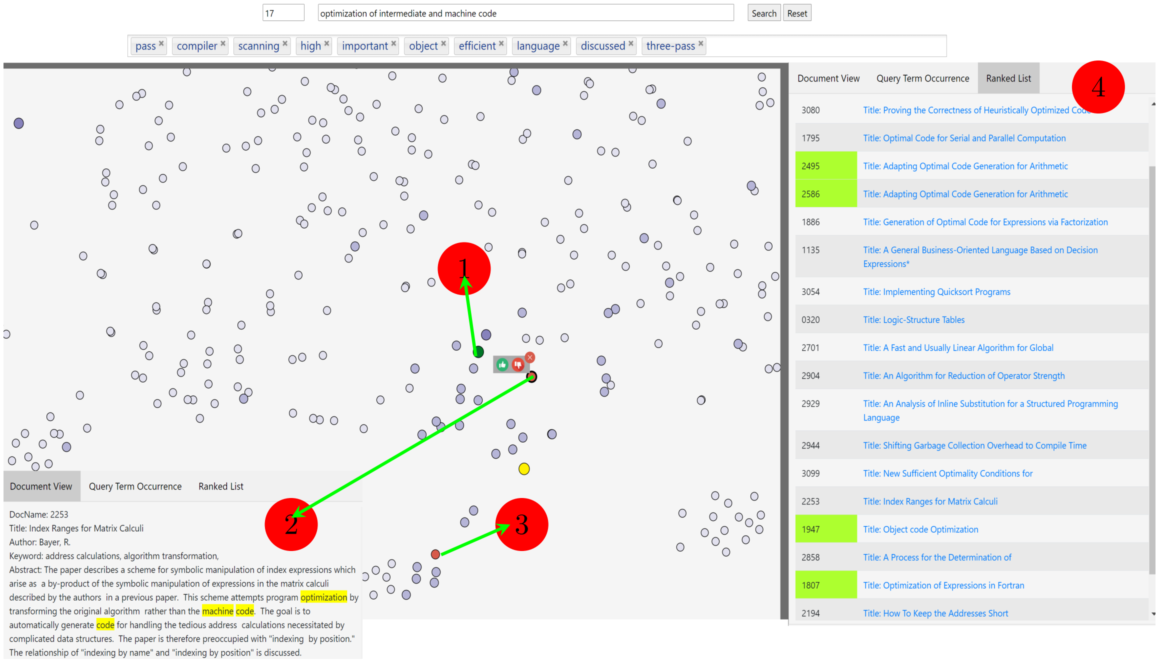
User inputs the given query to find the relevant documents using the interface proposed. The performance metrics of the baseline BM25 and the proposed document-query space (QE-VizIR) are shown in Table 4.2.

	Precision@20	Recall@20	MAP@20	NDCG@20
BM25	0.30	0.5454	0.2590	0.4796
QE-VizIR	0.35	0.6363	0.3784	0.5914

Table 4.2: Performance Measure of the baseline model BM25 and the interface proposed in this thesis on initial retrieval (VizIR) for the User Query in scenario 1.

4.3.2 Scenario 2

A user interacting with the interface gives positive and negative feedback and re-search after selecting the desired suggested query expansion terms to improve the performance of the system.



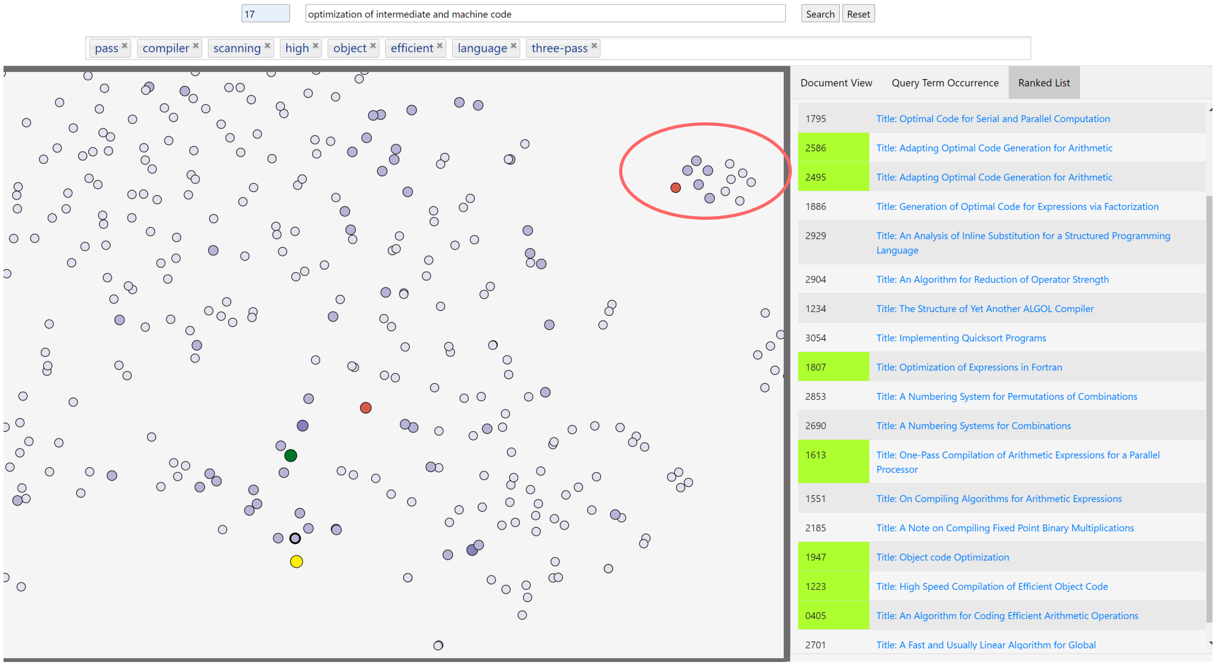
DocName: 2253
Title: Index Ranges for Matrix Calculi
Author: Bayer, R.
Keyword: address calculations, algorithm transformation,
Abstract: The paper describes a scheme for symbolic manipulation of index expressions which arise as a by-product of the symbolic manipulation of expressions in the matrix calculi described by the authors in a previous paper. This scheme attempts program optimization by transforming the original algorithm rather than the machine code. The goal is to automatically generate code for handling the tedious address calculations necessitated by complicated data structures. The paper is therefore preoccupied with "indexing by position." The relationship of "indexing by name" and "indexing by position" is discussed.

3080	Title: Proving the Correctness of Heuristically Optimized Code
1795	Title: Optimal Code for Serial and Parallel Computation
2495	Title: Adapting Optimal Code Generation for Arithmetic
2586	Title: Adapting Optimal Code Generation for Arithmetic
1886	Title: Generation of Optimal Code for Expressions via Factorization
1135	Title: A General Business-Oriented Language Based on Decision Expressions*
3054	Title: Implementing Quicksort Programs
0320	Title: Logic-Structure Tables
2701	Title: A Fast and Usually Linear Algorithm for Global
2904	Title: An Algorithm for Reduction of Operator Strength
2929	Title: An Analysis of Inline Substitution for a Structured Programming Language
2944	Title: Shifting Garbage Collection Overhead to Compile Time
3099	Title: New Sufficient Optimality Conditions for
2253	Title: Index Ranges for Matrix Calculi
1947	Title: Object code Optimization
2858	Title: A Process for the Determination of
1807	Title: Optimization of Expressions in Fortran
2194	Title: How To Keep the Addresses Short

Figure 4.2: Initial search results for the given user query. Top-10 expansion terms are suggested based on the user query and feedback.

Here, we consider the user query: "optimization of intermediate and machine code". According to the gold standard, there are 16 relevant documents for the given query.

Exploration and User Feedback: After obtaining the initial result for the given query, the user can then explore the document-query space as shown in Fig. 4.2. Here, the user can analyze the result based on the preferences by evaluating the document based on the term intensity color of the points if they have maximum query terms. Besides, users can filter the documents based on the query term occurrence to find the documents containing a certain group of specific query terms or evaluate the documents near the query point. The green point represents the positive feedback given, as it has high intensity based on query terms and is relevant to the query (1).



17

optimization of intermediate and machine code

1795	Title: Optimal Code for Serial and Parallel Computation
2586	Title: Adapting Optimal Code Generation for Arithmetic
2495	Title: Adapting Optimal Code Generation for Arithmetic
1886	Title: Generation of Optimal Code for Expressions via Factorization
2929	Title: An Analysis of Inline Substitution for a Structured Programming Language
2904	Title: An Algorithm for Reduction of Operator Strength
1234	Title: The Structure of Yet Another ALGOL Compiler
3054	Title: Implementing Quicksort Programs
1807	Title: Optimization of Expressions in Fortran
2853	Title: A Numbering System for Permutations of Combinations
2690	Title: A Numbering Systems for Combinations
1613	Title: One-Pass Compilation of Arithmetic Expressions for a Parallel Processor
1551	Title: On Compiling Algorithms for Arithmetic Expressions
2185	Title: A Note on Compiling Fixed Point Binary Multiplications
1947	Title: Object code Optimization
1223	Title: High Speed Compilation of Efficient Object Code
0405	Title: An Algorithm for Coding Efficient Arithmetic Operations
2701	Title: A Fast and Usually Linear Algorithm for Global

Figure 4.3: Second iteration, after submitting the select list of query expansion terms the document-query space re-renders.

Document selected with the document view (2) in the figure based on the query term occurrence of terms ‘optimization’, ‘machine’ and ‘code’. It has important terms, but the document’s context is not relevant, so it has been given a down-vote. One more negative feedback document was selected from the cluster near the query (3). The user can also see the ranked list generated based on the top 20 documents near to query point (4). (Note: light green color in the ranked list indicates the relevant document for the given query as per gold standard). Finally, one positive feedback and two negative feedback documents are submitted to query. Based on that, the system suggests a list of expansion terms.

Query Expansion and Result Analysis: After exploration, giving feedback(s) and obtaining the query expansion list. The user can remove the expansion term(s) if the terms are more general or might not add the correct information to the query. In this case, we select all the expansion terms suggested by the system. Document-query space adapts and re-render based on the feedback(s) and expanded user query in the second iteration as shown in Fig. 4.3. Now, in the ranked list, more relevant documents are retrieved compare to initial retrieval. The positive feedback document has now been included in the top-20 ranked list. Nevertheless, relevant documents did contain the terms from the suggested expansion list and attracted the query.

Additionally, one of the negative feedback documents surrounded by a cluster of documents near the query point in initial retrieval is moved far away with all the cluster documents around itself. The documents in the cluster highlighted in Fig. 4.3 were not relevant to the user query. They were of the same topic focusing on the optimization of code for decision tables.

The performance of the BM25 and visual interface proposed in initial retrieval and after query expansion is shown in Table 4.3. Here, the visual interface was able to retrieve the same number of relevant documents as BM25 (baseline), but NDCG and MAP were better in the case of BM25. After providing feedback documents and adding expansion terms in the second iteration, it was observed that the system was able to improve the retrieved results.

	Precision@20	Recall@20	MAP@20	NDCG@20
BM25	0.20	0.25	0.1241	0.3114
QE-VizIR Iter-0	0.20	0.25	0.0662	0.2312
QE-VizIR Iter-1	0.40	0.50	0.1619	0.4114

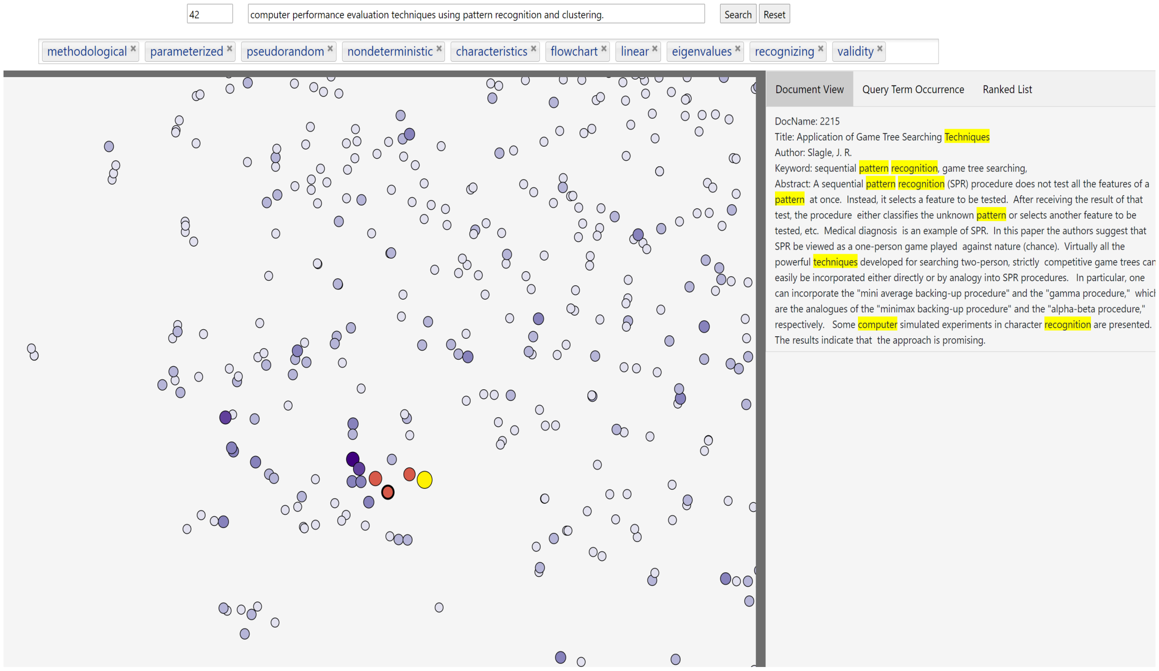
Table 4.3: Performance Measure of the baseline model BM25 and the interface proposed in this thesis on initial retrieval (QE-VizIR Iter-0) and after expansion term selection and user feedback provided in second iteration (QE-VizIR Iter-1) for the User Query in scenario 2.

4.3.3 Scenario 3

A user interacting with the interface cannot find any positive documents after analyzing few documents. So, the user provides only negative feedback documents and re-search after selecting the desired suggested query expansion terms to improve the system's performance.

Here, we consider the user query: "computer performance evaluation techniques using pattern recognition and clustering". According to the gold standard, there are 21 relevant documents for the given query.

Exploration and User Feedback: On the initial result, the user explores the document-query space as shown in Fig. 4.4. In the initial retrieved list, there is no



42

computer performance evaluation techniques using pattern recognition and clustering.

methodological × parameterized × pseudorandom × nondeterministic × characteristics × flowchart × linear × eigenvalues × recognizing × validity ×

DocName: 2215
Title: Application of Game Tree Searching Techniques
Author: Slagle, J. R.
Keyword: sequential pattern recognition, game tree searching,
Abstract: A sequential pattern recognition (SPR) procedure does not test all the features of a pattern at once. Instead, it selects a feature to be tested. After receiving the result of that test, the procedure either classifies the unknown pattern or selects another feature to be tested, etc. Medical diagnosis is an example of SPR. In this paper the authors suggest that SPR be viewed as a one-person game played against nature (chance). Virtually all the powerful techniques developed for searching two-person, strictly competitive game trees can easily be incorporated either directly or by analogy into SPR procedures. In particular, one can incorporate the "mini average backing-up procedure" and the "gamma procedure," which are the analogues of the "minimax backing-up procedure" and the "alpha-beta procedure," respectively. Some computer simulated experiments in character recognition are presented. The results indicate that the approach is promising.

Figure 4.4: Initial search results for the given user query. Top-10 expansion terms are suggested based on the user query and feedback.

relevant document in the top-20 documents. Considering the user decides to evaluate the documents near the query, after assessing 3-4 documents, the user cannot find any relevant document for the query. So the user submits only negative feedback documents to get the expansion term suggestion. In this case, terms from the query are penalized based on the negative feedback model, and expansion terms are suggested based on the word-embedding model.

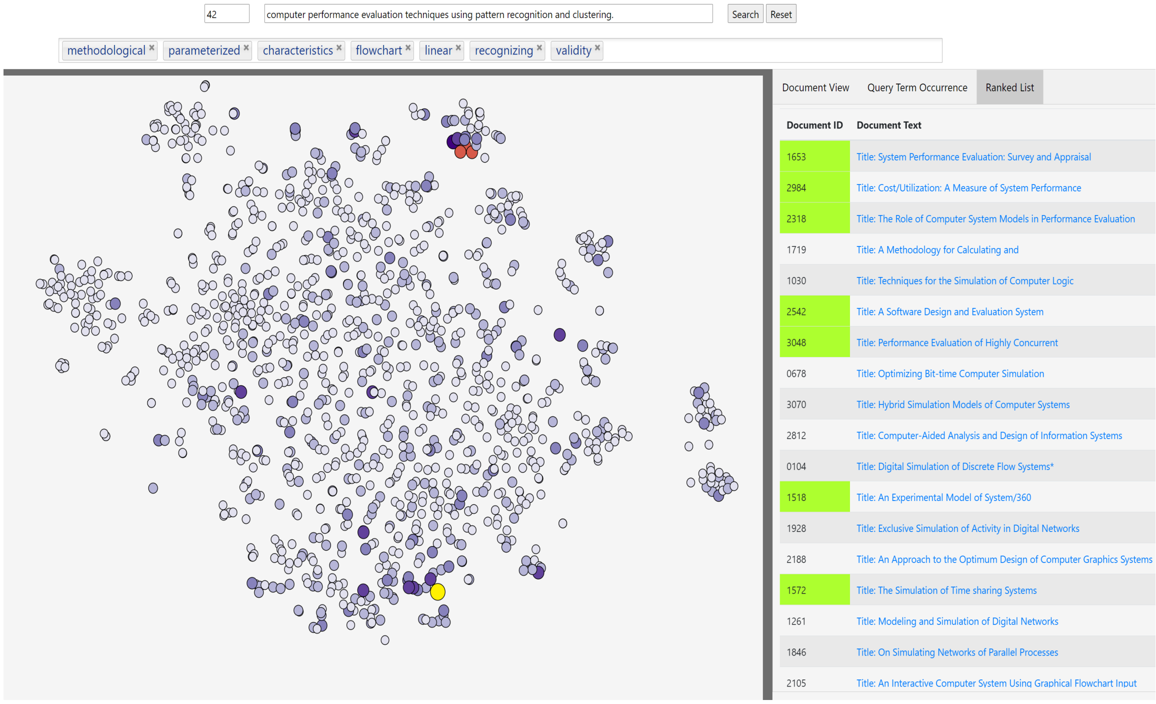
Query Expansion and Result Analysis: The list of 10 suggested expansion terms that are selected by default can be seen in Fig. 4.4. The expansion terms selected by the user from the suggested list, and the updated document-query space, can be seen in Fig. 4.5. In the new ranked list, the relevant documents that are fetched only contain two terms from the selected expansion terms:“methodology” and “characteristics”.

The performance of the BM25 and visual interface proposed in initial retrieval and after query expansion is shown in Table 4.4. Here, the visual interface (QE-VizIR Iter-0) was not able to retrieve any relevant documents, whereas BM25 (baseline) was able to retrieve 3 out of 21 relevant documents. After providing feedback documents

	Precision@20	Recall@20	MAP@20	NDCG@20
BM25	0.15	0.1428	0.0292	0.1425
QE-VizIR Iter-0	0.00	0.00	0.00	0.00
QE-VizIR Iter-1	0.35	0.3333	0.2546	0.5264

Table 4.4: Performance Measure of the baseline model BM25 and the interface proposed in this thesis on initial retrieval (QE-VizIR Iter-0) and after expansion term selection and user feedback provided in second iteration (QE-VizIR Iter-1) for the User Query in scenario 3.

and adding expansion terms in the second iteration, it was observed that the system was able to improve the retrieved results.



Document ID	Document Text
1653	Title: System Performance Evaluation: Survey and Appraisal
2984	Title: Cost/Utilization: A Measure of System Performance
2318	Title: The Role of Computer System Models in Performance Evaluation
1719	Title: A Methodology for Calculating and
1030	Title: Techniques for the Simulation of Computer Logic
2542	Title: A Software Design and Evaluation System
3048	Title: Performance Evaluation of Highly Concurrent
0678	Title: Optimizing Bit-time Computer Simulation
3070	Title: Hybrid Simulation Models of Computer Systems
2812	Title: Computer-Aided Analysis and Design of Information Systems
0104	Title: Digital Simulation of Discrete Flow Systems*
1518	Title: An Experimental Model of System/360
1928	Title: Exclusive Simulation of Activity in Digital Networks
2188	Title: An Approach to the Optimum Design of Computer Graphics Systems
1572	Title: The Simulation of Time sharing Systems
1261	Title: Modeling and Simulation of Digital Networks
1846	Title: On Simulating Networks of Parallel Processes
2105	Title: An Interactive Computer System Using Graphical Flowchart Input

Figure 4.5: Initial search results for the given user query. Top-10 expansion terms are suggested based on the user query and feedback.

4.4 Discussion

The different query expansion models' performance based on the user relevance feedback stimulated using the evaluator is shown in Fig. 4.1. It was observed that the query expansion model using only word-embeddings could improve the map by only 0.63% as even though the word-embeddings provided semantically similar terms.

However, some terms were not in the context. For example, for the query: What articles exist which deal with TSS (Time Sharing System), an operating system for IBM computers? The terms suggested were January, December, April, agreement, pascal. These terms are suggested due to the time and deal vector in the query. In contrast, when integrated with the positive and negative feedback model with a small weight ratio, it enhances the model's performance at using fewer relevance feedback documents. Moreover, we only penalize the terms present in the query, positive documents, and word embedding for the negative feedback model. Including the unseen terms generated by the negative feedback, the model might lower the rank of non-relevant documents.

For the visual interface proposed in this thesis, based on the sample queries, it is observed that the hypothesis of representing the documents in terms of other documents and query does place the documents having the most similar terms together. Also, it is observed that changing the representation based on the user feedback does impact the document query. As in user scenario 2, one document in a cluster was given negative feedback, and on the re-search, the entire cluster is moved away from the query. The interface does provide easy navigation, analysis, and feedback mechanism using the document-query visualization approach.

Chapter 5

Conclusion

5.1 Limitations

The positive and negative feedback models used have the same limitation of finding global optima as initial probabilities for E-step are chosen based on TF-IDF values of the term and negative feedback depends on the positive model probabilities when estimating the EM model [14]. Moreover, the word-embedding used are the vectors from the pre-trained model and has some corpus terms missing; training the word embedding over the larger set of computer science papers abstract dataset might boost the word-embedding model's performance.

To render the documents in 2-D document-query space, we use t-SNE. It was observed that fixed perplexity might not generalize to different queries or expansion terms since changes to perplexity can improve the results for few sample queries. In some cases, the relevant document to the query containing the most matched terms might be pulled away by the other documents having more relevance scores with respect to that document. As a result, sometimes, the visual interface's initial rank list might not be as effective as BM25, as seen in use case scenario 3. However, it can be improved by using the hybrid query expansion algorithm and user feedback.

Moreover, the weights generated by the hybrid query expansion model are internal. The user has no knowledge or control over the modified query weights and the expansion term weights used to create the document-query space representation.

The computation of the similarity matrix of relevance score is computationally complex compared to the standard rank list due to the computation of the BM25 relevance score for the documents with respect to the other documents.

5.2 Conclusion and Future Work

In this thesis, we proposed a hybrid query expansion assisted adaptive visual interface. The hybrid model incorporates the positive and negative feedback model with the word embedding model to provide the suggestion terms. The experiment result shows that integrating the word-embedding model with the positive and negative feedback model can boost the average precision and NDCG with fewer feedback documents.

The system allows users to interact with the visual interface in terms of analyzing the documents in the document-query space and provide relevance feedback. The hybrid query expansion model suggests the expansion terms to the user. Based on the selected expansion terms and user feedback, the document-query space is updated. User Scenarios shows that the adaptive document-query space is able to improve the performance of the system. However, to evaluate the system performance, a formal user study shall be conducted in the future.

The negative feedback model only penalizes the terms from the query, positive feedback and word embedding model. In the future, it would be interesting to provide a visual space for query expansion terms using a graph to illustrate the query, feedback documents, the terms suggested, and how they are linked to the feedback documents and word-embedding. Also, to evaluate the performance by penalizing the unseen terms generated by the negative feedback model. Furthermore, current query expansion model weights and parameters are fixed for the given dataset. Hence, as future work, the weights should be learned with reference to the number and type of feedback documents.

Bibliography

[1] Shariq Bashir. Improving retrievability with improved cluster-based pseudo-relevance feedback selection. *Expert Systems with Applications*, 39(8):7495 – 7502, 2012.

[2] M. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, 1989.

[3] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information Processing and Management*, 43(4):866–886, 2007.

[4] D. Blocks, C. Binding, D. Cunliffe, and D. Tudhope. Qualitative evaluation of thesaurus-based retrieval. In Maristella Agosti and Costantino Thanos, editors, *Research and Advanced Technology for Digital Libraries*, pages 346–361, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[5] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[6] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 243–250, New York, NY, USA, 2008. Association for Computing Machinery.

[7] Kevyn Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 837–846, New York, NY, USA, 2009. Association for Computing Machinery.

[8] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, page 325–332, New York, NY, USA, 2002. Association for Computing Machinery.

[9] Amanda Gonçalves Dias, Evangelos E. Milios, and Maria Cristina Ferreira de Oliveira. TRIVIR: A visualization system to support document retrieval with high recall. In *Proceedings of the ACM Symposium on Document Engineering 2019*, DocEng '19, New York, NY, USA, 2019. Association for Computing Machinery.

[10] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 367–377, Berlin, Germany, August 2016. Association for Computational Linguistics.

[11] Efthimis N. Efthimiadis. Interactive query expansion: A user-based evaluation in a relevance feedback environment. *Journal of the American Society for Information Science*, 51(11):989–1003, 2000.

[12] Edward A Fox. Characterization of two new experimental collections in computer and information science containing textual and bibliographic concepts. Technical report, Cornell University, 1983.

[13] M. Granitzer, W. Kienreich, V. Sabol, and G. Dosinger. Webrat: supporting agile knowledge retrieval through dynamic, incremental clustering and automatic labelling of web search result sets. In *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, pages 296–301, 2003.

[14] Shufeng Hao, Chongyang Shi, Zhendong Niu, and Longbing Cao. Modeling positive and negative feedback for improving document retrieval. *Expert Systems with Applications*, 120:253–261, 2019.

[15] Marti A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 59–66, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[16] Jae-wook Ahn and Peter Brusilovsky. Adaptive visualization for exploratory information retrieval. *Information Processing & Management*, 49(5):1139–1164, 2013.

[17] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[18] Khalil Klouche, Tuukka Ruotsalo, Luana Micallef, Salvatore Andolina, and Giulio Jacucci. Visual re-ranking for multi-aspect information retrieval. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, page 57–66, New York, NY, USA, 2017. Association for Computing Machinery.

[19] Ari Korhonen, Juha Litola, and Jorma Tarhio. Platform for elaboration of search results. In *Web Information Systems and Technologies, March 3-6, Barcelona, Spain*, pages 263–269, 01 2007.

[20] Robert Krovetz and W. Bruce Croft. Lexical ambiguity and information retrieval. *ACM Trans. Inf. Syst.*, 10(2):115–141, April 1992.

[21] Saar Kuzi, Anna Shtok, and Oren Kurland. Query expansion using word embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1929–1932, New York, NY, USA, 2016. ACM.

[22] 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, page 111–119, New York, NY, USA, 2001. Association for Computing Machinery.

[23] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 120–127, New York, NY, USA, 2001. Association for Computing Machinery.

[24] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org, 2014.

[25] Gary Marchionini. Exploratory search: From finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.

[26] Kai A. Olsen, Robert R. Korfhage, Kenneth M. Sochats, Michael B. Spring, and James G. Williams. Visualization of a document collection: The vibe system. *Information Processing & Management*, 29(1):69–81, 1993.

[27] Dipasree Pal, Mandar Mitra, and Kalyankumar Datta. Improving query expansion using wordnet. *Journal of the Association for Information Science and Technology*, 65(12):2469–2478, December 2014.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[29] Stephen Robertson, S. Walker, and M.M. Beaulieu. Okapi at TREC-7: automatic ad hoc, filtering, VCL and interactive track. *In*, January 1999.

[30] J. J. Rocchio. *Relevance Feedback in Information Retrieval.* Prentice Hall, Englewood, Cliffs, New Jersey, 1971.

[31] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. Interactive intent modeling: Information discovery beyond search. *Commun. ACM*, 58(1):86–92, December 2014.

[32] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[33] Amanda Spink, Dietmar Wolfram, Major B. J. Jansen, and Tefko Saracevic. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3):226–234, 2001.

[34] Zhiwen Tang and Grace Hui Yang. DeepTileBars: Visualizing term distribution for neural information retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):289–296, Jul. 2019.

[35] Egidio Terra and Robert Warren. Poison pills: Harmful relevant documents in feedback. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, page 319–320, New York, NY, USA, 2005. Association for Computing Machinery.

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[37] Ellen M. Voorhees. Query expansion using lexical-semantic relations. In Bruce W. Croft and C. J. van Rijsbergen, editors, *SIGIR '94*, pages 61–69, London, 1994. Springer London.

[38] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. Improve retrieval accuracy for difficult queries using negative feedback. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 991–994, New York, NY, USA, 2007. ACM.

[39] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, page 4–11, New York, NY, USA, 1996. Association for Computing Machinery.

[40] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, page 4–11, New York, NY, USA, 1996. Association for Computing Machinery.

[41] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of Tenth International Conference on Information and Knowledge Management*, pages 403–410, 2001.

[42] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 403–410, New York, NY, USA, 2001. Association for Computing Machinery.

[43] Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. Effective user interaction for high-recall retrieval: Less is more. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 187–196, New York, NY, USA, 2018. Association for Computing Machinery.