

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Shishir Bhattarai

Interactive User Intent Modeling: Usefulness of Session-level Relevance Feedback

Master's Thesis
Espoo, September 24, 2016

Supervisor: Professor Samuel Kaski
Advisor: M.Sc. (Tech.) Antti Kangasrääsio

Author:	Shishir Bhattarai	
Title:	Interactive User Intent Modeling: Usefulness of Session-level Relevance Feedback	
Date:	September 24, 2016	Pages: 44
Major:	Machine Learning and Data Mining	Code: CS
Supervisor:	Professor Samuel Kaski	
Advisor:	M.Sc. (Tech.) Antti Kangasrääsiö	
	<p>In information retrieval systems, users often have difficulties in forming precise queries to express their information need. One approach to express information need is to explore the information space by providing relevance feedback to recommended items. This feedback is then used to model user search intent. Studies have shown how retrieval performance could be improved by allowing users to give feedback to multiple items such as keywords and documents instead of keywords only. In this thesis, I extend an existing user model which uses document-level and keyword-level feedback to include session-level feedback, and study the usefulness of this extension. By conducting simulation studies in various settings, I investigate the effect of session-level feedback. Based on these simulation results, I conclude that additional session-feedback helps in finding relevant documents by improving F1-score. Results show that more the additional session-feedback, more the improvement in F1-score. However, trade-off of session-feedback instead of document and keyword feedback results in drop in document F1-score, therefore indicating that session-feedback is less informative than document and keyword feedback.</p>	
Keywords:	information re-finding, information retrieval, interactive user intent modeling, multi-armed bandit problem	
Language:	English	

Acknowledgements

My thesis has been carried out being part of MindSee project in Probabilistic Machine Learning Group, Aalto University, Finland. First I would like to thank my supervisor Prof. Samuel Kaski for his valuable time and effort to give constructive comments on my thesis writing and research. I would like to thank my adviser M.Sc.(Tech.) Antti Kangasrääsiö for his never ending ideas and support throughout the thesis. A share of thanks also goes to Pedram Daei, Andrej Gisbrecht and Yi Chen who helped me by providing ideas and suggestions during my research. A big share of thank goes to Probabilistic Machine Learning group who were always around me to boost my motivation and energy.

I would like to thank my friends Binda Pandey, Merlin Samy, Vivek Dhakal, Kunal Ghosh, Srikant Gardicherla and Yiping Liao for their comments and feedback in my thesis writing. Last but not the least my warm gratitude and thanks to my family, relatives and teachers for their guidance throughout my life.

Espoo, September 24, 2016

Shishir Bhattarai

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Motivation and Scope	2
1.3	Structure of Thesis	4
2	Information Retrieval	5
2.1	Introduction	5
2.2	Representation of Documents	7
2.3	Re-finding	8
2.4	Collaborative Search	9
2.5	Evaluation	10
3	Contextual Multi-Armed Bandit	
	Problem	12
3.1	Mathematical Formulation	12
3.2	Thompson Sampling	14
4	Interactive User Intent Modeling	15
4.1	Keywords Interaction System	15
4.2	Documents and Keywords Interaction System	17
4.3	Limitation of Existing Systems	19
5	Proposed Model	20
5.1	Model Assumption	20
5.2	Proposed Model as Contextual Multi-Armed Bandit Problem .	22
5.3	Probabilistic Model	23
5.4	Search Process of Proposed System	24
6	Simulation Environment	26
6.1	Dataset	26

6.2	Simulation of Ground-truth	28
6.3	Simulation of Document-Keyword Matrix and Past Search Session	28
6.4	Evaluation	29
7	Analysis of Results	30
7.1	Experiment 1: Session-Feedback in Addition to Document and Keyword Feedback	30
7.1.1	Setting	30
7.1.2	Results	30
7.2	Experiment 2: Session-Feedback Instead of Other Feedback	32
7.2.1	Setting	32
7.2.2	Results	33
7.3	Experiment 3: Additional Numbers of Session-Feedback	35
7.3.1	Setting	35
7.3.2	Results	35
8	Conclusion	39
8.1	Discussion	39
8.2	Limitation and Future Work	40

Chapter 1

Introduction

1.1 Introduction

The process of finding information that is relevant to the user's needs is called *information retrieval*. After the development of the World Wide Web (WWW) in 1990, there has been an exponential increase in the amount of information. For example, according to NetCraft, Web Server Survey October 2014 [29], there were over 1 billion websites in the WWW.

Finding and *re-finding* are two different information seeking behaviours in information retrieval [7]. In *finding*, users have partial information about their information need and are uncertain about existence, location and format of relevant information. To find relevant information, interactive approaches have been developed [12] [16] [33], where users start by forming an approximate query and refine it throughout the sessions according to their information need by providing relevance feedback. Dae et al. [12] demonstrated how retrieval performance could be improved by allowing users to give feedback to multiple types of items (keywords and documents) instead of just one (keywords).

Unlike finding, in *re-finding* users are certain that the information they are looking for exists because users have seen those information in their past search sessions. Studies and research shows re-finding information is a common activity in information retrieval. Teevan et al. [37] analysed web queries of 114 users from a one-year period and found that 40% of requested queries were re-finding queries.

To give an example of re-finding, let us assume that a user wants to collect information regarding geometry, biology and computer science, as shown in Figure 1.1. The user issues the queries as *geometry*, *biology* and *computer science* separately on the system and adjusts her queries throughout the sessions

by providing relevance feedback to the recommended items. Such feedback helps the system learn users need in each iteration. After one month, when the user needs information regarding squares, she remembers that she had found relevant information related to squares while searching for information related to geometry. However, it is difficult for her to recall the exact steps she needs to perform to get to the information she had found earlier. For a user like her, it is easier to recognise information regarding squares instead of recalling them. What if we could recommend relevant past search sessions to her based on the similarity between relevance of keywords and get feedback on them? If we can recommend relevant past search sessions similar to the user's current information need, it could help users recognise relevant past information, and this way, by interacting with it, she could be able to adjust her query.

What if we could use past search sessions of a number of users instead of just one? In the above example, there might be relevant past search session information related to squares by other users which can be useful in re-finding information she is looking for. The process of searching information using information generated by one or more users with common information need is known as *collaborative search* [28]. Studies have shown that users actively participate in collaborative search activities [27] [28].

1.2 Motivation and Scope

Re-finding the relevant past information is ubiquitous activity in information retrieval. It is easier for a human brain to recognise the relevant information that has been found before instead of recalling it from memory. One approach to help users re-find relevant information is to recommend past information based on the user's current search intent and collect feedback on recommended items to update the user model.

In this thesis, I study the usefulness of giving relevance feedback to recommended past search sessions. For this, I formulate a user model which recommends relevant past sessions along with keywords and documents and updates the user model based on relevance feedback. This thesis is motivated by the following facts:

1. Common Activity: Re-finding is a common activity in information retrieval.
2. Human Limitation: It is easier for users to recognise relevant items they have found before instead of memorising.

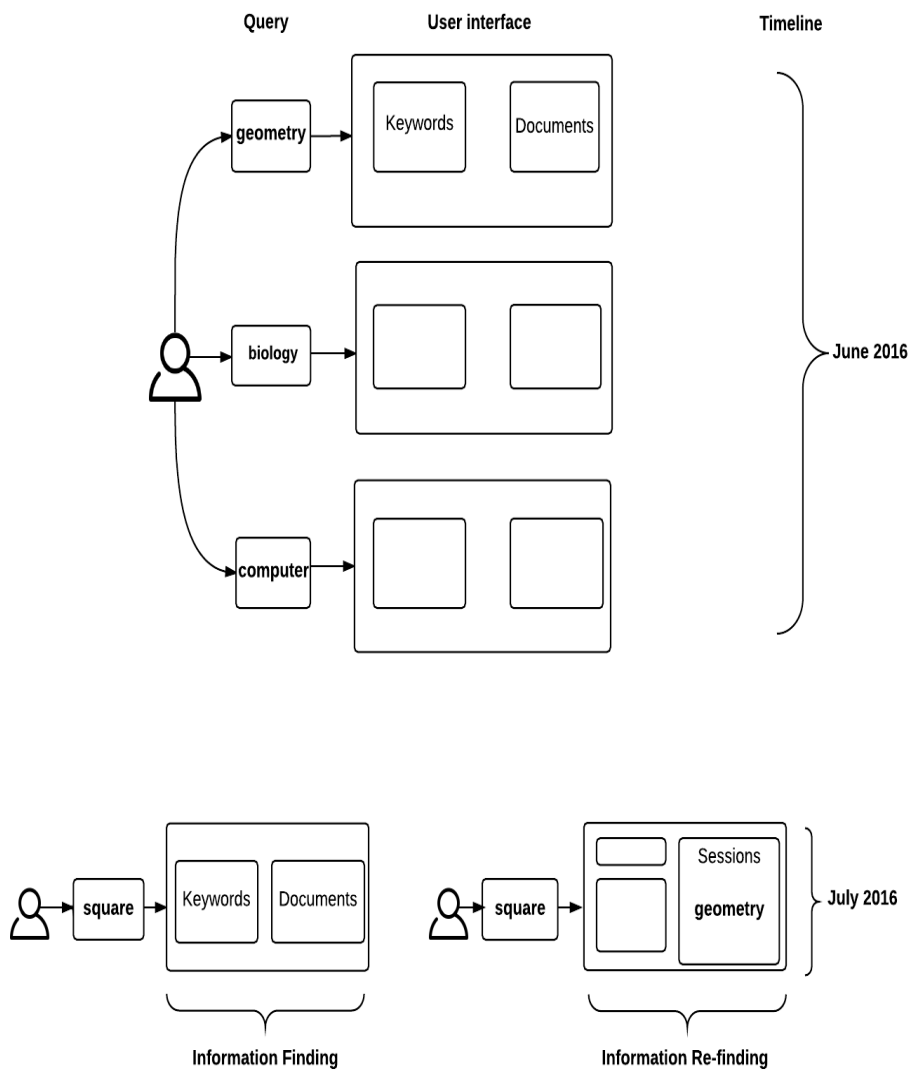


Figure 1.1: Comparison of two recommendation systems, one with information finding and other with information re-finding. Top: A user issued queries *geometry*, *biology* and *computer science* on June 2016. Bottom: A user needs information regarding square after one month and she remembers she had found information related squares before. Bottom Left: The system with information finding does not help her to re-locate previous relevant information she has found related to the square. Bottom Right: The system with information re-finding helps user to relocate previous relevant information.

3. Previous Work: Feedback from multiple items (documents and keywords) can improve the performance of finding relevant documents.

In this thesis re-finding is defined in collaborative setting. Instead of recommending past information of only one user, the new model recommends past search information of other users too. Performance of the model is evaluated by simulating interactive information system with different simulation settings. The performance of the new model is compared with the existing baseline model [12].

1.3 Structure of Thesis

Background information about information retrieval, contextual multi-armed bandit problem and previous works on interactive user intent modeling are described in Chapters 2-4. Chapter 2 introduces the necessary background related to information retrieval and information re-finding followed by contextual bandit problem, exploration and exploitation dilemma and Thompson sampling in Chapter 3. Chapter 4 introduces existing interactive user intent models.

Chapter 5 presents simplifying assumptions and formulation of the proposed model followed by interactive information retrieval simulation environment in Chapter 6. Chapter 7 presents results and analysis of the simulations. Chapter 8, discusses the summary of the thesis and further enhancements of the model.

Chapter 2

Information Retrieval

In this chapter, I give an overview of information retrieval, information re-finding, and information retrieval model evaluation methods. Section 2.1 gives a brief introduction to information retrieval and information retrieval processes followed by different techniques for document representation in Section 2.2. Section 2.3 discusses information re-finding and introduces some of the existing information re-finding tools. Section 2.4 introduces information retrieval collaborative processes and their classification based on user intent. Section 2.5 introduces information retrieval evaluation methods such as precision, recall and F1-score.

2.1 Introduction

Information Retrieval [26] [34] is *"the name of process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him"*. According to Elite and Rose [14], information retrieval started from the 3rd century when the Greek poet Callimachus created the world's first ever library catalogue known as *Pinakes*. However, only after the development of the WWW in 1990, there has been huge development in the field of information retrieval both in academic and commercial fields. Development of the WWW also made information retrieval popular among common people. A survey done by Pew Research Center found that 92% of internet users use search engines to find information on the web and 59% do so on a daily basis [31]. Before 1990, information retrieval systems were mostly used by professionals such as medical researchers, government organisation workers, librarians etc.

As defined by Croft [11], there are three basic processes in an information

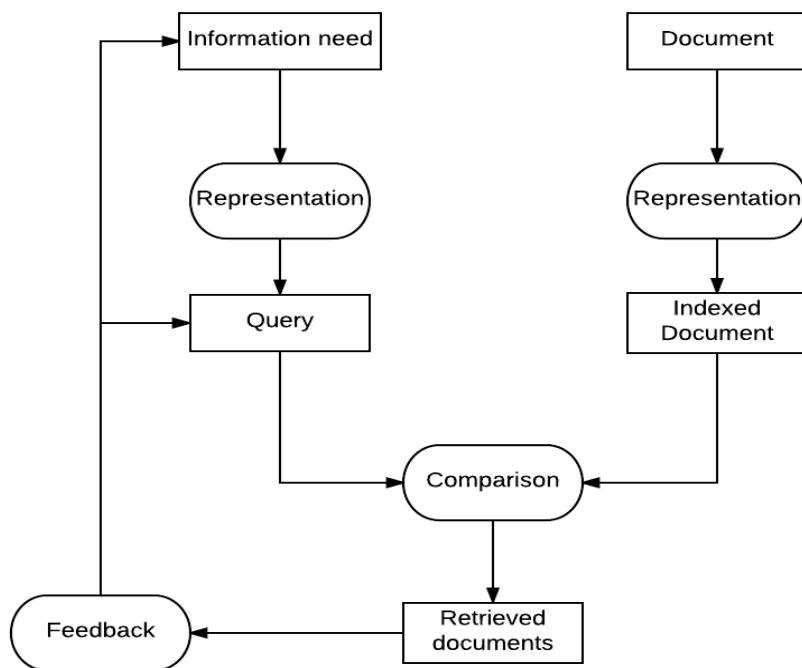


Figure 2.1: Information retrieval process. Squared boxes in figure represent data and rounded boxes represent processes. Directed arrows show flow of data.

retrieval system as shown in Figure 2.1:

- Representation of documents,
- Representation of user's information need,
- And a comparison of the two.

In order to compare user's information need and the collection of documents, each of these needs to be represented by its features. The process of representing documents by their features, like keywords, topics and so on is called the *indexing process*. An indexing process collects, cleans and normalises documents by stemming, removing stopwords and so on. Stemming is conflating words to their base form such as transforming 'playing', 'played', and 'player' to the root word 'play'. Removing stopwords is the process of eliminating very frequent terms such as 'the', 'of', 'a' and so on.

The process of representing a user's information need in the form of queries is known as a *query formulation process*. For example, in boolean

model, boolean representation of user's information need is query formulation process. The boolean vector from the process is the query.

The process of comparing a query against the indexed documents is known as the *comparison process*. The output from the comparison process is a list of documents. Depending on the system, comparison can be binary classification or ranking of documents. The output documents with relevant information are known as relevant documents.

In addition to the above processes, some information retrieval systems additionally have a *feedback process*. The process of representing responses provided by users to the recommended items is known as feedback process. Feedback is classified as:

1. Implicit Feedback, which is not explicitly given by the user but generated by the user behaviour such as click-through rate.
2. Explicit Feedback, which is explicitly given by the user such as relevance feedback to a system.

2.2 Representation of Documents

Documents are represented by their features such as terms (keywords), topics, and so on. In this thesis, documents are represented as bags of terms. The term is assigned a value which gives the importance of the term in the corresponding document. Some ways of calculating importance of terms are as follow:

- Term frequency (TF) is the frequency of a term in a document or in a collection of documents. Some terms like stopwords have high TF but these terms do not have discriminating power in information retrieval model.

Inverse document frequency (IDF) scales down the weight of terms with collection frequency to solve the limitation of term frequency. Collection frequency is the total number of occurrences of a term in a collection. Mathematically IDF is defined as

$$IDF_t = \log \frac{N}{DF_t}$$

where DF_t is collection frequency with term t and N is number of total documents.

- Term frequency - inverse document frequency (*TF-IDF*) mathematically combines term frequency and inverse document frequency as

$$TF\text{-}IDF_{t,d} = TF_{t,d} \times IDF_t$$

where $TF_{t,d}$ is term frequency of the term t in the document d and IDF_t is inverse document frequency of term t . *TF-IDF* is high when term t occurs many times within a small number of documents and low when the term occurs in many documents but fewer times in each documents.

- Conditional probabilities which give likelihood of document generating the terms. In document collection with $|D|$ documents and $|K|$ terms, document-term matrix $M_{|D| \times |K|}$ is

$$M_{|D| \times |K|} = \begin{pmatrix} P(k_1|d_1) & \cdots & P(k_{|K|}|d_1) \\ P(k_1|d_2) & \cdots & P(k_{|K|}|d_2) \\ \vdots & \ddots & \vdots \\ P(k_1|d_{|D|}) & \cdots & P(k_{|K|}|d_{|D|}) \end{pmatrix},$$

$$\sum_{i=1}^{|K|} P(k_i|d_j) = 1,$$

where $P(k_i|d_j)$ is likelihood of document d_j generating term k_i . $P(k_i|d_j)$ is probability of occurrence of term k_i in document d_j . The document-term matrix M is known as *language model*.

2.3 Re-finding

In information retrieval, finding and re-finding are two different information seeking behaviours [7]. In finding users have partial information about their information need and are uncertain about the existence, location and format. Finding relevant information for the first time is an exploratory search [7]. Exploratory search is an iterative information seeking behaviour where users information needs evolve throughout the session [24].

In re-finding users look for information related to what they have found before in their previous search session. In re-finding users are certain that the information exists in the document collection. Teevan et al. [37] analysed web queries of 114 users from a one-year period and found that 40% of requested

queries were re-finding queries. Re-finding is prevalent in Unix commands [17], library book search [6] and web-search [37]. To re-find information there are existing tools such as

- HayStack [1], which is similar to a personal notebook where users gather and store information like corpus, document information, meta information, user information and use these pieces of information for personalising their information retrieval. Haystack uses links between documents with similar content and user interactions to find relevant items from the stored information.
- LifeStream [15] is a time ordered system where users collect and manage information created by themselves. Lifestream uses techniques such as stream filter to organise, locate, summarise and monitors incoming information. Items displayed in the user interface are ordered by time such that the tail contains past documents and the head contains recent documents. Lifestream uses different colours and animations to indicate important document features such as read, unread, edited and writeable.
- Stuff I've Seen [13] is a system which uses information a user has seen before and recommends that information as clues to recognise them. It has five main components: 1. *Gather* shows contents such as email, web page, document and appointment in their original format. 2. *Filter* reads the input file and creates streams for processing. 3. *Tokenizer* tokenizes streams to words. 4. *Indexer* creates an index of the tokenised tokens. 5. *Query language* uses a boolean query to recommend relevant information.

The existing re-finding tools discussed above use past information of single user only to re-find relevant items, and use boolean query and filter software to recommended relevant items. To overcome this limitations, feedback on recommended items can be used to model user search intent and past information of multiple users can be used to find relevant information.

2.4 Collaborative Search

The process of seeking information with the help of more than one users with common information need is known as collaborative search [28]. Most traditional information retrieval systems are single-user systems where the information generated by single user is used to model the user's need. Studies

have shown that users actively collaborate in search activities [27] [28]. In collaborative search, users are benefited by the wisdom of the crowd, known as *knowledge sharing* [10].

Collaborative search can be classified into explicit or implicit based on user intention [30]. In explicit collaborative search, users are aware of the common information need. Implicit collaborative search is a passive collaboration where search performance is improved by using relevant past search information from other users without the user being aware. Collaborative filtering is an example of an implicit collaboration where the system recommends items based on the preference of other users expressed on those items.

2.5 Evaluation

Accuracy is the simplest measure used in machine learning algorithms to evaluate the model performance. But in information retrieval, data is extremely skewed: 99.9% of documents are in the not relevant category [23], so maximum accuracy can be achieved trivially by assigning all the documents as irrelevant which is not the user's need.

There are several measures in information retrieval, such as precision, recall, ROC (Receiver Operating Curve), mean average precision and F-measure. Precision and recall are popular evaluation metrics in information retrieval community.

Precision (P) is defined as the ratio of number of relevant items retrieved to the number of retrieved items, such that

$$Precision = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})},$$

where $\#$ is count function.

Recall(R) is defined as the ratio of relevant items retrieved to number of relevant items, such that

$$Recall = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})},$$

It measures the ability to find relevant items in collection.

F-measure is the weighted mean of precision and recall. Mathematically,

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P+R}, \text{ where } \beta^2 = \frac{1-\alpha}{\alpha}$$

The default balanced F-measure, or F1-score, corresponds to equal weights of precision and recall, which means $\alpha = \frac{1}{2}$ and $\beta = 1$. Therefore F1-score is

$$F_1 = \frac{2PR}{P+R}.$$

Chapter 3

Contextual Multi-Armed Bandit Problem

Studies have shown that relevance feedback in information retrieval system helps in improvement in search performance but often leads to a context trap; as a result, the system does not recommend new items and exploits confined information space [18] [19]. One approach to avoid this problem is to explore the recommended items while exploiting the current information to maintain search quality. In an interactive information retrieval system, exploration and exploitation are balanced by formulating user search intent modeling as reinforcement learning [16] [33] or contextual multi-armed bandit problems [9] [12] [20].

In this chapter, I present the mathematical formulation of the contextual multi-armed bandit problem, exploration-exploitation dilemma and an approach to control exploration and exploitation dilemma using Thompson sampling.

3.1 Mathematical Formulation

Multi-Armed Bandit Problem (MABP) [32] is a sequential decision making problem in which an agent tries to optimise his decisions to maximize his overall reward. For example, MABP describes how a gambler decides which arm to pull and how many times, in order to maximize his profit. The gambler needs to make decision at each iteration t , $t \in 1, 2, 3 \dots T$ to pull arm a_t from finite set $a \in 1, 2, 3 \dots N$, where N is the total number of arms. In each iteration t the agent gets reward r_t associated with the arm a_t . The distribution of rewards at any arm is unknown. An algorithm to choose the next arm given information (I) available to the agent is known as a policy. I

is set of past information represented by tuples (r_t, a_t) . The objective of the gambler is to maximize the cumulative expected reward given the available information. Mathematically,

$$\max \left[\sum_{t=1}^T r_t \right].$$

Contextual multi-armed bandit problem [21] [39] is the special type of MABP where the agent observes feature vector associated with each arm, known as the context. The reward from each arm is function of the context associated with each arm. Agent uses context and reward information collected from the previous play to choose the next arms. The main objective of the agent is to collect enough information about the relationship between the reward and the context associated with each arm. Problems such as clinical trials [39], on-line advertisement display [9] and news article recommendation [22] can be formulated contextual multi-armed bandit problem.

Let us assume that in each iteration the gambler observes context x_t and pulls arm a_t such that information I_t is described by triplets $I_t = (x_t, r_t, a_t)$. The expected reward has distribution $p(r_t|I, \theta)$ depending on some policy, parametrized by θ and information I . After T iterations, using Bayes theorem, the posterior distribution of θ is

$$p(\theta|x, a, r) \propto p(\theta) \prod p(r_t|x_t, a_t, \theta).$$

where $p(\theta)$ is the prior distribution on θ . The parameter θ is unknown.

The objective of the gambler is to pull arms such that it maximises his expected reward. The gambler can maximise his immediate reward by pulling the arm a such that,

$$\max_a \int \mathbf{E}[r|a, x, \theta] p(\theta|x, a, r) d\theta.$$

By pulling the arm a , the gambler maximises the immediate reward but does not guarantee to maximise cumulative expected reward. Due to the uncertainty, gambler has following options:

1. Exploration: Try out each arm to find the best one.
2. Exploitation: Pull the best arm according to current information believing

it gives the best reward.

3. Balance exploration and exploitation: It is not possible to exploit and explore at same time. Gambler can balance exploration and exploitation to maximize cumulative reward.

3.2 Thompson Sampling

In sequential decision-making process where the distribution of rewards is unknown, there is always a dilemma between exploration and exploitation. Thompson sampling is one of several approaches to balance exploration and exploitation. Thompson sampling is a Bayesian optimisation technique [5], first proposed by Thompson [38] in his work Randomised Bayesian algorithm. Thompson sampling was considered as a heuristic approach but later studies demonstrated both theoretical [2] and empirical performance guarantee [9]. There have been empirical studies which show Thompson sampling outperforms other popular alternatives such as UCB [8] in information retrieval tasks, such as advertisement selection and news article recommendation [9].

The basic idea of Thompson sampling is to randomly sample θ from its uncertainty and select the arm which gives maximum reward using the sampled θ , such that

$$a_t = \arg \max_a \mathbf{E}[r|x_t, a, \theta_t].$$

Pseudo code for the Thompson sampling which summarises all the above steps presented as Algorithm 1.

Algorithm 1: Thompson Sampling

```

1 I = {}
2 for j = 1 to T do
3   Get context  $x_t$ 
4   Draw  $\theta_t \sim p(\theta|I)$ 
5   select arm,  $a_t = \operatorname{argmax}_a \mathbf{E}[r|x_t, a, \theta_t]$ 
6   I = I  $\cup$  ( $x_t, a_t, r_t$ )

```

Chapter 4

Interactive User Intent Modeling

The process of modeling users information need and using that model for improving search quality and performance is known as user intent modeling in information retrieval. One approach of modeling users' intent is by involving users to interact with their current search intent, known as interactive user intent modeling. User interaction on recommended items by estimated current search intent helps to update the model throughout the session. Such model starts with the initial user model created based on the user queries and is updated based on user's feedback in each iteration.

In this Chapter, I introduce different existing interactive approaches developed to model user intent. Section 4.1 introduces an interactive user model where users interact with recommended keywords, followed by a model where users interact with recommended documents along with keywords in Section 4.2.

4.1 Keywords Interaction System

Studies have shown that users have difficulty in forming precise queries in about 50% of the search sessions [36]. To help users form queries, Glowacka et al. [16] developed an approach where users start with approximate query and adjust it throughout the session by giving relevance feedback to keywords displayed in the user interface. Their approach was a combination of the machine learning algorithm and interactive user interface design. The main idea of the user interface was to display the keywords which represent the user search intent and let users interact with the displayed keywords after typing initial query. In the user interface, keywords were displayed in a circle as shown in the Figure 4.1. In this user interface, keywords closer to the center are more relevant. Therefore, users can give explicit positive feedback

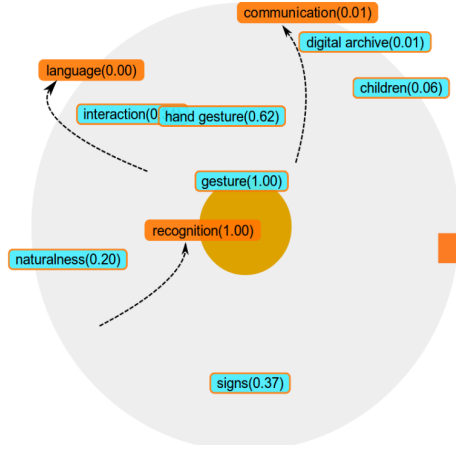


Figure 4.1: Interaction with keywords [16]. The orange colored circle at center represents user model. Dotted arrow from towards the center represents relevant keyword being drag toward the center; dotted arrow away from center represents an irrelevant keyword being dragged away.

by dragging relevant keywords towards the centre (in Figure 4.1 the user drags 'recognition' towards the centre) and negative feedback by dragging non-relevant keywords away from the centre (in Figure 4.1 the user drags 'language' and 'communication' far from centre).

The relation between expected relevancy of keywords k_j and its feature vector x_j^k was assumed to be

$$\mathbf{E}[r_j] = (x_j^k)^T \cdot \theta$$

where θ is unknown weight vector. The weight vector θ is estimated by applying machine learning algorithm LinRel [3], a special type of reinforcement learning [35] algorithm.

Document is ranked by the conditional probability

$$P(k|M_{di}) = \prod_{k_j \in k} \hat{\theta}_j P'_{mle}(k_j|M_{di}),$$

where k is keyword vector weighted by $\hat{\theta}_j$ for each keyword, M_{di} is language model for the document d_i and $P'_{mle}(k_j|M_{di})$ is the estimated maximum likelihood of keyword k_j for given M_{di} .

Furthermore, Ruotsalo et al. [33] improved model to display keywords of users current search intent and estimated future search intent in radar display as shown in Figure 4.2. The center of radar (grey) represents the current search intent and the outer grey area represents estimated future

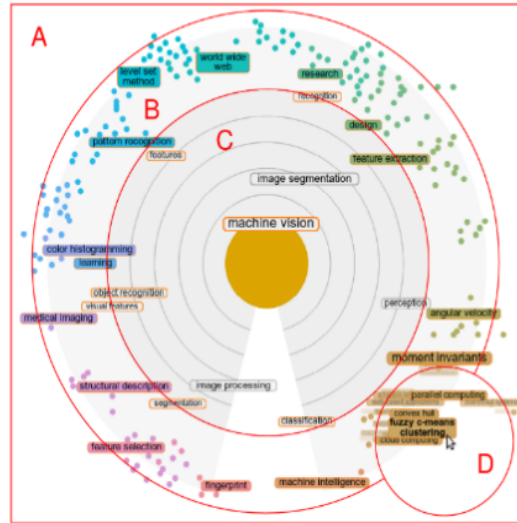


Figure 4.2: Radar Visualization [33]. Search intents visualized through keywords (A). Projected future intents are visualised in outer circle (B); Current intent visualized in inner grey circle (C). Keywords inspected with fish eye lens (D).

search intent as shown in Figure 4.2. In the radar structure, the radius gives the relevance of the keywords and angle gives the similarity between the keywords.

4.2 Documents and Keywords Interaction System

The previous model models user's intent using keyword feedback only. Daee et al. [12] developed an approach where users can provide feedback on multiple item types (keywords and documents) instead of one item (keywords). In their work they have demonstrated how adding extra item type improves search performance.

They use document-keyword matrix $M_{|D| \times |K|}$ to express the relationship between keywords and documents, such that

$$M_{|D| \times |K|} = \begin{pmatrix} P(k_1|d_1) & \cdots & P(k_{|K|}|d_1) \\ P(k_1|d_2) & \cdots & P(k_{|K|}|d_2) \\ \vdots & \ddots & \vdots \\ P(k_1|d_{|D|}) & \cdots & P(k_{|K|}|d_{|D|}) \end{pmatrix}$$

where $P(k_i|d_j)$ is likelihood of document d_j generating keyword k_i . $|D|$ is document count and $|K|$ is keywords count.

Dae et al. [12] have made following assumptions for simplicity:

1. Relationship between expected relevancy of documents and keywords is as follows:

$$\mathbf{E}[r_j^d] = \sum_{i=1}^{|K|} \mathbf{E}[r_i^k] P(k_i|d_j)$$

in matrix notation

$$\mathbf{E}[R^d] = M\mathbf{E}[R^k]$$

2. Expected relevance of keywords have linear relationship with its feature vector, such that

$$\mathbf{E}[R^k] = M^T\theta$$

where θ is unknown weights.

3. Based on the two assumptions above, expected relevance of documents is

$$\mathbf{E}[R^d] = MM^T\theta$$

Based on these three assumptions, Dae et al. [12] formulated the problem of finding relevant keywords and documents as bandit problem. The relevance (reward) distribution of keywords and documents is the function of a feature vector, x^d for the document and x^k for the keyword, and the parameter θ . x^k is the k^{th} column from matrix M and x^d is the d^{th} column from matrix MM^T . In each iteration, users give feedback to the displayed keywords and documents and $\hat{\theta}$ is estimated. Thompson sampling has been used to balance exploration and exploitation.

4.3 Limitation of Existing Systems

Teevan et al. [37] analysed web queries of 114 users from a one-year period and found that 40% of requested queries were re-finding queries. Studies have also shown that users actively collaborate in search activities [27] [28]. Therefore, information re-finding and collaborative search are two common activities in information retrieval. However, all the three systems discussed above are single user activity systems and do not use any past information to improve search performance.

Chapter 5

Proposed Model

Section 4.3 discusses the limitation of existing user intent models. In this thesis, a model developed by Dae et al. [12] has been extended to infer user search intent by using session feedback in addition to document and keyword feedback. The extended model can be used in both single user and collaborative setting.

Section 5.1 presents simplifying assumptions of this model, followed by representation of the model as the bandit problem in Section 5.2. Section 5.3 gives probabilistic model of proposed model. Section 5.4 presents overall search process of the proposed system.

5.1 Model Assumption

The proposed model extends simplifying assumptions stated in the model developed by Dae et al. [12], in order to find expected relevancy scores of past search sessions. The simplifying assumptions are:

1. Relationship between expected relevancy of documents and keywords is as follows:

$$\mathbf{E}[r_j^d] = \sum_{i=1}^{|K|} \mathbf{E}[r_i^k] P(k_i|d_j).$$

where $P(k_i|d_j)$ is likelihood of document d_j generating keyword k_i and M document-keyword matrix. In matrix notation

$$\mathbf{E}[R^d] = M\mathbf{E}[R^k],$$

where

$$M_{|D| \times |K|} = \begin{pmatrix} P(k_1|d_1) & \cdots & P(k_{|K|}|d_1) \\ P(k_1|d_2) & \cdots & P(k_{|K|}|d_2) \\ \vdots & \ddots & \vdots \\ P(k_1|d_{|D|}) & \cdots & P(k_{|K|}|d_{|D|}) \end{pmatrix}.$$

- Expected relevancies of keywords have linear relationship with their feature vector such that,

$$\mathbf{E}[R^k] = M^T \theta,$$

where θ is an unknown weight vector.

- Based on assumptions 1 and 2, expected relevance of documents is given as

$$\mathbf{E}[R^d] = MM^T \theta.$$

- A past search session is likely to be relevant if the estimated relevancies of keywords are similar. The expected relevancy of the j^{th} session with respect to current session c is given by

$$E[r_j^s]_c \propto sim(E[r^k]_j, E[r^k]_c)$$

where $E[r^k]_c$ and $E[r^k]_j$ are expected relevancies of keywords in c^{th} and j^{th} sessions respectively. The function sim is a similarity function between two vectors which gives their dot product such that,

$$sim(a, b) = a \cdot b$$

where a and b are the vectors. Replacing the vectors by $E[r^k]_j$ and $E[r^k]_c$, we get

$$sim(E[r^k]_j, E[r^k]_c) = \theta_j^T MM^T \theta_c.$$

Expected relevancy of the j^{th} session with respect to current session c is given by

$$E[r_j^s]_c = P(s_j|s_c) = \frac{1}{z} sim(E[r^k]_j, E[r^k]_c)$$

where

$$\begin{aligned} \frac{1}{z} \text{sim}(E[r^k]_j, E[r^k]_c) &= \frac{1}{z} \theta_j^T M M^T \theta_c, \\ z &= \sum_j \text{sim}(E[r^k]_j, E[r^k]_c), \\ E[R^s]_c &= \begin{pmatrix} P(s_1|s_c) \\ \vdots \\ P(s_{|S|}|s_c) \end{pmatrix} = \begin{pmatrix} E[r_j^s]_c \\ \vdots \\ E[r_{|S|}^s]_c \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{z_1} \theta_1^T M M^T \theta_c \\ \vdots \\ \frac{1}{z_{|S|}} \theta_{|S|}^T M M^T \theta_c \end{pmatrix} = M_s \end{aligned}$$

5.2 Proposed Model as Contextual Multi-Armed Bandit Problem

In the proposed model, relevance (reward) distribution for documents, keywords and session are,

$$\begin{aligned} p(r^d|x^d, \theta), \\ p(r^k|x^k, \theta), \\ p(r^s|x^s, \theta), \end{aligned}$$

where x^d, x^k, x^s are feature vectors of documents, keywords and past search sessions respectively. After n^d document feedbacks, n^k keyword feedbacks and n^s session feedbacks, the posterior distribution of θ is given as

$$p(\theta|x^d, x^k, x^s, r^d, r^k, r^s) \propto p(\theta) \prod_{d \in n^d} p(r^d|x^d, \theta) \prod_{k \in n^k} p(r^k|x^k, \theta) \prod_{s \in n^s} p(r^s|x^s, \theta),$$

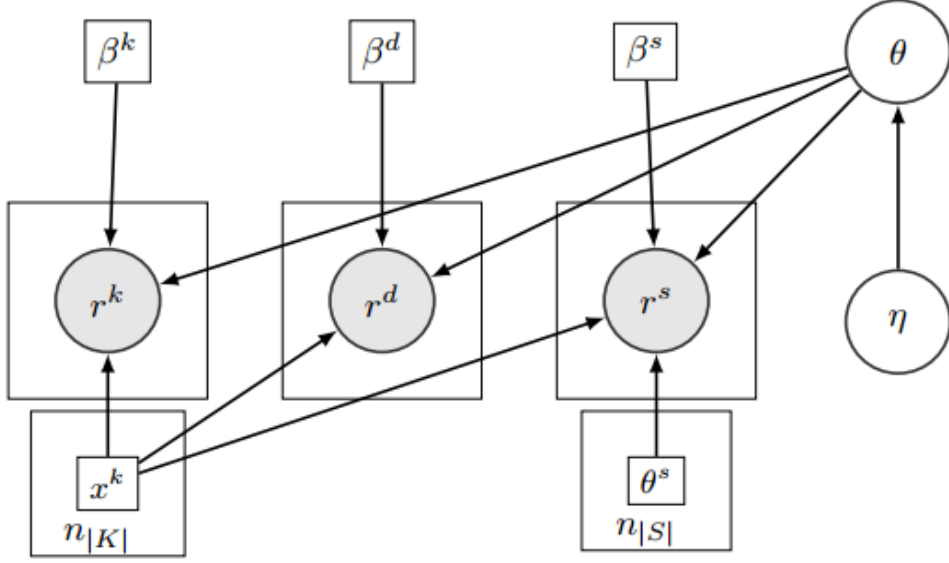


Figure 5.1: Probabilistic model for user feedback on documents, keywords and past search sessions.

where $p(\theta)$ is the prior distribution for θ . In order to apply Thompson sampling:

1. A sample θ is drawn from the posterior distribution;
2. Expected relevancy scores of documents, keywords and past sessions are evaluated in each iteration.

5.3 Probabilistic Model

To estimate the value of unknown weight θ , I formulate a simple probabilistic model which considers relevance score for documents, keywords and past search sessions as a Gaussian distribution, such that

$$\begin{aligned} p(r^k|x^{kT}, \theta, \beta_K^2) &= \text{Normal}(r^k|(x^k)^T\theta, \beta_K^2), \\ p(r^d|x^{dT}, \theta, \beta_D^2) &= \text{Normal}(r^d|(x^d)^T\theta, \beta_D^2), \\ p(r^s|x^{sT}, \theta, \beta_S^2) &= \text{Normal}(r^s|(x^s)^T\theta, \beta_S^2), \end{aligned}$$

where x_j^d, x_i^k, x_l^s are feature vectors of documents, keywords and past sessions respectively. x_i^k is the i^{th} column of M , x_j^d is j^{th} column of MM^T and x_l^s is l^{th} row of M_s . The prior distribution for θ is Gaussian distribution with

mean 0 and variance η^2 ; Mathematically,

$$p(\theta) = \text{Normal}(\theta; 0, \eta^2 I).$$

Both prior and likelihood are from Gaussian distribution so the posterior has closed form solution as Gaussian distribution with mean μ and variance Σ , such that

$$\begin{aligned} \Sigma^{-1} &= \beta^{d-2}(X_n^d)^T X_n^d + \beta^{k-2}(X_n^k)^T X_n^k + \beta^{s-2}(X_n^s)^T X_n^s, \\ \mu &= \Sigma(\beta^{d-2}(X_n^d)^T R_n^d + \beta^{k-2}(X_n^k)^T R_n^K + \beta^{s-2}(X_n^s)^T R_n^s). \end{aligned}$$

Algorithm 2 summarizes working of the proposed model. In each iteration θ is sampled from the posterior of the θ . Using sampled θ , expected relevancies of documents, keywords and sessions are calculated. N documents, keywords and sessions with high relevancies are displayed in user interface and user feedback is collected (lines 3-5). Based on user feedback on recommended items, the posterior is updated and θ is sampled for next iteration.

Algorithm 2: Proposed Model

```

1 for each iteration do
2   Draw  $\theta^d \sim p(\theta|x^d, x^k, x^s, r^d, r^k, r^s)$ 
3   for session arm: select  $s^+ = \arg \max[1 : N]_{s \in S} x_s^T \theta^d$ 
4   for document arm: select  $d^+ = \arg \max[1 : N]_{d \in D} x_d^T \theta^d$ 
5   for keyword arm: select  $k^+ = \arg \max[1 : N]_{k \in K} x_k^T \theta^d$ 
6   update the posterior based on the user feedback and observed
   feature vectors.

```

5.4 Search Process of Proposed System

The proposed system infers the user intent from feedbacks given to recommended documents, keywords, and past sessions. Users provide feedback on recommended items displayed in user-interface, based on which the proposed model updates inferred the user intent and calculates expected relevancy of items. Items are ranked and recommended by their expected score and displayed in user-interface in each iteration. Figure 5.2 summaries the search process of proposed system.

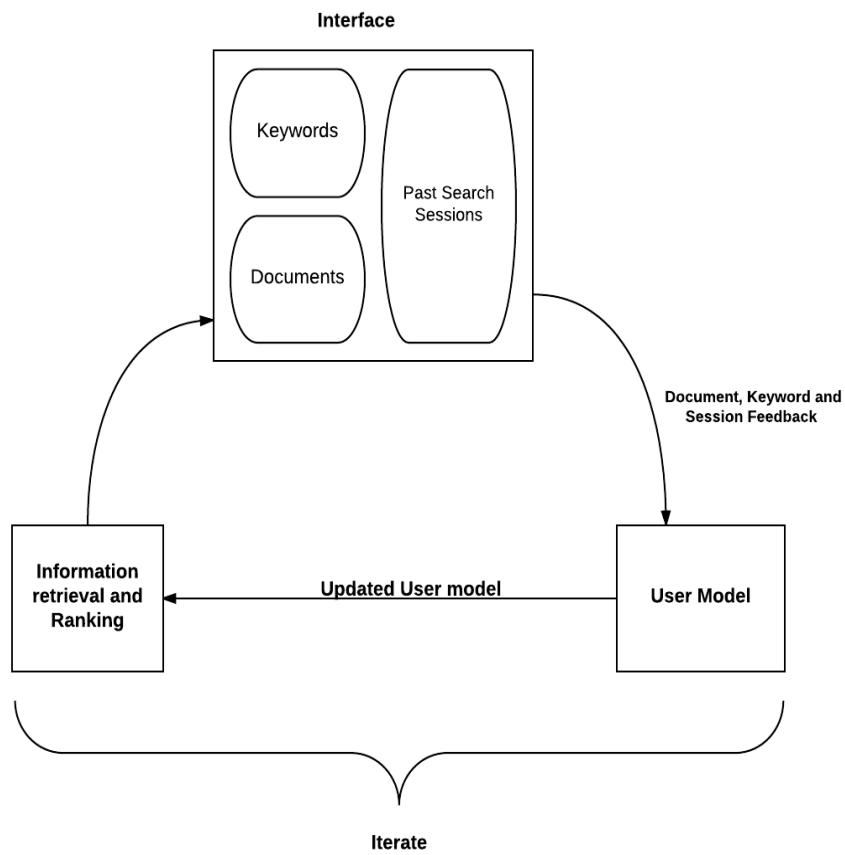


Figure 5.2: Search process of proposed system. Directed arrows shows flow of data between the blocks.

Chapter 6

Simulation Environment

In this thesis, I evaluate performance of the proposed model by simulating an interactive information retrieval system. In this Chapter, I report simulation of documents, keywords and past sessions ground-truth, simulation of document-keyword matrix M and simulation of past search sessions.

Section 6.1 describes the dataset used to create document-keyword matrix M . Section 6.2 presents how document, keyword and past session ground-truth is simulated, followed by simulation of document-keyword matrix and simulation of past sessions in Section 6.4. Finally, Section 6.5 presents evaluation metrics used to evaluate simulated results.

6.1 Dataset

Document-keyword matrix M is constructed from categorical data prepared by parsing an XML file which contains documents collected from the ACM digital library database, the actual query that has been used to find the document and pre-evaluated relevancy (1 for positive and 0 for negative) for given query (Source of XML data file: Information Retrieval course (ME-E4400) at Aalto University). Each query has list of relevant and irrelevant documents for that query. The XML file is parsed and documents are categorised and stored based on the queries. Further categorical dataset was manually cleaned by removing the empty category, empty documents and categories with fewer documents. The final dataset is categorical data with 29 categories based on the query used to find that document in ACM-database. In simulation, simulated user selects at random only one category from the search target in each session.

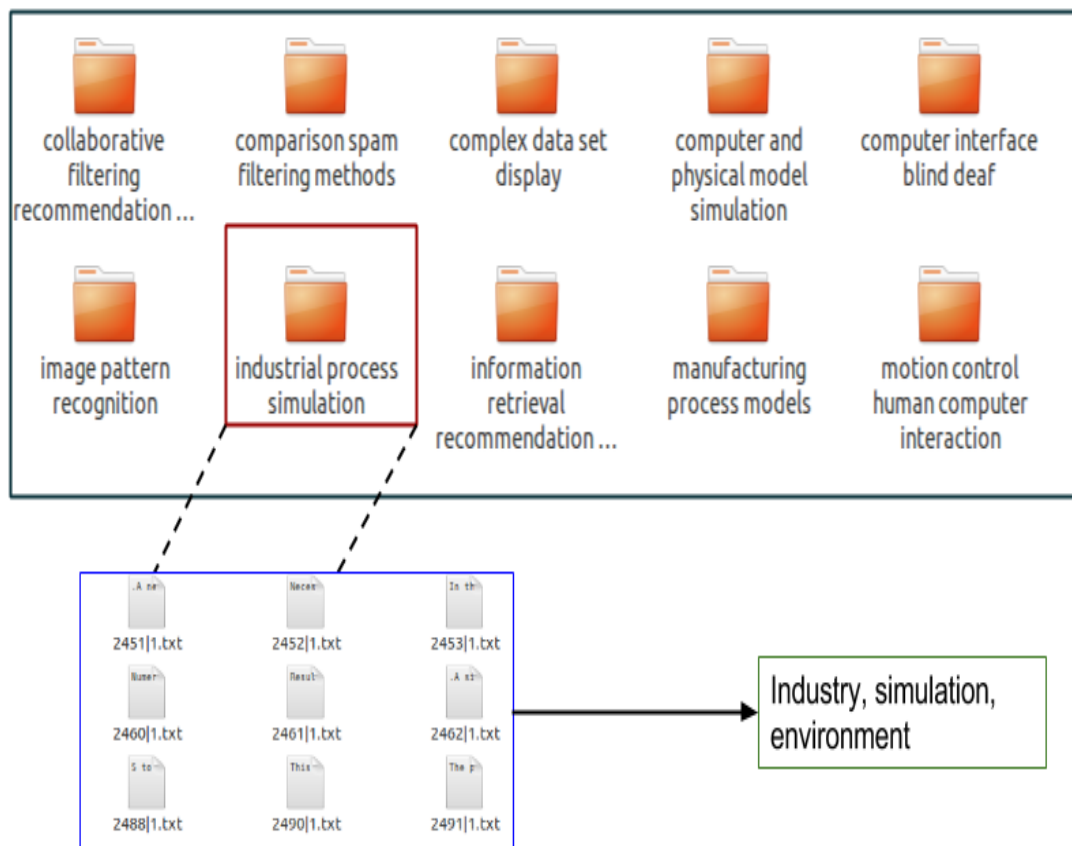


Figure 6.1: Example of search target, document ground-truth and keyword ground-truth. The Black box (top) represents categorical dataset. Name of the folder in the black box is categorical label for the set of documents stored in that folder, for example documents in the blue box (bottom-left) is a set of documents under the category *industrial process simulation*. Red box (box inside box in top) indicates search target, blue box indicates document ground truth for selected target group and green box indicates keyword ground truth for selected target group.

6.2 Simulation of Ground-truth

In search session simulations, ground-truth documents and keywords allow us to simulate respective feedbacks, i.e a positive feedback if the recommended document or keyword is part of the ground-truth, and a negative feedback if otherwise. Documents in each category serve as the *ground-truth* documents for the search target. Also, twenty keywords with high $TF-IDF$ weight in the search target are ground-truth keywords. In the example shown in Figure 6.1, *industrial process simulation* is a search target, and the documents in the folder (shown in the blue box) are the document ground-truth. In the same example, keywords in green box are the keyword ground-truth.

Similarly, ground-truth sessions for the purpose of generating feedback for recommended sessions are constructed from multiple session simulations using document and keyword feedbacks, to be discussed in 6.3. These ground-truth sessions are categorised and stored by their respective category names.

6.3 Simulation of Document-Keyword Matrix and Past Search Session

Simulated document-keyword matrix M is the normalised $TF-IDF$ representation of the documents collection. The Matrix M is the language model which gives conditional probabilities of each keyword in a given document.

Past search session is simulated using only document and keyword feedbacks. I assume that in each past search session, user interacts with thirty keywords and thirty documents among which fifteen are relevant and fifteen are irrelevant items for each type. To simulate documents and keywords positive feedback, fifteen items are selected in random from document and keyword ground-truth. Similarly for negative feedback, fifteen items which are not in documents and keywords ground-truth are randomly selected. Simulated search sessions are grouped and stored by their target category name.

Based on past search sessions, in addition to single-user setting, the proposed model can be used in collaborative setting as well, where past search sessions are created by other users. In this thesis, for simplicity past search sessions are assumed to be created only by single-user.

6.4 Evaluation

The proposed model helps users find both relevant documents and re-find past search sessions and its performance is evaluated against a similar model which does not use session feedback, developed by Dae et al. [12], as the baseline . The baseline model does not help in re-finding search sessions, so I evaluate performance trend of re-finding search sessions across iterations.

Performance in finding is evaluated using *document F1-score* compared against baseline model. Re-finding past search session is evaluated using *session F1-score*.

Chapter 7

Analysis of Results

Search performance of the proposed model is evaluated by conducting different simulation experiments for both finding relevant documents and re-finding relevant sessions.

7.1 Experiment 1: Session-Feedback in Addition to Document and Keyword Feedback

7.1.1 Setting

Assumption: *Number of feedback given by the user depends on number of item types recommended to the user.*

In this experiment setting, simulated user in proposed system gives feedback to past search sessions in addition to documents and keywords. The performance of proposed model is evaluated against the baseline where feedback is given to documents and keywords only. The numbers of document and keyword feedback on both systems are equal. In proposed system, simulated user gives 4, 4, 2 feedbacks to document, keyword and session respectively, whereas in baseline model simulated user gives the 4 feedbacks each to documents and keywords in each iteration as shown in Figure 7.1.

7.1.2 Results

Figure 7.2 shows simulated results from above experiment setting. The documents F1-score plot (left) shows that session feedback in addition to keyword and document feedback marginally improves the performance in finding relevant documents. This result shows that there is improvement in document F1-score after twentieth feedback iteration.

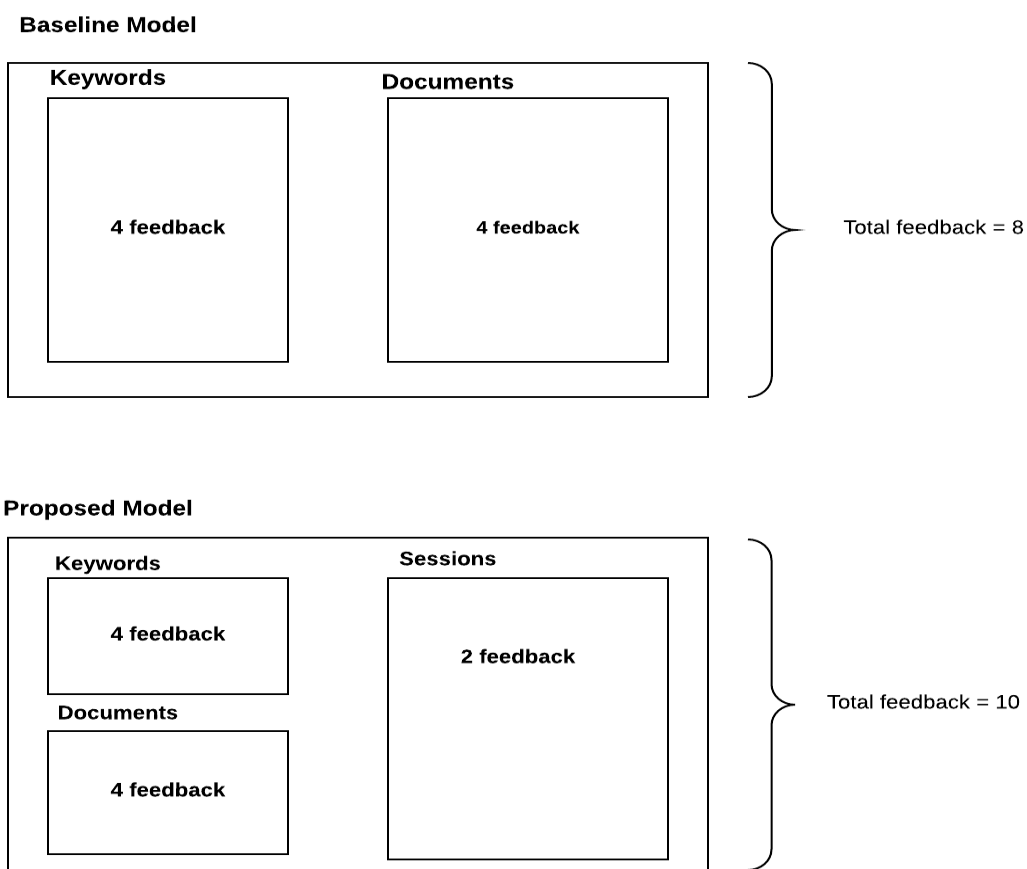


Figure 7.1: Distribution of total number of feedback in proposed and base-line systems for Experiment 1.

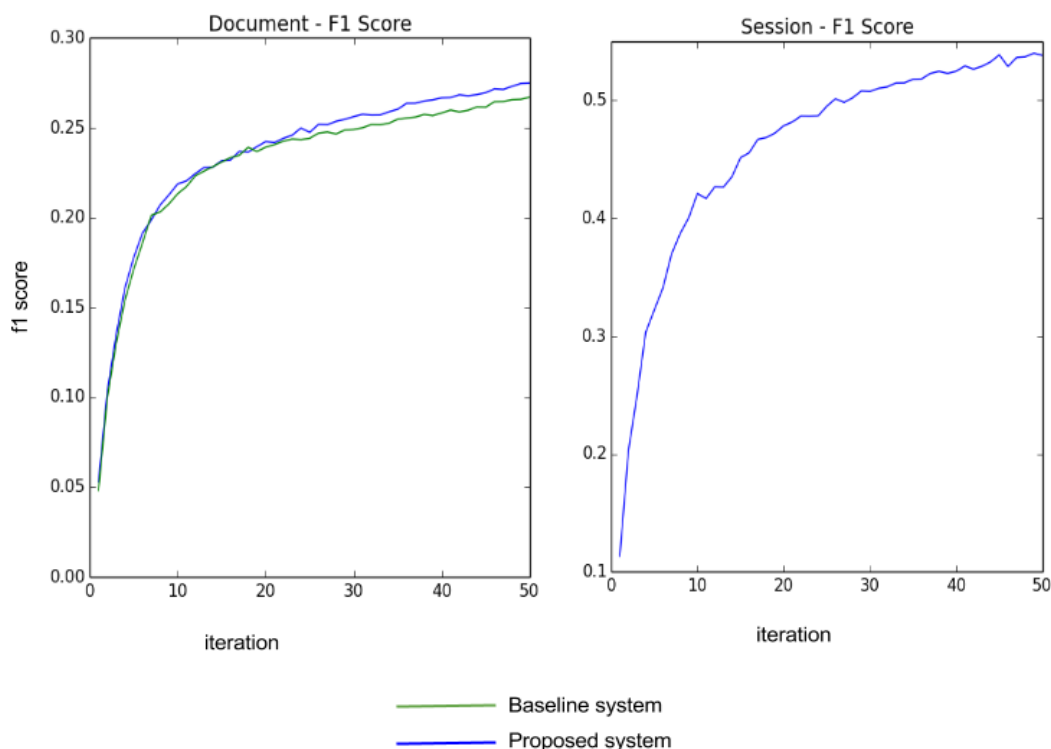


Figure 7.2: Adding session feedback marginally improves the performance in finding documents. F1-score is average over 500 simulated sessions and 50 feedback iteration. X-axis represents feedback iteration and y-axis represents the corresponding F1-score. The blue line in plots represents proposed system and green line represents baseline system. Left: Document F1-score for each feedback iteration. Right: Session F1-score for each feedback iteration.

Session F1-score plot (Figure 7.2 right) shows session F1-score in each iteration in experiment 1. Results shows that sessions F1-score increases across feedback iteration in experiment 1 setting.

7.2 Experiment 2: Session-Feedback Instead of Other Feedback

7.2.1 Setting

Assumption: *Number of feedback given by the user is equivalent to time spent in each iteration.*

In previous experiment setting, the proposed system gets additional 2

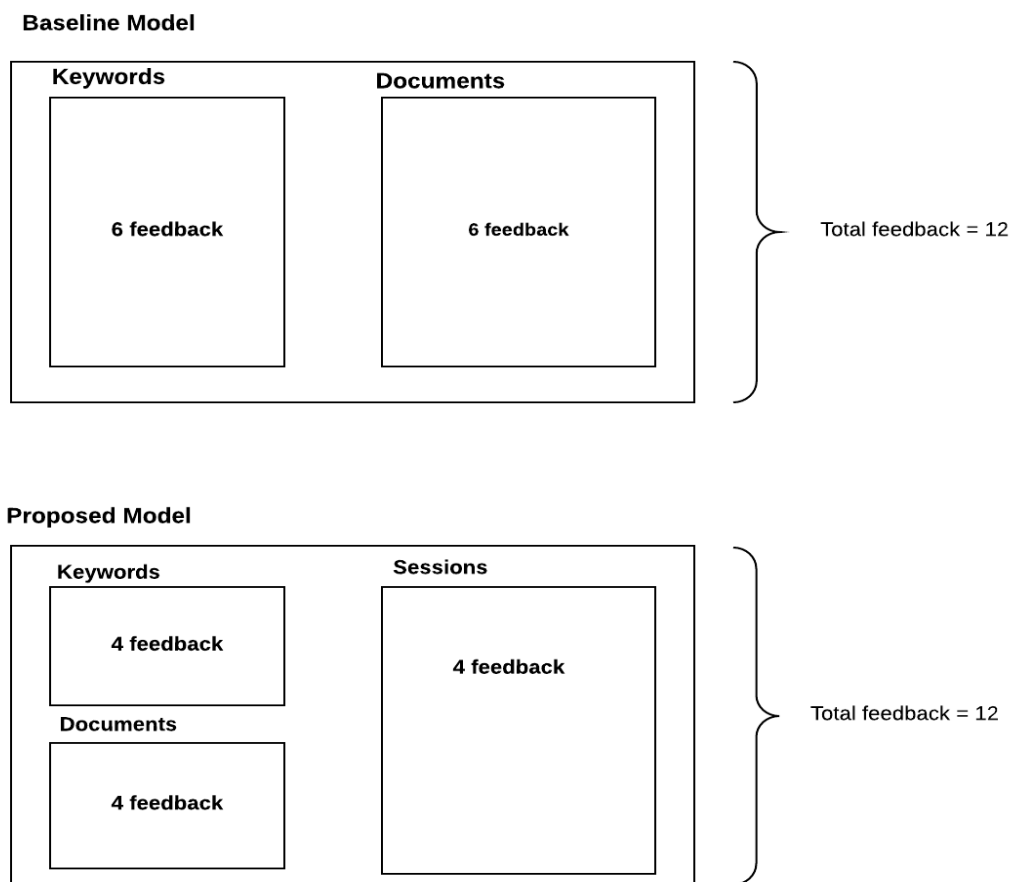


Figure 7.3: Distribution of total number of feedbacks in the proposed system and the baseline system for Experiment 2.

feedback in each iteration compared to baseline. In this experiment, simulated users give equal number of feedbacks to both systems. In the baseline system, simulated user gives 6 feedbacks to documents and keywords each. In the proposed system, documents, keywords and past sessions get 4 feedbacks each. The reduced number of feedbacks to keywords and documents are compensated by session feedback such that the total remains unchanged in each iteration, as shown in Figure 7.3.

7.2.2 Results

Figure 7.4 shows results of the simulation from Experiment 2 setting. It shows documents F1-score plot (7.4 left) and session F1-score (7.4 right) in each feedback iteration. Document F1-score plot shows a drop in perfor-

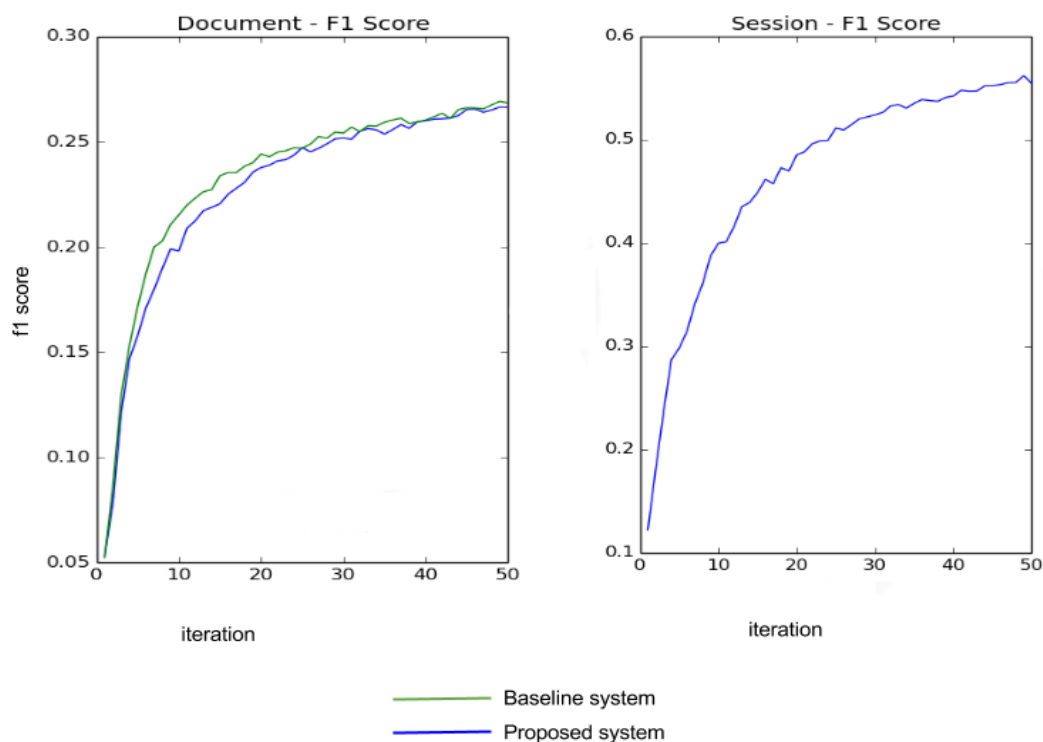


Figure 7.4: Feedback on session improves results less than feedback on documents and keywords. F1-score is average over 500 simulated sessions for 50 feedback iterations. X-axis represents feedback iteration and y-axis represents corresponding F1-score. The blue line represents the proposed system and green line represents baseline system. Left: Document F1-score for each feedback iteration. Right: Session F1-score for each feedback iteration.

mance compared to the baseline, which indicates that session feedback is less informative than document and keyword feedback.

Session F1-score plot (Figure 7.4, right) shows session F1-score in each feedback iteration. Results shows that session F1-score increases in across feedback iteration in experiment 2.

7.3 Experiment 3: Additional Numbers of Session-Feedback

7.3.1 Setting

Experiment 1 shows that additional session feedback along with document and keyword feedback marginally improves search performance of the proposed system. In Experiment 3, I examine performance of the proposed system by varying the number of session feedbacks. For example in Figure 7.5, in *2 session feedback* block, simulated user provides 2 session feedback and 4 document and keyword feedback each whereas in *4 session feedback* block, simulated user provides the same number of document and keyword feedback but 4 session feedback. In this experiment, I consider *0 session feedback* (document and keyword feedback only) as baseline.

7.3.2 Results

Simulated document F1-score result form Experiment 3 in Figure 7.6 shows that after seventh iteration there is clear improvement in document F1-score when more than 6 additional session feedbacks are given in each iteration. Also, giving session feedbacks results in improvement in F1-score after at most twenty feedback iterations.

Figure 7.7 shows performance comparison between varying number of session feedbacks on session F1-score. Result shows session feedback helps in improvement in re-finding past session after atmost eight feedback iteration.

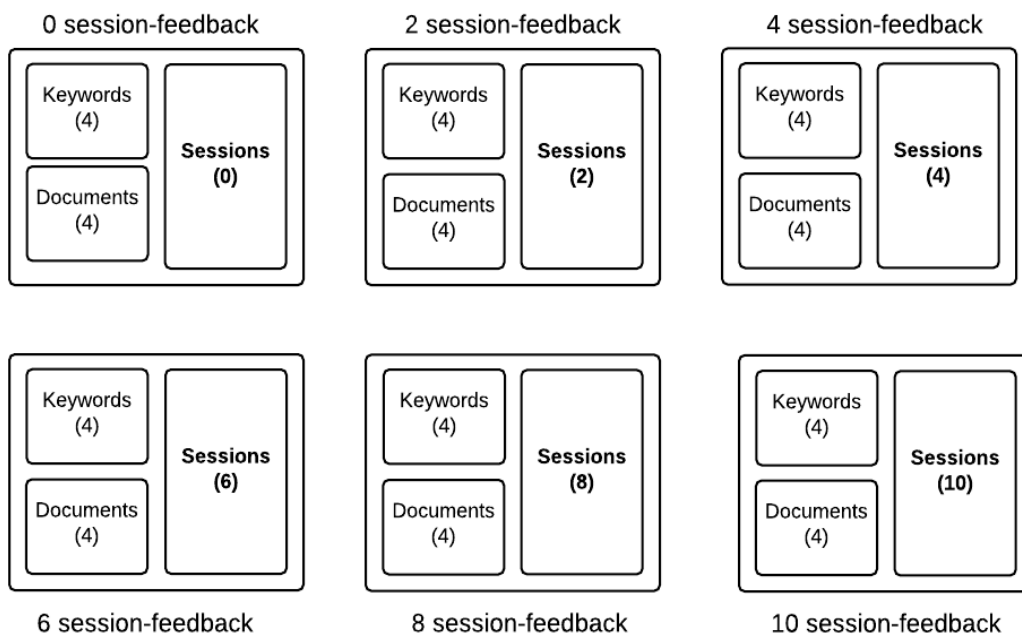


Figure 7.5: Variations in number of session feedbacks for the Experiment 3.

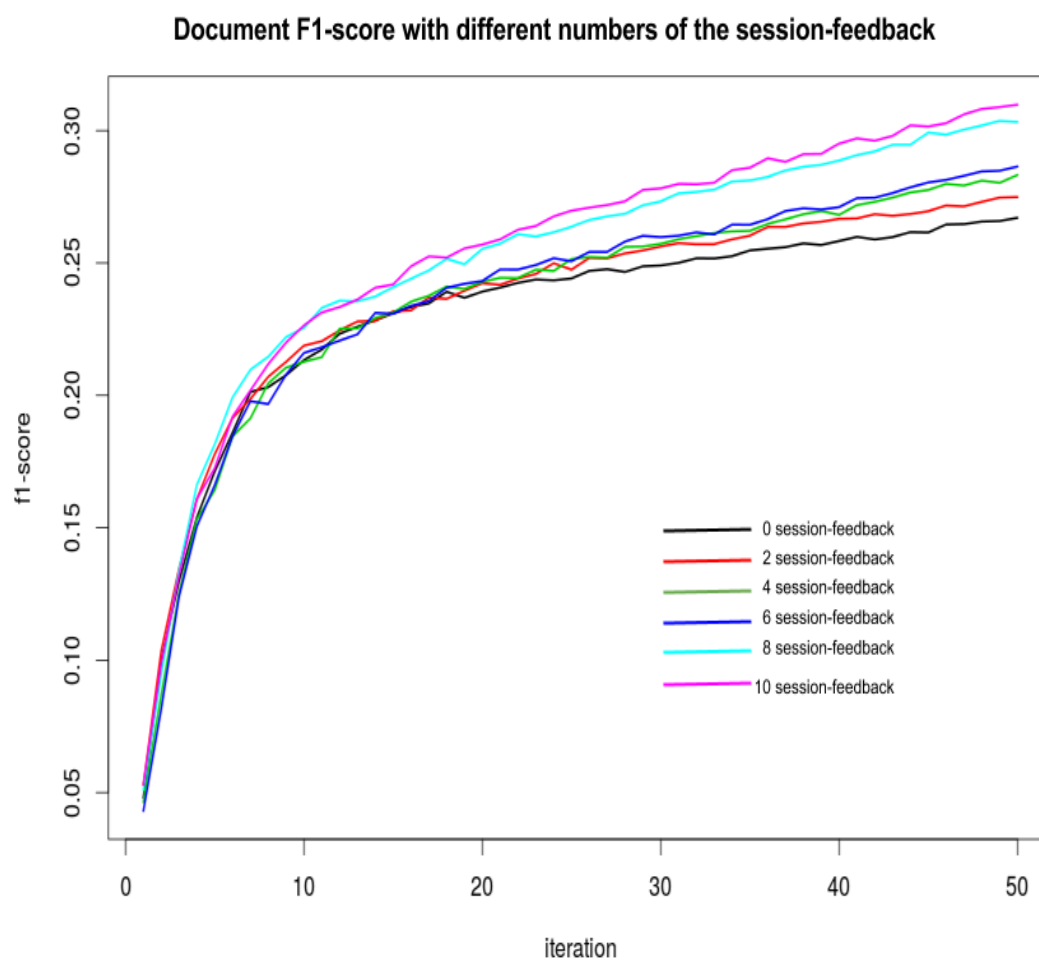


Figure 7.6: Simulated document F1-score for 50 feedback iterations, averaged over 500 search sessions. Different colors represents different number of session feedbacks (Black: 0, red: 2, green: 4, blue: 6, light blue: 8 and pink: 10). X-axis represents feedback iteration and y-axis represents the corresponding document F1-score.

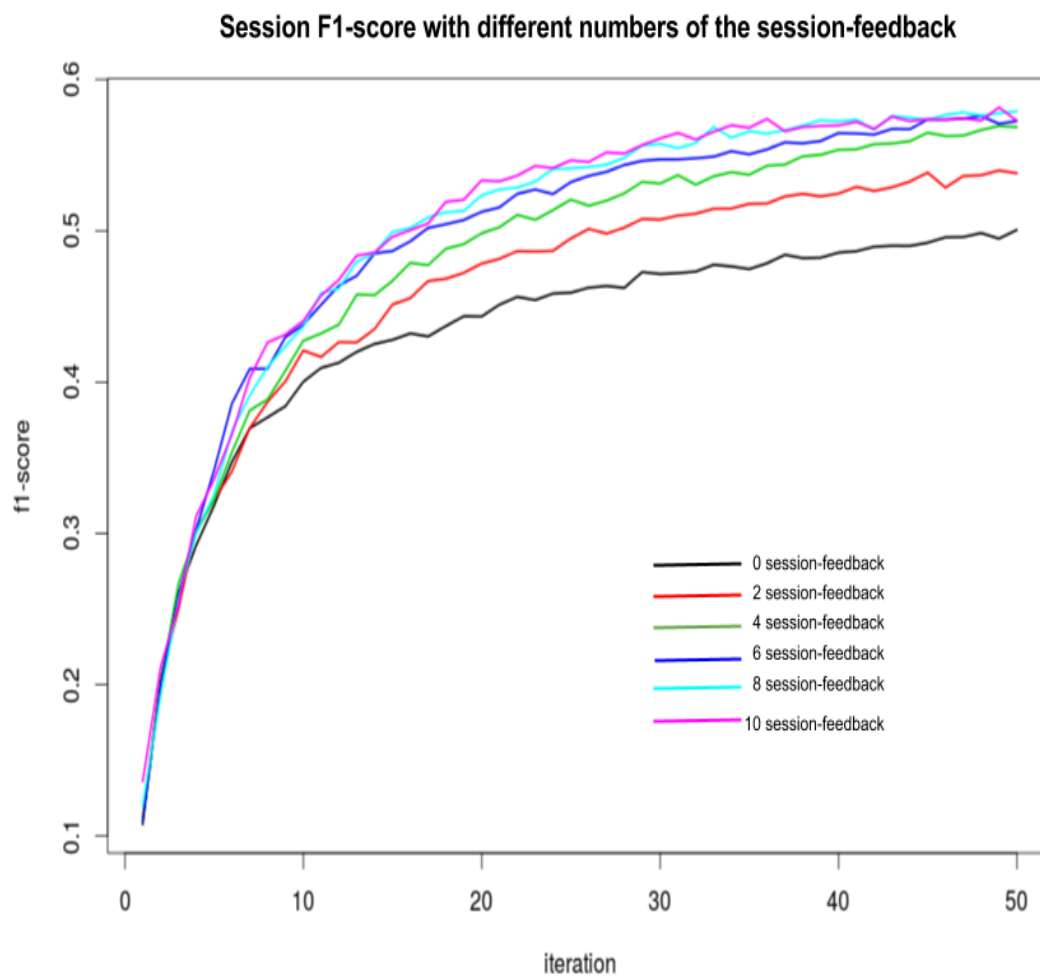


Figure 7.7: Simulated session F1-score for 50 feedback iterations, averaged over 500 search sessions. Different colors represents different number of session feedbacks (Black: 0, red: 2, green: 4, blue: 6, light blue: 8 and pink: 10). X-axis represents feedback iteration and y-axis represents the corresponding session F1-score.

Chapter 8

Conclusion

In this section I summarize the thesis and present ideas for further work.

8.1 Discussion

Dae et al. [12] demonstrated that feedback from multiple sources (document and keyword) instead of one (keyword) helps in finding the relevant information. Studies have shown that re-finding previous searched information is ubiquitous in information retrieval. Not only re-finding, collaboration is also common activity in information retrieval. Studies show that people collaborate while searching for information. In this thesis, I extended the existing model developed by Dae et al. (2016) [12] using session-feedback in addition to document and keyword feedbacks which can be used in both single user and collaborative setting. I formulated the extended model as a passive collaborative model which uses past search information of other users to improve search performance. Additionally, I investigated the usefulness of giving session-feedback by simulating various experiment settings.

The results from Experiment 1 showed that leveraging additional session-feedback along with document and keyword feedback improves document F1-score but improvement was marginal. Further, to study the effect of session-feedback, I compared performance of the model by varying the number of session-feedback. Results showed that more the session feedback, more the improvement in F1-score, which indicates that additional feedback on past search sessions helps to infer user search intent and improves quality of finding documents.

However, in experiment 2 providing session feedback instead of other feedback resulted in drop in document F1-score, which means session-level feedback is less informative than document and keyword feedback. Drop in

F1-score is due to noisy session-feedback. Past sessions are recommended based on similarity between expected relevancies of keywords in past session with respect to current session. Although past search sessions might not be relevant, few keywords in those session might be relevant. Feedback to these session is noisy which drops performance of model.

8.2 Limitation and Future Work

One limitation of this work is that all analysis in this thesis are based on simulations. In real world scenario, a user-friendly interface is required to display past search sessions in order to get feedback from users. Therefore reproducing the experiments in real-user scenarios can be part of future work. Also, investigations of performance of the model in re-finding as well as collaborative settings can follow from this work, where I only investigate performance in the case of finding in single-user setting.

As user search intent is estimated with smaller data sample compared to the dimension, use of spike and slab prior [25] can be investigated as part of later work instead of using Gaussian prior as in the current model for sparse model.

Additionally, in addition to the three types of explicit feedback considered for this work, the model can be extended in the future by using possible implicit relevance feedbacks [4].

Bibliography

- [1] Eytan Adar, David Karger, and Lynn Andrea Stein. Haystack: Per-user information environments. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 413–422, 1999.
- [2] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the International Conference on Machine Learning*, pages 127–135, 2013.
- [3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422.
- [4] Oswald Barral, Manuel JA Eugster, Tuukka Ruotsalo, Michiel M Spapé, Ilkka Kosunen, Niklas Ravaja, Samuel Kaski, and Giulio Jacucci. Exploring peripheral physiology as a predictor of perceived relevance in information retrieval. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 389–399, 2015.
- [5] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical report, University of British Columbia, Department of Computer Science, 2010.
- [6] Quentin Burrell. A simple stochastic model for library loans. *Journal of Documentation*, 36:115–132, 1980.
- [7] Robert Capra, Mary Pinney, and Manuel A Perez-Quinones. Refinding is not finding again. Technical report, Technical Report TR-05-10, Department of Computer Science, Virginia Technology.
- [8] Alexandra Carpentier, Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos, and Peter Auer. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *Proceedings of the International Conference on Algorithmic Learning Theory*, pages 189–203, 2011.

- [9] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.
- [10] Max Chevalier, Christine Julien, and Chantal Soule-Dupuy. *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling: Techniques for Improved User Modeling*. 2008.
- [11] W. Bruce Croft. Knowledge-based and statistical approaches to text retrieval. *IEEE Expert: Intelligent Systems and Their Applications*, 8: 8–12, 1993.
- [12] Pedram Daei, Joel Pyykkö, Dorota Glowacka, and Samuel Kaski. Interactive intent modeling from multiple feedback domains. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 71–75, 2016.
- [13] Susan Dumais, Edward Cutrell, Jonathan J Cadiz, Gavin Jancke, Raman Sarin, and Daniel C Robbins. Stuff i’ve seen: a system for personal information retrieval and re-use. In *Proceedings of the International Conference on Research and Development in Informaion Retrieval*, pages 28–35, 2016.
- [14] Simon Eliot and Jonathan Rose. *A Companion to the History of the Book*, volume 98. 2009.
- [15] Scott Fertig, Eric Freeman, and David Gelernter. Lifestreams: an alternative to the desktop metaphor. In *Proceedings of the Conference Companion on Human Factors in Computing Systems*, pages 410–411, 1996.
- [16] Dorota Glowacka, Tuukka Ruotsalo, Ksenia Konuyshkova, Samuel Kaski, Giulio Jacucci, et al. Directing exploratory search: Reinforcement learning from user interactions with keywords. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 117–128, 2013.
- [17] Saul Greenberg. *The computer user as toolsmith*. Cambridge Series on Human-Computer Interaction. Cambridge University Press, 1993.
- [18] Sayantan Hore, Lasse Tyrvaïnen, Joel Pyykko, and Dorota Glowacka. A reinforcement learning approach to query-less image retrieval. In *International Workshop on Symbiotic Interaction*, pages 121–126, 2014.

- [19] Diane Kelly and Xin Fu. Elicitation of term relevance feedback: an investigation of term source and context. In *Proceedings of the International Conference on Research and Development in Information Retrieval*, pages 453–460.
- [20] Andreas Krause and Cheng S Ong. Contextual gaussian process bandit optimization. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 2447–2455, 2011.
- [21] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 817–824, 2008.
- [22] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the International Conference on World wide web*, pages 661–670, 2010.
- [23] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. 2008.
- [24] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49:41–46.
- [25] Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83: 1023–1032, 1988.
- [26] Calvin N Mooers. Information retrieval viewed as temporal signaling. In *Proceedings of the International Congress of Mathematicians*, pages 572–573.
- [27] Meredith Ringel Morris. A survey of collaborative web search practices. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 1657–1660, 2008.
- [28] Meredith Ringel Morris and Jaime Teevan. Collaborative web search: Who, what, where, when, and why. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1:1–99, 2009.
- [29] NetCraft. October 2014 web server survey. <https://news.netcraft.com/archives/2014/10/10/october-2014-web-server-survey.html>.

- [30] Jeremy Pickens and Gene Golovchinsky. Collaborative exploratory search. In *Proceedings of the Human-Computer Interaction and Information Retrieval Workshop*, pages 21–22, 2007.
- [31] Kristen Purcell. Search and email still top the list of most popular online activities. <http://www.pewinternet.org/2011/08/09/search-and-email-still-top-the-list-of-most-popular-online-activities>.
- [32] Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177.
- [33] Tuukka Ruotsalo, Jaakko Peltonen, Manuel Eugster, Dorota Głowacka, Ksenia Konyushkova, Kumaripaba Athukorala, Ilkka Kosunen, Aki Reijonen, Petri Myllymäki, Giulio Jacucci, et al. Directing exploratory search with interactive intent modeling. In *Proceedings of the International Conference on Conference on information & knowledge management*, pages 1759–1764, 2013.
- [34] Pasquale Savino and Fabrizio Sebastiani. Essential bibliography on multimedia information retrieval, categorisation and filtering. In *Proceedings of the European Digital Libraries Conference Tutorial on Multimedia Information Retrieval*, 1998.
- [35] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. 1998.
- [36] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. The perfect search engine is not enough: A study of orienteering behavior in directed search. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 415–422.
- [37] Jaime Teevan, Eytan Adar, Rosie Jones, and Michael A. S. Potts. Information re-retrieval: Repeat queries in yahoo’s logs. In *Proceedings of the International Conference on Research and Development in Information Retrieval*, pages 151–158, 2007.
- [38] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25: 285–294, 1933.
- [39] Michael Woodroffe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74:799–806, 1979.