Accepted Manuscript

An adaptive contextual quantum language model

Jingfei Li, Peng Zhang, Dawei Song, Yuexian Hou

PII: S0378-4371(16)00259-4
DOI: http://dx.doi.org/10.1016/j.physa.2016.03.003
Reference: PHYSA 16966

To appear in: *Physica A*

Received date: 1 September 2015
Revised date: 22 January 2016

Please cite this article as: J. Li, P. Zhang, D. Song, Y. Hou, An adaptive contextual quantum language model, *Physica A* (2016), http://dx.doi.org/10.1016/j.physa.2016.03.003

An Adaptive Contextual Quantum Language Model

Jingfei Li[a], Peng Zhang[a,*], Dawei Song[a,b,*], Yuexian Hou[a]

[a] *Tianjin Key Laboratory of Cognitive Computing and Application, School of Computer Science and Technology, Tianjin University, 300350, No.135 Yaguan Road, Jinnan District, Tianjin, P.R.China*

[b] *The Computing and Communications Department, the Open University, Walton Hall, Milton Keynes, MK7 6AA. United Kingdom*

Abstract

User interactions in search system represent a rich source of implicit knowledge about the user's cognitive state and information need that continuously evolves over time. Despite of massive efforts that have been made to exploiting and incorporating this implicit knowledge in information retrieval, it is still a challenge to effectively capture the term dependencies and the user's dynamic information need (reflected by query modifications) in the context of user interaction. To tackle these issues, motivated by the recent Quantum Language Model (QLM), we develop a QLM based retrieval model for session search, which naturally incorporates the complex term dependencies occurring in user's historical queries and clicked documents with density matrices. In order to capture the dynamic information within users' search session, we propose a density matrix transformation framework and further develop an adaptive QLM ranking model. Extensive comparative experiments show the effectiveness of our session quantum language models.

Keywords: Quantum language model, density matrix transformation, session search, query change information

*Corresponding authors: Dawei Song (dwsong@tju.edu.cn, dawei.song@open.ac.uk) and Peng Zhang (pzhang@tju.edu.cn)

1. Introduction

Searching over the Internet and other online data repositories has become a preferred way for people to find relevant information and acquire useful knowledge in our daily life. To fulfil an information need, the users information seeking process is often exploratory, involving continuous interaction with information and dynamic refinement of input queries within a search session. The traditional query-response mode of search (i.e., ad-hoc retrieval) has turned out to be insufficient. Instead, modeling session search is becoming an important yet challenging task. User interactions in search sessions represent a rich source of implicit knowledge about the user's cognitive state and information need that constantly evolves over time. Effectively explicating and incorporating such knowledge in a search system would potentially improve search accuracy and user experience. Despite of its potential application values, there still exit some big challenges for this new retrieval mode. For example, modeling term dependencies and capturing the dynamic query change in the context of user interactions. In this paper, we aim at addressing these challenges by presenting two session models inspired by the quantum language model.

The traditional unigram Language Model (LM) for IR is a widely used and robust model for ad-hoc retrieval. It can also be applied in the modeling of the contextual IR which is similar to the session search. However, it does not consider the term dependency in queries which is important to disambiguate the queries. For example, the query LED is ambiguous, which can refer to the Light Emitting Diode (LED) or Latex Editor (LEd). When a user issues a query "led", the search engine may "misunderstand" the real meaning if the search engine only observe the query term "led". However, if the search engine observes other words (e.g., semi-conductor, electron, and photon) co-occurring with "led" in a document frequently, we may have high probability to predict the "led" means the Light Emitting Diode (LED).

There have been some IR models which attempted to incorporate the term dependencies, such as the n-gram (e.g., bigrams and trigrams) language models, extended VSM (Vector Space Model), and MRF (Markov Random Field) models. However, they regard the term dependencies as additional dimensions and fail to model the inter-relationship among components (e.g., single words) of term dependencies. Thus some important information may be lost. For example, traditional dependency models regard “information retrieval” as a special term, the importance of which is weighted as whole with statistics like approaches, while do not quantify the “inner dependency” (inner relationship) between “information” and “retrieval”.

Recently, a Quantum Language Model (QLM) [1] has been proposed for the ad hoc retrieval task, and can model the term dependency in a principled way. In QLM, both queries and documents are represented as a list of projectors that are corresponding to single terms or compound dependencies (i.e., consisting of the internal relations among two or more terms). Projectors are treated as the quantum elementary events, sampled from the quantum probability space, i.e. a Hilbert space. The compound dependency is a superposition event which is a special kind of projector. Over the list of projectors for a query (or a document), there exists a quantum probability distribution encapsulated in a density matrix, known as a Quantum Language Model for the query (or the document). One can utilize an EM-based training algorithm to estimate quantum language models by maximizing a likelihood function. The estimated quantum language models for queries or documents can not only model the importance of single terms with the diagonal elements of the density matrix, but also quantify the inner relationship between components of each phrases or term dependencies with the non-diagonal elements. Let us now look at the phrase “information retrieval” as an example. In traditional LM, the phrase is regarded as a variable often with the frequency as its value, while in the QLM, the phrase is represented as a 2-order density matrix. The values of diagonal elements denote the weights of each single word, and the non-diagonal elements denote the relationship between two single words. It can be observed that the matrix representation of the phrase “information retrieval” contains more information about the inner dependency of the phrase “information retrieval”. Furthermore, the ranking function for a document is the VN-divergence [1][2] between query and document QLMs.

Motivated by the sound theory of QLM, in this paper we propose a session search approach, denoted as Contextual QLM (C-QLM), which naturally incorporates the complex term dependencies occurring in user's historical queries and clicked documents. C-QLM utilizes user's historical queries, clicked documents and the pseudo-relevance documents to obtain a reliable representation of user's hidden information need as a contextual query QLM. It quantifies the internal relationship among components of the term dependencies by encapsulating the QLM into a density matrix which is trained by maximizing a likelihood function of a document representation as a sequence of projectors. Projectors can capture more compound dependency among terms. The C-QLM significantly improves the original ranked results returned by the search engine.

However, the C-QLM does not model the dynamics of users' information need. Thus it fails to capture the evolution information when users are changing their queries to find final search results. To deal with this problem, we further improve the C-QLM model and propose an Adaptive Contextual QLM (AC-QLM) to capture the query reformulation information when the user is completing a specific search task. To do this, we assume that there exists an ideal density matrix which can well represent user's real-time information need perfectly. However, it is impossible to obtain this "ideal" density matrix, we can only approach the ideal density matrix as closely as possible. To this end, we propose to transform the density matrix for contextual query QLM by incorporating query reformulation information within a search session. An adaptive algorithm is developed to approximate the density matrix transformation framework. From the experimental results, we find that the AC-QLM improved the ranking of more relevant documents, compared with the C-QLM. Although there have been some attempts to use the query change information in session search task [3][4], they do not consider term dependencies in query representation.

The rest of this paper is organized as follows. Section 2 presents a brief review of the related work. Section 3 gives a detailed introduction to the quantum language model. The QLM-based session search models are proposed in Section

4. In Section 5, we report the empirical experiments. Section 6 concludes the paper and points out some future research directions.

2. Related Work

There are two lines of related work, i.e. search personalization and quantum theory inspired information retrieval.

There have been massive approaches that utilize the user's historical interaction data to personalize search results. The typical approach is to expand the current query with contextual terms or entities extracted from historical interaction data or user's personal information repository, in order to enhance the query representation and reduce ambiguity [5][6][7]. Some researchers [8][9][10][11] construct a user profile based on user search logs to model users' long-term or short-term interests that reflect the user's hidden information need. In the ranking process, the similarity between user profile and the document was considered. For example, Dou *et al.* constructed user profiles using different personalization strategies [8]. Harvey *et al.* personalized the web search by building user profiles from topic models [9]. Li *et al.* construct user profiles with subspaces and rank documents based on the projection of a document onto the user profile subspace [10]. Few existing work has captured both term dependencies and the evolving information need in the session search task. Although Guan *et al.* [3] and Zhang *et al.* [4] have applied query information into session search task by adjusting the weights of query terms, they did not consider term dependencies in their approaches. In this paper, we integrate the query change information into a quantum language model, which considers the term dependencies in a principled way.

Following the pioneering work by van Rijsbergen on quantum IR [12], many subsequent IR methods [13][1][14] are proposed. The main inspiration is rooted on the Quantum Theory (QT) as a sound framework for manipulating vector spaces and probability. In [13], queries are represented as density operators and documents are represented as multi-dimensional objects, i.e. subspaces. Zuccon and Azzopardi proposed a Quantum Probability Ranking Principle (QPRP) [15], which captures the inter-document dependencies in the form of "quantum interference". Inspired by the photon polarization experiment underpinning the "quantum measurement", Zhao *et al.* [14] proposed a novel re-ranking approach, and Zhang *et al.* [16] proposed a query expansion model. Recently, Sordoni *et al.* [1] developed the quantum language model and gained an improved performance in ad-hoc retrieval. Despite of many QT-inspired methods that have shown acceptable retrieval effectiveness in ad hoc retrieval, there is little evidence showing their effectiveness when applied to contextual search task, e.g., session search, personalized search, etc. For this paper, we introduce the quantum language model into a novel application field in IR, which shows great potential.

3. Quantum Language Model (QLM)

As an extension of classical unigram LMs, Quantum Language Model (QLM) [1] uses the projectors to represent a single term or compound dependency. The density matrix, corresponding to a QLM, represents the quantum probability distribution of quantum events, i.e. projectors.

3.1. The Quantum Probability

3.1.1. Quantum Elementary Events

The quantum probability space is naturally encapsulated in a Hilbert space, noted as H^n, while for the convenience of computation, in this paper we limit it to finite real space[1], denoted as R^n. A unit column vector $\vec{u} \in R^n, \|u\|_2 = 1$,

[1](i) In the field of quantum IR, many existing work (e.g., [1, 17, 18, 19]) only considered the real field for the sake of the conceptual interpretability and computational feasibility. Therefore, we follow their settings in our manuscript. (ii) Currently, we cannot find the intuitive meaning of the complex field in the information retrieval task. (iii) Although the quantum IR is motivated by the quantum theory, in practice, we have to do some simplifications for our specific task. In this paper, we need to train a density matrix with the $R\rho R$ approach, but we do not know how to include a complex field so far.

and its transpose $\vec{u}^T$ are respectively written as a ket $|u\rangle$ and a bra $\langle u|$ (Dirac's notation). Each $|u\rangle$ is associated with a dyad $|u\rangle\langle u| \in R^{n\times n}$, which is the projector that projects any vector onto the direction of $|u\rangle$[2]. These dyads are the elementary events of the quantum probability space [1][20]. $|e_i\rangle$ are used to represent elements of the standard basis in R^n, i.e. $|e_i\rangle = (\xi_{1j}, ..., \xi_{nj})$, where $\xi_{ij} = 1$, if and only if $i = j$, otherwise $\xi_{ij} = 0$. For example, if $n = 2$, the quantum elementary events corresponding to $|e_1\rangle = (1, 0)^T$ and $|f_1\rangle = (1/\sqrt{2}, 1/\sqrt{2})^T$, are respectively represented by the dyads:

$$|e_1\rangle\langle e_1| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, |f_1\rangle\langle f_1| = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \tag{1}$$

This is different from the conventional probability space, which consists of n points, each point corresponding to an elementary event in the space, and a probability distribution over the n points is specified by a real number vector $\alpha = (\alpha_1, ..., \alpha_n)$, where $\alpha_i > 0$ and $\sum_i \alpha_i = 1$. In the quantum probability space, there are infinite number of dyads even if the dimension n is finite. Specifically, if we estimate a traditional probability distribution for the language model given a finite vocabulary, the probability of a term in the vocabulary can be estimated, while probability of a phrase that is not in the vocabulary will not be estimated. The quantum probability can solve this problem. Even though the training set does not contain the phrase, the probability of this phrase can also be computed by the Gleason's theory [12].

3.1.2. Density Matrix

Density matrices (*n-dimensional* matrices denoted as S^n) are the generalization of the conventional finite probability distributions. A density matrix can be defined as mixture of dyads $\rho = \sum_i \phi_i |u_i\rangle\langle u_i|$, where $\phi_i \geq 0$ and

[2]The projection for any n-dimensional vector $\vec{v}$ onto the direction $|u\rangle$, can be computed by following equation: $p_v = |u\rangle\langle u|\vec{v}$, e.g. $\vec{v} = (2, 3)^T, |u\rangle = |e_1\rangle = (1, 0)^T, p_v = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}(2, 3)^T = (2, 0)^T$

$\sum_i \phi_i = 1$. Each dyad is symmetric, positive definite and has trace one, i.e. $tr(|u_i\rangle \langle u_i|) = 1$[3]. Therefore, density matrices are also symmetric, positive definite and have trace one. The number of components in the mixture is arbitrary, while any n-dimensional density matrix can be decomposed into a mixture of n orthogonal eigendyads corresponding to the eigenvectors of the density matrices, i.e. $\rho = \sum_{i=1}^{n} \theta_i |v_i\rangle \langle v_i|$, where $|v_i\rangle$ are the eigenvectors and θ_i are the corresponding eigenvalues. By the Gleason's Theorem [12], a density matrix ρ assigns quantum probability $\mu_\rho(|u\rangle \langle u|) = tr(\rho |u\rangle \langle u|)$ to each unit vector $|u\rangle$ and its associate dyad $|u\rangle \langle u|$. The probability can also be rewritten as $\mu_\rho(|u\rangle \langle u|) = \langle u| \rho |u\rangle$.

Let us consider the aforementioned conventional probability distribution α. The n-dimensional matrix $diag(\alpha)$ is a density matrix. It is worth noting that $diag(\alpha) = \sum_{i=1}^{n} \alpha_i |e_i\rangle \langle e_i|$, where $|e_i\rangle$ are the standard basis vectors. Thus the conventional probability distributions are special cases of the density matrices, where the eigensystem is restricted to the identity matrix [20]. Take the traditional unigram language model as an example, we have a vocabulary with two terms $\mathcal{V} = \{a, b\}$, and $\alpha = (0.2, 0.8)$ is a unigram language model over the vocabulary, then the corresponding density matrix is defined as follows:

$$\rho_\alpha = diag(\alpha) = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.8 \end{pmatrix} \tag{2}$$

where each term corresponds to a projector and the term projectors are orthogonal each other, i.e. the quantum elementary events associated with the terms are disjoint. Given the density matrix, the probability of the term t is gained by $tr(\rho_\alpha |e_t\rangle \langle e_t|)$, $t \in V$ e.g., $P(a) = tr(\rho_\alpha |e_a\rangle \langle e_a|) = 0.2$.

A density matrix ρ can be depicted as an ellipse which is an affine transformation of the unit ball: $\{\rho |u\rangle : |u\rangle \in R^n\}$ and the quantum probability in direction $|u\rangle$ can be depicted as an "eight-like" shape: $\{tr(\rho |u\rangle \langle u|) |u\rangle : |u\rangle \in R^n\}$ (see Figure 1) [1][20]. As quantum probability can be defined on an infinite number of

[3] $tr(\bullet)$ is the trace of a matrix, i.e. the sum of the main diagonal elements of a matrix.

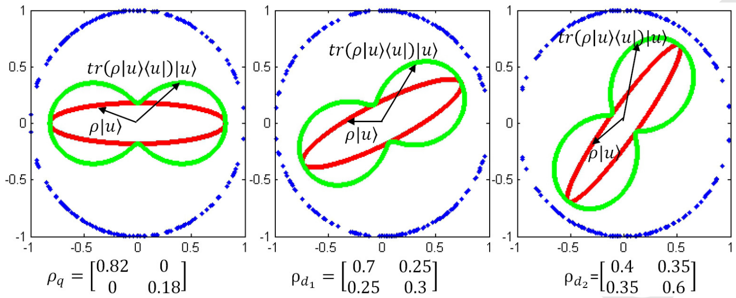


Figure 1: The outer circles are the unit spheres, the red ellipses depict the set of points $\{\rho\,|u\rangle : |u\rangle \in R^2\}$, the "eight-like" shape is the quantum probability in direction $|u\rangle$. The eigenvectors determine the direction of the axis of the ellipses and the corresponding eigenvalues define the length of the corresponding axis. $relScore(d_1) = -\triangle_{VN}(\rho_q||\rho_{d1}) = -0.6536$ and $relScore(d_2) = -\triangle_{VN}(\rho_q||\rho_{d2}) = -1.2332$, $relScore(d_1) > relScore(d_2)$.

eigensystems [1], the ellipses and "eight-like" shapes associated with the density matrices can be "rotated" arbitrarily. Intuitively, if two ellipses rotated towards each other, the corresponding density matrices will have relatively smaller divergence. Utilizing this feature, one can estimate proper density matrices to build a more reliable representation of documents and queries capturing more complex semantic information than single terms. More details will be given in the next subsection.

3.2. QLM for Information Retrieval

Before introducing the QLM into retrieval, we first review the Kullback-Leibler (KL) divergence retrieval model [21][22], known as a probability similarity model. Assuming that the query and document[4] are sampled from a query language model θ_Q and a document language model θ_D respectively, then the KL-divergence of two models is used to measure how distant they are, and the distance (indeed, negative distance) is used as a score to rank documents. Based

[4]In the unigram language model, both the query and document are represented by a sequence of i.i.d term events, e.g. a document is represented by $W_D = w_i : i = 1, ..., N$, where N is the document length. Each w_i belongs to a sample space V, corresponding to the vocabulary of size n [1].

on this idea, traditional KL models can be generalized to the quantum language model. In the QLM based retrieval model, both the query and document are represented by a sequence of quantum elementary events rather than a sequence of terms as events as in traditional KL model. The sequences for a query P_Q and for a document P_D are assumed to be sampled from a query QLM and a document QLM, associated with the density matrices ρ_Q and ρ_D respectively. Given a representation of the query and document, one can use the Maximum Likelihood (ML) to estimate the ρ_Q and ρ_D (to be detailed in subsection 3.3), then use the negative Von-Neumann (VN) divergence as the scoring function to rank the documents [1][2]:

$$\begin{aligned} -\triangle_{VN}(\rho_Q||\rho_D) &= -tr(\rho_Q(log(\rho_Q) - log(\rho_D))) \\ &\stackrel{rank}{=\!=} tr(\rho_Q log(\rho_D)) \end{aligned} \tag{3}$$

Figure 1 illustrates three density matrices, corresponding to a query (ρ_q) and two documents (ρ_{d_1}, ρ_{d_2}) respectively, the relevance score (negative VN-divergence) of d_1 is larger than that of d_2, indicating d_1 is more relevant than d_2 to the query. This retrieval framework is applied to our session search models. More specifically, we train a contextual QLM for the current query using historical queries, clicked documents (in the same session) and top k returned documents, then rank the documents considering the VN-divergence between contextual query QLM and document QLM. More details will be presented in Section 4.

3.3. Estimating QLM for a Document

Here, we introduce a method for estimating QLM for a document. A quantum language model assigns quantum probabilities to arbitrary single terms or compound dependencies (2 or more terms in an unordered fixed window), and the parameters are encapsulated in a $n \times n$ density matrix $\rho \in S^n$, where n is the size of the vocabulary $\mathcal{V}$, S^n is the set of n-dimensional matrices (symmetric, positive definite and have trace one). In QLM, a document is considered as a sequence of quantum events associated to a density matrix ρ:

$$\mathcal{P}_{\mathcal{D}} = \{\Pi_i : i = 1, ..., M\} \tag{4}$$

where each Π_i is a quantum elementary event $|u\rangle \langle u|$ representing a single term or a compound dependency. The number of quantum elementary events M can be different from N (i.e. the number of terms in the document). Note that, the sequence of projectors is an independently sampling sequence rather than a temporal sequence. The relative order of projectors corresponds to the original position of the terms in the document. Sordoni *et al.* [1] defined a way of mapping single terms and term dependencies to quantum elementary events, formally, as $m : \mathbb{P}(\mathcal{V}) \to \mathcal{L}(R^n)$. $\mathbb{P}(\mathcal{V})$ is a power set of the vocabulary and includes all possible single terms or compound dependencies and $\mathcal{L}(R^n)$ is the corresponding set of quantum elementary events, i.e. dyads on R^n. The mapping from single terms to a dyad is:

$$m(w) = |e_w\rangle \langle e_w| \tag{5}$$

where $w \in \mathcal{V}$, $|e_w\rangle$ is the standard basis vector associated to a term w. It is worth noting that if we only observe the single terms, i.e. $M = N$, $\mathcal{P}_{\mathcal{D}}$ is essentially reduced to the traditional representation in the bag-of-words models.

In this paper, the compound dependency means the relationship linking two or more terms, which is represented by a subset of the vocabulary, i.e. $k = \{w_1, ..., w_K\}$. The mapping from a compound dependency to a dyad is:

$$m(k) = m(\{w_1, ..., w_K\}) = |k\rangle \langle k| \,, where \, |k\rangle = \sum_{i=1}^{K} \delta_i \, |e_{w_i}\rangle \tag{6}$$

where the coefficients $\delta \in R$ must meet the constraint of $\sum_{i=1}^{K} \delta_i^2 = 1$ to guarantee that $|k\rangle$ is a unit vector. The larger coefficients add more weight to the corresponding $|e_w\rangle \langle e_w|$. The well-defined dyad $|k\rangle \langle k|$ is a superposition event, which allows for a representation of relationships within a group of terms by creating a new quantum event in the same n-dimensional space [1].

After defining the mapping $m(\bullet)$, we can represent a document with a sequence of quantum elementary events. Note that both the single term projectors and the compound dependency projectors are added to the representation. We will describe how to select the compound dependency in Section 4.2. A query can be seen as a special document, so that we can represent a query in the same way as the document representation.

Given the observed quantum elementary events $\mathcal{P}_{\mathcal{D}} = \{\Pi_1, ..., \Pi_M\}$ for document $\mathcal{D}$, Sordoni *et al.*[1] utilized the ML method to train the document QLM $\rho_{\mathcal{D}}$ (a density matrix) by maximizing the product of quantum probability for each quantum elementary event, formalized as Equation 7. We can observe that the equation corresponds to the likelihood function in language model.

$$\mathcal{L}_{\mathcal{P}_{\mathcal{D}}}(\rho_{\mathcal{D}}) = \prod_{i=1}^{M} tr(\rho_{\mathcal{D}}\Pi_i) \tag{7}$$

The estimation of the document QLM $\rho_{\mathcal{D}}$ can be transformed to the following maximization problem, in which the objective function is the logarithm of the likelihood:

$$\begin{aligned} \max_{\rho_{\mathcal{D}}} \quad & \mathcal{L}_{\mathcal{P}_{\mathcal{D}}} \\ \textit{subject to} \quad & \rho_{\mathcal{D}} \in S^n \end{aligned} \tag{8}$$

An optimization method for estimating density matrices, denoted as "$R\rho R$ " algorithm [23][1], is used to solve the optimization problem. The "$R\rho R$" algorithm updates the estimate $\rho_D(k+1)$ according to $\rho_D(k)$ iteratively (starting from an initial density matrix $\rho_D(0)$ until the training objective reaches a convergence or the number of iterations is beyond a preset value. To conduct the iteration process, the following operator is introduced:

$$R(\rho) = \sum_{i=1}^{M} \frac{1}{tr(\rho\Pi_i)}\Pi_i \tag{9}$$

Intuitively, this operator conducts quantum measurement for each quantum events. Based on this operator, the iteration process is formalized as Equation

10, which leads to a new quantum state or quantum probability distribution, i.e., a new density matrix:

$$\widehat{\rho}_D(k+1) = \frac{1}{Z} R(\widehat{\rho}_D(k))\widehat{\rho}_D(k) R(\widehat{\rho}_D(k)) \tag{10}$$

where, $Z = tr(R(\rho_D(k))\rho_D(k)R(\rho_D(k)))$ is a normalization factor in order to ensure that $\rho_D(k+1)$ satisfies the constraint of unitary trace [23][1]. This is a nonlinear iteration process that may suffer an overshooting problem, just like the gradient descent algorithm with too big step size. To guarantee convergence, if the training objective is decreased at the $k+1$ step, we apply the following damped update[5]:

$$\overline{\rho}_D(k+1) = (1-\lambda)\widehat{\rho}_D(k) + \lambda\widehat{\rho}_D(k+1) \tag{11}$$

where $\lambda \in [0, 1)$ determines the degree of the damping and is selected by a linear search in order to ensure the maximum increase of the objective function.

Note that $R\rho R$ algorithm is not the only training method for maximizing the likelihood function. One possible alternative choice is the $R\rho$ algorithm, which updates the iterative density matrix as Equation 12:

$$\widehat{\rho}_D(k+1) = (1-\xi)\widehat{\rho}_D(k) + \xi \frac{\widehat{\rho}_{D(k)} R(\widehat{\rho}_{D(k)}) + R(\widehat{\rho}_{D(k)})\widehat{\rho}_{D(k)}}{2} \tag{12}$$

where, if ξ is sufficiently small, this updating function can guarantee that the target function can reach the global convergence. However, it will be extremely time consuming to achieve the global convergence, thus we did not choose this function for the sake of efficiency.

In this paper, each document QLM is smoothed by the Dirichlet method. If $\widehat{\rho}_{\mathcal{D}}$ and $\widehat{\rho}_{\mathcal{C}}$ are respectively the document QLM and collection QLM obtained by ML, the smoothed version of document QLM is defined as Equation 13:

$$\rho_{\mathcal{D}} = (1-\alpha_{\mathcal{D}})\widehat{\rho}_{\mathcal{D}} + \alpha_{\mathcal{D}}\widehat{\rho}_{\mathcal{C}} \tag{13}$$

[5]As a density matrix S^n is convex [24], the combination of different density matrix is also a density matrix.

where $\alpha_{\mathcal{D}} = \frac{\mu}{\mu+M}$ is the well-known form of the parameter for Dirichlet smoothing [25], μ is set to the default value of 2500, and M is the number of projectors in the representation of the collection.

4. Session Quantum Language Models

4.1. Background

Now, we briefly describe the session search task. A search session can be seen as a user's single search task or goal, which consists of a sequence of queries issued by one particular user and records the user's interaction information with search engines (click-through data, dwell time on a web page, etc.) [26][27][28]. The first query of a session is the start point of a search task, and correspondingly the last query is the end point of the task (see Equation 14).

$$S = << q_1, C_1 >, ..., < q_{n-1}, C_{n-1} >, < q_n, R_o >> \tag{14}$$

where q_i is the i_{th} query in the session, q_n is the current query, R_o represents the original ranked results of the current query returned by the search engine, $q_1, ..., q_{n-1}$ are the historical queries, and C_i corresponds to a set of clicked documents under a query q_i. In this paper, we propose to re-rank the original results using the proposed session QLMs trained with the top K returned documents of the current query and historical information in the same session.

In rest of this section, we first introduce a contextual query QLM for session search, followed by an adaptive session QLM.

4.2. Contextual Query Quantum Language Model (CQ-QLM)

We propose to build a Contextual Query QLM for current query q_n through the combination of q_n's QLM and the history QLM, formalized as Equation 15.

$$\rho_{CQ} = \xi \times \rho_Q + (1 - \xi) \times \rho_H \tag{15}$$

where $\xi \in [0, 1]$ is a combination parameter, which determines the extent of impact of historical interaction information on the contextual query QLM, ρ_Q

is the query QLM estimated using the current query and top k returned documents. ρ_H is the history QLM obtained by combining all of the historical QLMs for different interaction units (i.e., corresponding to different $< q_i, C_i >$).

To train the contextual query QLM and document QLM, we define a vocabulary $\mathcal{V}_S = \{w_1, ..., w_N\}$ for each session, which only includes the words of the current query and previous queries in the same session, where N is the size of the vocabulary. In this way, the computation cost is reduced greatly compared with using a larger vocabulary, e.g., the collection vocabulary. To detect term dependencies in documents and query, we need first build a candidate se. Specifically, we define a word set (with 2 or 3 single words) as a term dependency only if all components of the set co-occur in one query (current query or previous queries). For example, for a session s with two queries $q_1 = ab$ and $q_2 = bc$, the vocabulary $\mathcal{V}_s$ is $\{a, b, c\}$ and the candidate set of term dependency is $\{ab, bc\}$. Note that ac is not detected as a term dependency. All single words in the vocabulary and the candidate term dependencies constitute a candidate set $\mathbb{P}(\mathcal{V}_S)$, which is used in the representation of query and document as sequences of projectors. With respect to the earlier example, $\mathbb{P}(\mathcal{V}_s) = \{a, b, c, ab, bc\}$. The rest of this section will describe the detailed process of how to obtain the contextual query QLM based on the candidate set $\mathbb{P}(\mathcal{V}_S)$.

4.2.1. Training the QLM for each Interaction Unit

The QLM for a historical interaction Unit $U_i =< q_i, C_i >$, denoted as ρ_{U_i}, is a QLM estimated from the i_{th} historical query and its clicked documents. Before estimating a Unit QLM, one should represent the query and its clicked documents with a sequence of quantum elementary events (terms and compound dependency patterns). We apply Algorithm 1 to get the representation as a sequence of projectors, denoted as $\mathcal{P}_{u_i}$.

In Algorithm 1, ϵ is a single term or a compound dependency. If ϵ is a single term, $\#(\epsilon, d)$ is the occurrence frequency of the term in document d. If ϵ is a compound dependency, the $\#(\epsilon, d)$ is a function that returns how many times the dependency ϵ is observed in d. In this paper, we choose to detect a

Algorithm 1 Represent a historical query and its clicked documents given $\mathcal{V}_S, q_i, C_i$

Require: $\mathcal{V}_S, q_i, C_i$

$\mathcal{P}_U \Leftarrow \phi$;
for each $d \in \{q_i, C_i\}$ **do**
  // d is a document, q_i is seen as a special document
  for each $\epsilon \in \mathbb{P}(\mathcal{V}_S)$ **do**
    //ϵ is a single term or a compound dependency
    for $i = 1$; $i \leq \#(\epsilon, d)$; $i++$ **do**
      // add the projector to the sequence
      $\mathcal{P}_U \Leftarrow \mathcal{P}_U \bigoplus m(\epsilon)$;
    end for
  end for
end for
return $\mathcal{P}_U$

dependency if the component terms co-occur in any order in a window of fixed length L. Given the representation of the interaction unit, one can apply the training method described in [1] to get an estimated unit QLM ρ_{U_i}. Note that the smoothing step is needed to avoid the zero-probability problem.

Noted that, the QLM for the current query q_n is estimated in the similar way using the top k returned documents in the original returned results, which is different from the training of historical query QLM using clicked documents.

4.2.2. Combining the Interaction Unit QLMs

Intuitively, the similar historical queries will have more impact on current query [29]. In this paper, we assign different weights to the historical interaction unit QLMs when combining them according to the similarity with current query. This is formalized as Equation 16.

$$\rho_H = \sum_{i=1}^{n-1} \gamma_i \times \rho_{U_i} \tag{16}$$

where $n-1$ is the number of historical unit QLMs; γ_i is the similarity between q_i and current query q_n, and is normalized to one, i.e. $\sum_{i=1}^{n-1} \gamma_i = 1$. To compute the similarity, we represent each query as a TF-IDF vector v_i derived from the concatenation of all its result documents as did in [29] did.

$$v_i[w] = C(w, \mathcal{D}_i) \times log \frac{N+1}{DF(w) + 0.5} \tag{17}$$

where $C(w, \mathcal{D}_i)$ is the term frequency of w in the concatenated documents corresponding to q_i. N is the number of document in the document collection, $DF(w)$ is w's document frequency, and γ_i is the Cosine similarity between vectors v_i and v_n.

$$\gamma_i = cosine(v_i, v_n) = \frac{v_i \cdot v_n}{|v_i||v_n|} \tag{18}$$

4.2.3. More Explanations

Intrinsically, we combine the current query's quantum probability distribution and the historical query's quantum probability distribution, in order to obtain a mixing quantum probability distribution. In this way, the query will be embedded into a higher dimensional representation space than the traditional probability distribution. The advantage is dependent on the strong representation ability of quantum language model for information need. The combination weights of different contextual quantum probability distribution are determined by some classical function. In this way, the more relevant topic will be considered in the contextual modeling of users' information need.

4.3. Adaptive Contextual Quantum Language Model for Query (AC-QLM)

Using the proposed method in Section 4.2, we can train an contextual query QLM, which well incorporates user's information need for the current session, by encapsulating his/her interaction information and the pseudo relevance feedback information into one density matrix. However, the contextual query QLM also introduces some noise into the representation of the current query and does not capture the user's evolving information need. To address this problem, we propose to transform the density matrix for the static contextual quantum language model by incorporating query change information over the search session. In this way, we can model the dynamic evolution of the user's information need.

4.3.1. Density Matrix Transformation Framework

We hypothise that there exists an ideal density matrix which can perfectly model user's dynamic information need for current query. We should transform the density matrix for the static contextual quantum language model to approach the ideal density matrix. The density matrix transformation framework is formalized as follows

$$\rho_{CQ} \xrightarrow[and\ rotating]{scaling,\ shifting} \rho_{ideal} \tag{19}$$

where, we assume that the ideal density matrix for current information need is obtained by transforming the estimated contextual density matrix with scaling, shifting and rotation operations, etc.

Any density matrix can be decomposed into the form as shown in Equation 20:

$$\rho = \sum_i \lambda_i \left| r_i \right\rangle \left\langle r_i \right| \tag{20}$$

where λ_i is an eigenvalue and $|r_i\rangle$ is the corresponding eigenvector. Each eigenvector can be interpreted as a dimension of current information need and the corresponding eigenvalue is the quantum probability of the specific projector spanned by the eigenvector (also known as information dimension). The ideal density matrix may be obtained by adjusting the direction of each eigenvector and the quantum probability of each projector. Take a 2-order density matrix as an example (Figure 2), it can be geometrically interpreted as an ellipse whose axes correspond to two eigenvectors and the length of each axis corresponds to the corresponding eigenvalue. Intuitively, the ideal matrix corresponds to an ellipse in a specific position, with a specific shape and at a specific angle (see the example in Figure 2). The estimated density matrix for contextual query QLM can be transformed to the ideal density matrix by scaling the quantum probability of corresponding axis and rotating the directions of eigenvectors.

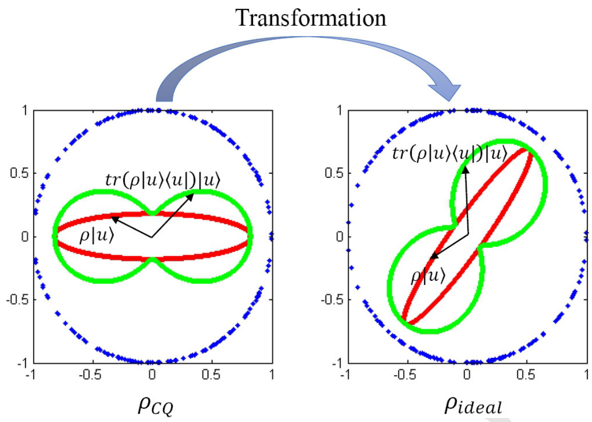


Figure 2: Transformation for a 2-order density matrix spanned by a 2-words query.

In this paper, we use the "ideal matrix" to represent the ideal case when we can ideally capture and represent user's current information need. We also assume that there exists some transformation from an initially trained density matrix to the ideal matrix. However, the "ideal matrix" only exists in theory, but in practice we can hardly obtain it. This is because that although the user's information need can be to some extent reflected by the issued query terms, these terms cannot perfectly represent user's information need. Therefore, instead of trying to obtain such an "ideal matrix", we proposed an approximation transformation approach to re-train a density matrix with query change information based on the initially trained one. Certainly, it is interesting to find a more effective and efficient method to train a better density matrix.

4.3.2. Query Change Information

Query change is considered as a kind of important implicit feedback information [3][4]. Let us consider a typical search scenario: after one issues a query into the search engine, it will return a list of results. Then, the user judges the relevance of returned results and determines which results to click. If these results cannot satisfy his/her information need, he/she may modify the query constantly, until finding results he/she is satisfied with. This process provides a clue for us to adjust the query QLM using historical query change information. For the current query in a session, it can be decomposed into three parts, i.e. the *common part*, *added part* and *removed part* compared with the previous query. For example, $q_n = abd$, $q_{n-1} = abc$, where ab is the common part, d is the added part, c is the removed part. Intuitively, the common part reflects the user's search theme for the current session, the added part and removed part reflect a change of the user's information need and knowledge state about the search task. We propose to adjust the QLM for current query, so that the QLM will assign relatively higher probability to the quantum elementary events corresponding to common part and added part. Meanwhile it assigns relatively lower probability to the events corresponding to the removed part. For instance: given an estimated QLM for a query $\rho_{example}$ and an quantum elementary event for a term ϵ_{term} (See Equation 21). The quantum probability that $\rho_{example}$ assigns to ϵ_{term} is $\mu_{\rho_{example}}(\epsilon_{term}) = tr(\rho_{example}\epsilon_{term}) = 1/3$. If we adjust the QLM from $\rho_{example}$ to $\rho^{'}_{example}$ as Equation 22, the corresponding quantum probability increases to $\mu_{\rho^{'}_{example}}(\epsilon_{term}) = 1/2$. The detailed adjusting method will be described in the next subsections.

$$\rho_{example} = \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{pmatrix}, \epsilon_{term} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{21}$$

$$\rho^{'}_{example} = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{pmatrix} \tag{22}$$

4.3.3. Obtaining the Common Part and Added Part

Here we obtain the common part, added part and removed part compared between the current query and immediate previous query. When we increase probability of common part and added part, the probability of removed part will be relatively reduced naturally. The common part is the intersection of current query and previous query, while added part (or removed part) are the difference between the current query (or previous query) and common part, See Equation 23

$$\begin{aligned} CPart &= wordset(q_n) \cap wordset(q_{n-1}) \\ APart &= wordset(q_n) - CPart \\ RPart &= wordset(q_{n-1}) - CPart \end{aligned} \tag{23}$$

The common part, added part and removed part are three word sets. We propose to map each single word and dependency in the word sets into projectors using Equations 5 and 6, and then form the corresponding sequences of projectors, i.e. $\mathcal{P}(CPart)$, $\mathcal{P}(APart)$ and $\mathcal{P}(RPart)$. Essentially, each projector is a special density matrix, which will be used to adjust the contextual QLM as detailed in the next subsection.

4.3.4. Training an Adaptive Contextual Query QLM

We propose an adaptive contextual query QLM (AC-QLM) by training a new dynamic density matrix with query change projectors. Specifically, we utilize the pre-estimated density matrix for the static contextual query QLM as the initial density matrix for training. Moreover, we develop a new query change likelihood function, which rewards the added projectors and common projectors, and penalizes the removed projectors. In this way, we can change the quantum probability of common projectors, added projectors and removed projectors. We assume that, both common part and added part are important to represent user's information need, but the degree of importance is different. The common part is seen as the theme terms for current session. If these theme terms occur in the previous clicked documents many times, this may indicate that the user hopes to find more different documents containing these terms. In this case, we should give a penalty item for the theme terms to avoid retrieving too many repeated documents. The added part reflects the change direction of user's information need, so we should assign more importance to it than the common part. Based on the static contextual query quantum language model

ρ_{CQ}, we re-train a new quantum language model as follows.

$$\rho_{adaptive}(0) = \rho_{CQ} \tag{24}$$

$$\begin{aligned}\mathcal{L}(\rho_{adaptive}) = &\prod_{\Pi_\epsilon \in \mathcal{P}(CPart)} tr(\rho_{adaptive} \cdot \Pi_\epsilon)^{\frac{1}{\#(\epsilon, rDocs)+0.5}} \\ &\prod_{\Pi_\epsilon \in \mathcal{P}(APart)} tr(\rho_{adaptive} \cdot \Pi_\epsilon)^{log(\frac{N_c}{DF(\epsilon)+0.5})} \\ &\prod_{\Pi_\epsilon \in \mathcal{P}(RPart)} tr(\rho_{adaptive} \cdot \frac{1}{tr(I-\Pi_\epsilon)}(I-\Pi_\epsilon))^{log(\frac{N_c}{DF(\epsilon)+0.5})}\end{aligned} \tag{25}$$

where ϵ represents a single word or a term dependency, Π_ϵ represents the corresponding projector, $\#(\epsilon, cDocs)$ (or $\#(\epsilon, rDocs)$) is the number of ϵ occurs (or co-occur) in a fixed window (length L) in the clicked documents of previous query (or the top-most 10 returned documents for the current query), N_c is the document number in collection, and $DF(\epsilon)$ is the document frequency for a single word or the co-occurrence document frequency for a compound dependency. I is the identity matrix and $\frac{1}{tr(I-\Pi_\epsilon)}(I-\Pi_\epsilon)$ is still a projector which gives a penalty to the removed projectors. After formalizing the new likelihood function, we can still use the $R\rho R$ algorithm to estimate the adaptive density matrix $\rho_{adaptive}$.

4.3.5. More Explanations

We re-train the contextual quantum language model with the query change information in order to capture the dynamics of users' information need. Specifically, the contextual query quantum language model can be regarded as the mixing of quantum probability spanned from the short-term query history, which statically reflects users' search interests. The adaptive quantum language model adjusts the static contextual model with the query change signals in a principled way, which leads to the quantum language model move to the a right direction. Thus there is great potential to further improve the search performance.

4.4. Re-Ranking the Original Ranked Results

After estimating a quantum language model for the current query and each document in original ranked results, we can apply Equation 3 to obtain an relevance score, denoted as $vnScore$. To re-rank the original ranked results, we assign a re-ranking score, denoted as $rScore$ for each document, by combining the relevance score $vnScore$ and the original rank score, denoted as $oScore$, as Equation 26.

$$rScore(d) = (1 - \beta) \times oScore(d) + \beta \times vnScore(d) \tag{26}$$

where d is a document, $oScore(d) = 1/log_2(1+rank(d))$, $rank(d)$ is the original rank of d in the original ranked results, and $\beta \in (0, 1]$ is the linear combination parameter.

5. Experiments

5.1. Data Set

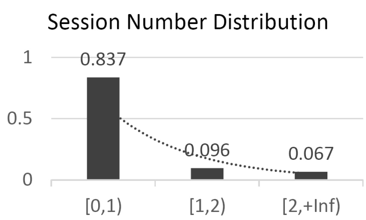
Our experiments are conducted on a subset of query logs from a prominent search engine over a period of 10 days from 1^{st} July, 2012 to 10^{st} July, 2012. The subset (See Table (b) in Figure 3) contains about 116154 queries issued by 1166 users identified by anonymous user IDs. We segmented them into 17647 sessions, according to the timeout threshold of 30 minutes [26][27]. On average, each session includes 6.58 queries. The last query of each session is seen as the current query, of which the original result list will be re-ranked using the re-ranking algorithms. The previous queries in the session are considered as the historical queries whose click-through data is used in re-ranking process. Note that the number of the original returned results is not certain (may be 10, 20, 30 or more), since one user usually examines more than 1 SERPs (Search Engine Result Pages) and the search engine records all of the results in the SERPs viewed by the user.

The statistics about the test data are illustrated in Figure 3. Figure (3-a) shows that in our test data, about 83.7% test sessions have a click entropy[6] less than 1; 9.6% sessions' click entropy are between 1 and 2; and only 6.7% sessions have click entropy more than 2. We also compute the average RankScoring (an evaluation metric for ranked results, see Equation 27) for original ranked result lists of the current queries over different click entropy intervals. Table (b) in Figure 3 illustrates that the retrieval performance (in terms of average RankScoring) over click entropy interval $[0, 1)$ gains a good result (the possible maximum RankScoring is 100), leaving little potential to be improved, and that over interval $[1, 2)$ also achieves a relative good result with RankScoring value 75.41. The current queries with click entropy more than 2 have the worst retrieval effectiveness, which shows that the retrieval effectiveness for this portion of queries have the most potential to be improved. The observation is consistent with those reported in the literature [8]. For this reason, we only test the algorithms over the sessions whose click entropy are larger than 2. Although this portion of sessions only accounts for a small proportion of the total sessions, improving the retrieval performance of them is important in terms of the large number of searches in real web search environments.

We maintain a large scale document collection including 1895850 web pages that are downloaded from the Internet, which are indexed with Indri[7], a widely used language model toolkit in IR. Each document is preprocessed by removing stop words and stemming with the Porter stemmer [30]. The full texts corresponding to the clicked documents in historical interaction units (i.e., for historical queries) and the original returned documents for current query can be found in this collection.

[6]Click Entropy is a concept proposed in [8], which is a direct indication of query click variation. Smaller click entropy of a query indicates that clicks between users are focused on a smaller number of web pages. The larger click entropy of a query, the larger variation of the clicked documents among different users. A session's click entropy is the click entropy of the current query.

[7]http://www.lemurproject.org



(a)

#Queries	116154
#Sessions	17647
#Average	6.58

(b)

ClickEntropy	RankScoring
[0,1)	97.09
[1,2)	75.41
[2,+Inf)	39.77

(c)

Figure 3: Statistics about the test data. Figure (a) is the distribution of session number over different click entropy intervals. Table (b) shows the basic information of the dataset; Table (c) reports the RankScoring distribution of the original ranked results over different click entropy intervals;

5.2. Evaluation Metrics

We use two evaluation metrics to measure the quality of ranked results for test sessions (the current queries). They are RankScoring [8] and Re-Ranking Gain [9][31]. In this paper, we utilize the "SAT click" criteria (i. the user dwells on the clicked documents for at least 30 seconds; ii. the click is the last click in current query session) [32] to judge the relevance of a document.

5.2.1. RankScoring

RankScoring [8], denoted as R_q, evaluates the quality of a ranked list for a query, $R_{average}$ is the average RankScoring over the test queries.

$$R_q = \sum_j \frac{\delta(q,j)}{2^{(j-1)/(\alpha-1)}}; R_{average} = 100\frac{\sum_q R_q}{\sum_q R_q^{Max}} \tag{27}$$

where j is the rank of a result in the list, $\delta(q,j)$ is 1 if result j is relevant to user's information need and 0 otherwise, and α is set to 5, which follows the setting in [8]. R_q^{Max} is the obtained maximum possible rank scoring for a query when all relevant URLs appearing at the top of the ranked list. The larger rank scoring indicates the better quality of the ranked result list.

5.2.2. Re-Ranking Gain (R-Gain)

After the re-ranking process, some relevant documents' ranks are improved, while some drop and others stay the same. Intuitively, the more number of relevant documents are improved and the less number of relevant documents are hurt, the better quality of a re-ranked results list will be. Accordingly, we use R-Gain to evaluate how stably the re-ranking algorithm improves the ranking performance over a baseline across all test queries [9].

$$R\text{-}Gain = \frac{\#Inc - \#Dec}{\#Inc + \#Dec} \tag{28}$$

where $\#Inc$ and $\#Dec$ represent the number of relevant documents whose rank are increased and decreased, respectively, compared with the original result list. A higher positive R-Gain value indicates a better overall robustness of a re-ranking algorithm in terms of improving performance over the baseline.

5.3. Experimental Setup

To verify the effectiveness of our session QLM models, we conduct extensive comparative experiments. Six models are evaluated, including the Original ranking model, basic QLM, RM3, C-LM, C-QLM and AC-QLM. They are summarized as follows:

1. **Original**, is the original ranking model of the search engine, which returns the original ranked results.

2. **QLM**, is the basic Quantum Language Model, the density matrix of which is trained only using current query. We use this model as a baseline in order to investigate if our contextual quantum language model can outperform the basic quantum language model.

3. **RM3**, is the combination of original query language model and expansion model, which is a state-of-the-art pseudo-relevance feedback model [33]. We use this model as a baseline in order to know if the click based IR model can outperform the pseudo-relevance feedback IR model.

4. **C-LM**, is a baseline re-ranking model (refers to the models in [29]), which is a traditional language model estimated using the top 5 documents and

the historical unit LMs in the same session. LMs are smoothed by Dirichlet smoothing method and μ is set as the default value of 2500 [25]. This model is seen as a non-quantum counterpart of C-QLM.

5. **C-QLM**, refers to the proposed contextual query QLM (see Section 4.2).
6. **AC-QLM**, refers to the proposed Adaptive Contextual query QLM.

The above models have following settings:

(1) For QLM, C-QLM and AC-QLM, we tested two different weighting schemes for mapping a term dependency into a projector (see Equation 6) as [1] did, the first scheme assigns simple uniform superposition weights to each dependency components, i.e. $\delta_i = 1/\sqrt{K}$, denoted as UNI; another scheme assigns more reasonable weights according to the importance of each component in the term dependency, i.e. $\delta_i = \sqrt{idf_{w_i}/\sum_j idf_{w_j}}$, denoted as IDF.

(2) For QLM, C-QLM and AC-QLM, we tested different fixed-window sizes (i.e. different $L \in \{1, 2, 4, 6, 8\}$) when detecting the term dependencies.

(3) For RM3, we tested different RM3 parameters, i.e., number of expansion terms, combination parameter λ. The number of pseudo-feedback documents is set to 5, which is the same to other tested contextual models. Note that, we re-implement it as a re-ranking model to keep consistent with other models.

(4) For all click based re-ranking models, we tested two different ways to select different click-through data when training the QLMs and LMs, i.e. all clicked documents (denoted "CLICK") and SAT-clicked documents (denoted "SAT-CLICK").

(5) For all click based re-ranking models, we tested different combination parameters (ξ and β in Equations 15 and 26).

5.4. Results and Discussion

The evaluation results of different models with different settings are reported in this section. We first analyze the best performance of each model with some specific settings (weight schemes, window size, CLICK or SAT-CLICK, combinations parameters and expansion term numbers etc.). Then, the influence of different parameter settings on the re-ranking performance is analyzed.

5.4.1. Overall Best Performance

In this section, we analyze the best experimental results of different re-ranking models in terms of RankScoring and R-Gain. As Table 1 illustrates, all of the re-ranking models improve the retrieval performance of original ranked results (positive increasing percentage of RankScoring and positive evaluation value of R-Gain).

(a) Best performance with respect to RankScoring			
Models	Best-RS	Increase	Settings
Original	39.77	0%	-
QLM	40.11	0.85%	IDF, $L = 8$
RM3	41.15	3.47%†	$d_n = 5, t_n = 20, \lambda = 0.3$
C-LM	46.57	17.09%‡	LM, $\xi = 0.1, \beta = 1$, SAT-Click
C-QLM	47.08	**18.39%**‡	IDF, $L = 8, \xi = 0.1, \beta = 1$, SAT-Click
AC-QLM	46.97%	18.1%‡	IDF, $L = 6, \xi = 0.1, \beta = 1$, SAT-Click

(b) Best performance with respect to R-Gain				
Models	#Inc	#Dec	Best-R-Gain	Settings
Original	0	0	0	-
QLM	255	133	0.3144	UNI, $L = 1$
RM3	247	128	0.3173	$d_n = 5, t_n = 20, \lambda = 0.3$
C-LM	493	166	0.4962†	$LM, \xi = 0.6, \beta = 0.3$, SAT-Click
C-QLM	528	174	**0.5043**†	UNI, $L = 1, \xi = 0.5, \beta = 0.3$, SAT-Click
AC-QLM	537	181	0.4958	UNI, $L = 1, \xi = 0.5, \beta = 0.3$, SAT-Click

Table 1: Best re-ranking performance for different models in terms of RankScoring (Table a) and R-Gain (Table b). In the "Settings" column of the two tables, we report the settings of each model corresponding to their best performance. The symbol ‡ means $p < 0.01$ with paired t-test, † means $p < 0.05$.

We first briefly analyze the results of basic QLM and RM3 compared with other contextual models. From both tables, the performance of QLM and RM3 are similar to original model and all contextual models outperform QLM and RM3 significantly. The results show that our proposed contextual quantum language model is better than basic QLM. Therefore, in the following, we focus on analyzing the comparative performance between C-LM and our proposed quantum language model, in order to study the difference of contextual modeling ability for contextual language model and contextual quantum language model.

Table (1-a) (using RankScoring evaluation metric in short RS) shows that all re-ranking models can improve the original results significantly. Moreover, our two QLM-based models outperformed the C-LM. Both the two QLM-based models gain the best performance with IDF weighting scheme and the same combination parameters. Re-ranking models trained with SAT-clicked documents are more effective than that with all clicked documents. It is a bit surprising that "AC-QLM" does not outperform the "C-QLM" in terms of RankScoring. A possible reason is that the process of applying "query change information" may introduce some risk while improving the query representation (Table (1-b) can support our hypothesis).

Table (1-b) reports the best results evaluated with R-Gain metric (reflecting both the effectiveness and robustness of a re-ranking model). All of the three models gain a positive improvement. The columns "#Inc" and "#Dec" respectively display the numbers of relevant documents whose rank is improved and decreased in comparison with the original rank. In this table, we find that C-QLM achieves the highest R-Gain value, while the AC-QLM has the lowest R-Gain value. From the "#Inc" and "#Dec" point of view, AC-QLM improves 537 relevant documents's rank, which is more than C-LM and C-QLM does. Unfortunately, AC-QLM also leads to a more number of decreasing ranks than the other two models. This phenomenon validates our hypothesis that applying query change information to C-QLM model can improve more relevant documents' rank, and meanwhile it may introduce some extra risk. The window sizes of QLM-based models are 1, which indicates that the best performance is achieved without considering term-dependency and in fact the weight scheme becomes meaningless. The performance of C-LM is not affected by the weighting schemes and window sizes.

As can be observed from two tables, three models achieve their best performance in different settings with respect to different evaluation metrics. One possible explanation is that two metrics focus on evaluating different properties of ranked results. More specifically, R-Gain only considers if a document's rank is improved or not after re-ranking, while RankScoring not only considers if the rank is improved, but also considers how much the rank is improved. As is shown in Table (1-a), all re-ranking models gain the best evaluation results with the same combination parameters ($\xi = 0.1$ and $\beta = 1$). According to Equation 15 and 26, ξ controls the impact strength of historical interaction QLMs on building of contextual QLM. The less ξ is, the more impact will be. β is the coefficient when combining the original results and the results from the re-ranking model, which controls how much the influence of contextual factor is on re-ranking. The larger β is, the more influence will be. The setting of combination parameters are $\xi = 0.1$ and $\beta = 1$, indicating that more contextual weight(β) and the historical interaction information weight (ξ) can lead to the largest improvement over the overall ranks most. In terms of R-Gain, the selection of parameters is just at the opposite: the less re-ranking model influence ($\beta = 0.3$) and less historical interaction influence ($\xi = 0.5$) leads to more robust re-ranking performance. Additionally, all models gain the best performance (in terms of two metrics) using SAT-Clicked documents, which demonstrates that using SAT criteria can filter out a remarkable amount of noises from the raw click-through data and thus improve the representation of user information need.

5.4.2. Performance on Different Training Parameters

In this section, we investigate how the training parameters (weighting schemes, fixed-window size, SAT-CLICK or CLICK) influence the training of QLM and further influence the re-ranking effectiveness. To this end, we first fix the other two combination parameters (ξ and β), and then analyze the evaluation results by varing the training parameters. In this paper, we select two parameter pairs to report evaluation results with respect to RankScoring and R-Gain. Specifically, we control $\xi_1 = 0.1, \beta_1 = 1$ to report the RankScoring evaluation results and control $\xi_2 = 0.5, \beta_2 = 0.3$ to report the R-Gain results, since the best performance of our models are achieved using these settings. Table 2 reports the evaluation results with respect to RankScoring and R-Gain. C-LM is based on traditional unigram language model. Therefore, its performance is not affected by window size and weighting schemes (UNI or IDF). QLM-based models outperform the C-LMs at various settings.

Table (a): Performance (RankScoring (+%)) on Training Parameters (control $\xi_1 = 0.1, \beta_1 = 1$)						
L	C/S	C-LM	C-QLM		AC-QLM	
			UNI	IDF	UNI	IDF
1	C	16.43	14.36	14.52	14.56	17.35
	S	**17.09**	15.94	16.05	14.86	16.67
2	C	16.43	14.21	14.43	15.71	15.86
	S	**17.09**	15.57	16.67	16.02	15.73
4	C	16.43	14.87	14.74	15.32	15.74
	S	**17.09**	16.09	17.15	16.11	16.72
6	C	16.43	14.04	14.08	16.35	16.77
	S	**17.09**	15	16.06	**17.31**	**18.1**
8	C	16.43	14.47	17.33	15.75	17.25
	S	**17.09**	**16.14**	**18.39**	16.68	17.66

Table (b): Performance (R-Gain) on Training Parameters (control $\xi_2 = 0.5, \beta_2 = 0.3$)														
L	C/S	C-LM			C-QLM						AC-QLM					
					UNI			IDF			UNI			IDF		
		#I	#D	R-Gain	#I	#D	R-Gain	#I	#D	R-Gain	#I	#D	R-Gain	#I	#D	R-Gain
1	C	486	169	0.4840	528	177	0.4979	530	179	0.4951	534	182	0.4916	537	196	0.4652
	S	487	167	**0.4893**	528	174	**0.5043**	529	177	**0.4986**	537	181	**0.4958**	540	194	**0.4714**
2	C	486	169	0.4840	528	183	0.4852	530	181	0.4909	534	186	0.4833	534	199	0.4570
	S	487	167	**0.4893**	527	182	0.4866	529	181	0.4901	541	184	0.4924	539	199	0.4607
4	C	486	169	0.4840	528	189	0.4728	534	186	0.4833	543	192	0.4776	547	201	0.4626
	S	487	167	**0.4893**	531	187	0.4791	536	184	0.4889	543	191	0.4796	547	201	0.4626
6	C	486	169	0.4840	488	168	0.4878	528	183	0.4852	538	186	0.4862	546	200	0.4638
	S	487	167	**0.4893**	533	196	0.4623	528	181	0.4894	541	186	0.4883	547	200	0.4645
8	C	486	169	0.4840	526	187	0.4755	529	185	0.4818	533	196	0.4623	542	208	0.4453
	S	487	167	**0.4893**	529	183	0.4860	531	188	0.4771	536	196	0.4645	542	208	0.4453

Table 2: Table (a) reports improvement percentages in terms of RankScoring on different training parameters given a controlling parameter pair ξ_1 and β_1. Table (b) is the Performance on different training parameters with respect to R-Gain given the parameter pair ξ_2 and β_2. In the tables, L represents window size, "C/S" represents CLICK/SAT-CLICK. $\#I$ represents $\#Inc$, $\#D$ represents $\#Dec$. The black fonts highlight the best evaluation results in a column.

Table (2-a) and (2-b) show that given a setting of window size and weighting scheme, the models with "SAT-CLICK" setting have better performance than models with "CLICK" setting. This phenomenon further validates our point that using "SAT-Click" criteria can filter out some noise that are useless for detecting user's information needs.

Window size influences the re-ranking effectiveness in some more complex way. Table (2-a) illustrates that under a specific setting of "C/S" (CLICK/SAT-

CLICK) and weighting scheme, the QLM-based models tend to have better performance over larger window size (apart from some special cases, e.g., AC-QLM with settings $L = 1$, CLICK, IDF). AC-QLM reaches the best results when $L = 6$, and C-QLM when $L = 8$ (we did not test the window size greater than 8, as our focus is on improving precision). As is illustrated in Table (b), there is a trend that the R-Gain evaluation value decreases with the increase of window size, while the "#Inc" and "#Dec" increase slightly with the increase of L. In other words, larger window size can improve the re-ranking performance to some extent, and meanwhile introduce some risks.

The weighting schemes have a similar impact on evaluation results as window size does. Table (2-a) shows that QLM-based models have relative better evaluation results when using IDF weighting than using UNI in terms of RankScoring. Table (2-b) shows the opposite trend in terms of R-Gain evaluation metric. QLM-based models with IDF weighting scheme improve more relevant documents' ranks while also hurt more than that with UNI scheme after re-ranking, while also harm more documents rank. This phenomenon illustrates that the nice-to-do properties of IDF weighting scheme may contribute to represent more complex semantic information of term dependencies, but meanwhile may add some uncertainty to the representation of information needs. Having said that, we can still expect to find a more reliable weighting scheme to reduce the risk. This will be left to the future work. One case should be clarified. In theory, the effectiveness of C-QLMs (or AC-QLMs) using two weighting schemes (UNI or IDF) when L=1 should be equal, because when L=1 term dependencies are not detected. However they are not identical in the experimental results, because some slight difference exists in the training process of QLMs.

5.4.3. Re-ranking Effectiveness and Stability over Different Combination Parameters

In this section, we will study how the combination parameters (ξ and β) influence the effectiveness and stability of re-ranking models. To this end, we conduct a "Mean Effectiveness and Variance" analysis [34] under different combination parameter pairs, to determine the impact of historical interaction information and original ranked results on re-ranking. To be fair, we select the best training parameters (L, weighting schemes, "C/S", See Table 3) of each model as controlling factors, then test the re-ranking performance under different parameter pairs ($\xi \in \{0, 0.1, 0.2, ..., 1\}$ and $\beta \in \{0.1, 0.2, ..., 1\}$, the possible number of pairs is $11 \times 10 = 110$).

Models	RankScoring-Variance			RGain-Variance		
	L	C/S	UNI/IDF	L	C/S	UNI/IDF
Original	-	-	-	-	-	-
C-LM	-	SAT	-	-	SAT	-
C-QLM	6	SAT	IDF	1	SAT	UNI
AC-QLM	8	SAT	IDF	1	SAT	UNI

Table 3: Training parameters setting for different models when conducting the "Mean Effectiveness and Variance" analysis corresponding to RankScoring and R-Gain.

Figure 4 presents an analysis of results, in which Figure (a) and Figure (b) correspond to Average RankScoring and R-Gain respectively. The y-axis is the mean effectiveness corresponding to an evaluation metric, x-axis is the standard deviations (STD) of the evaluation metric for different queries. $STD(M) = \sqrt{E(M_q - E(M_q))}$, where M is an evaluation metric, M_q is the evaluation value for a query q, $E(M_q)$ is the expectation of the evaluation value over all queries. In this paper, $E(RankScoring)$ is the average RankScoring, $E(R\text{-}Gain)$ is the R-Gain value for all queries, and $RankScoring(q)$ is obtained by computing the average RankScoring for one query (see Equation 27). The larger y-value indicates the better average effectiveness of a model, and the smaller STD(M) indicates the better retrieval stability of evaluation value across queries (the lower risk). Each point in the figure corresponds to one model with a specific combination parameters pair (ξ and β). Each re-ranking model has 110 points.

Figure (4-a) shows that all re-ranking models improve the retrieval effectiveness and stability in comparison with the original ranking model (corresponding to the point denoted by the black square symbol). Our QLM-based models achieve the better effectiveness and the more stability than C-LMs. Specifically, the points in elliptic region A have the best effectiveness and the best stability. Moreover, most points in region A are corresponding to the QLM-based models. Through analyzing the parameter settings of points, we find that all points have the the settings $\beta = 1$ and $\xi \in \{0, 0.1, 0.2, ..., 1\}$, which means abandoning the original ranking information, can gain the best effectiveness and the best stability. However, in this region, we do not find any obvious correlation (positive or negative) between the value ξ and $STD(RankScoring)$. We consider that there should exist more complex mechanism to influence the stability. Points in region B (only contains the C-LM's points) have good effectiveness, but their STDs are far greater than those that in A, leading to less stability. The region C, which mainly contains the points of QLM-based models have relative better effectiveness and stability than the region D. The settings of points in region C are $\beta \in \{0.4, 0.5, ..., 0.9\}$ and $\xi \in \{0, 0.1, 0.2, ..., 1\}$. Most points of C-LM fall in the worst region D.

As Figure (4-b) demonstrates, we find that, for evaluation metric R-Gain, the best region is region E, where the QLM-based models have better effectiveness and equal stability in comparison with C-LMs. Region F is the worst region, which shows that the C-LM get the worst effectiveness and the worst stability. An interesting phenomenon is that the points are distributed on different ribbon regions, and the large majority of points in one region have the same setting of β. More specifically, the larger β is, the larger STD (less stability) of the points in the specific ribbon region will be, which is opposite to that in Figure (4-a). This phenomenon is caused by the different properties of two evaluation metrics. Observing the re-ranking results, we find that "#Inc" and "#Dec" of models with larger β are larger than that with smaller β.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose to apply the Quantum Language Model (QLM) in Session Search. Specifically, we develop two Session based QLMs, namely Contextual query QLM (C-QLM) and Adaptive Contextual query QLM (AC-

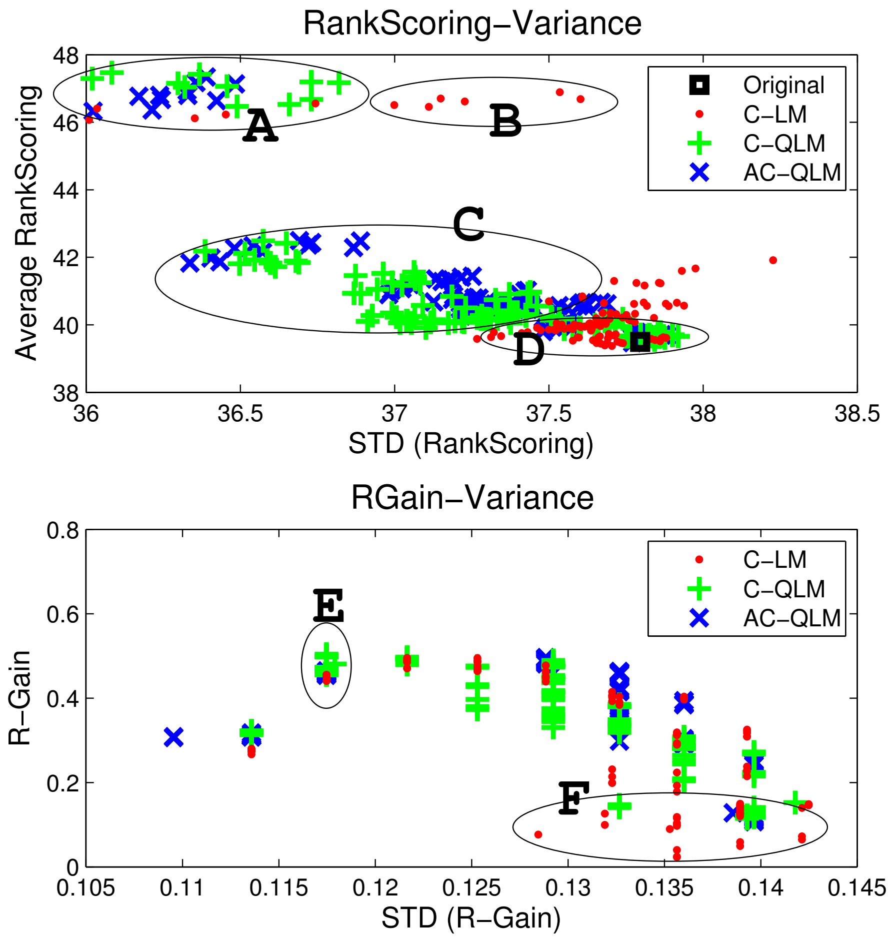


Figure 4: Re-ranking effectiveness and stability over different combination parameters. Figure (a) and (b) correspond to RankScoring and R-Gain evaluation metrics respectively.

QLM). C-QLM trains a contextual query QLM based on the quantum events (i.e., projectors associated with single terms or compound terms) extracted from the historical clicked documents and the top-most documents in original results for the current query. Moreover, we incorporate the dynamic query change information into the training of contextual query QLM, leading to an Adaptive Contextual query QLM (AC-QLM). We utilize the density matrix of C-QLM as the initial density matrix for AC-QLM, and an adaptive density matrix is then trained by maximizing a new designed likelihood function. The likelihood function rewards the quantum events associated with the added and common terms in the current query compared with the immediate previous query, and correspondingly penalizes those associated with the removed terms.

We have conducted extensive experiments on a subset of query logs collected form a prominent commercial search engine. The experimental results show that all contextual models (including C-QLM, AC-QLM and Contextual Language Model (C-LM)) improve the original results significantly, demonstrating that a search engine can largely benefit from incorporating implicit knowledge embedded in user interactions. The proposed Session based QLMs (C-QLM and AC-QLM) outperform C-LM significantly. This shows that quantum language model can effectively model the compound term as the superposition of elementary quantum events encapsulated into density matrices. Although AC-QLM does not outperform C-QLM in terms of the average evaluation metrics (RankScoring and R-Gain), it improves more number of relevant documents' rank compared with C-QLM. This phenomenon shows that AC-QLM achieves some gain while also brings some extra risks.

In the future, we will investigate more stable methods to better capture the query change information for the Adaptive Contextual query QLM. We still investigate on how to reduce the number of parameters in our approaches and moreover develop an automatic parameter selection mechanism. Additionally, we have observed a great potential to represent dynamic knowledge, e.g., in the form of term dependence patterns, with density matrix for a complex information system, e.g., a search engine. Therefore, the density matrix may be applied to wider contexts in the future. For example, a density matrix may be used to represent an entity or an entity relationship in a knowledge base. The retrieval model based on user interactions can also be extended to other applications, such as the question answering (QA). In a QA system, user interactions can reveal much implicit knowledge about the users' cognition level for some specific topics. The density matrix can be used for representing the cognition level for a user. Overall, the modeling of user interactions in information systems with the density matrix would seem a promising research topic in the future.

Acknowledgements. This work is supported in part by the Chinese National Program on Key Basic Research Project (973 Program, grant No.2013CB329304, 2014CB744604), the Chinese 863 Program (grant No. 2015AA015403), the Natural Science Foundation of China (grant No. 61272265, 61402324), and the Tianjin Research Program of Application Foundation and Advanced Technology (grant no. 15JCQNJC41700).

References

[1] A. Sordoni, J.-Y. Nie, Y. Bengio, Modeling term dependencies with quantum language models for ir, in: SIGIR, ACM, 2013, pp. 653–662.

[2] H. Umegaki, et al., Conditional expectation in an operator algebra. iv. entropy and information, in: Kodai Mathematical Seminar Reports, Vol. 14, Tokyo Institute of Technology, Department of Mathematics, 1962, pp. 59–85.

[3] D. Guan, S. Zhang, H. Yang, Utilizing query change for session search, in: SIGIR, ACM, 2013, pp. 453–462.

[4] S. Zhang, D. Guan, H. Yang, Query change as relevance feedback in session search, in: SIGIR, ACM, 2013, pp. 821–824.

[5] G. Akrivas, M. Wallace, G. Andreou, G. Stamou, S. Kollias, Context-sensitive semantic query expansion, in: (ICAIS 2002), IEEE, 2002, pp. 109–114.

[6] M. Bertier, R. Guerraoui, V. Leroy, A.-M. Kermarrec, Toward personalized query expansion, in: ACM EWSNS, ACM, 2009, pp. 7–12.

[7] P.-A. Chirita, C. S. Firan, W. Nejdl, Personalized query expansion for the web, in: SIGIR, ACM, 2007, pp. 7–14.

[8] Z. Dou, R. Song, J.-R. Wen, A large-scale evaluation and analysis of personalized search strategies, in: WWW, ACM, 2007, pp. 581–590.

[9] M. Harvey, F. Crestani, M. J. Carman, Building user profiles from topic models for personalised search, in: CIKM, ACM, 2013, pp. 2309–2314.

[10] J. Li, D. Song, P. Zhang, J.-R. Wen, Z. Dou, Personalizing web search results based on subspace projection, in: Information Retrieval Technology, Springer, 2014, pp. 160–171.

[11] M. Melucci, R. W. White, Utilizing a geometry of context for enhanced implicit feedback, in: CIKM, ACM, 2007, pp. 273–282.

[12] C. J. Van Rijsbergen, The geometry of information retrieval, Cambridge University Press, 2004.

[13] B. Piwowarski, I. Frommholz, M. Lalmas, K. Van Rijsbergen, What can quantum theory bring to information retrieval, in: CIKM, ACM, 2010, pp. 59–68.

[14] X. Zhao, P. Zhang, D. Song, Y. Hou, A novel re-ranking approach inspired by quantum measurement, in: Advances in Information Retrieval, Springer, 2011, pp. 721–724.

[15] G. Zuccon, L. Azzopardi, Using the quantum probability ranking principle to rank interdependent documents, in: Advances in information retrieval, Springer, 2010, pp. 357–369.

[16] P. Zhang, D. Song, X. Zhao, Y. Hou, Investigating query-drift problem from a novel perspective of photon polarization, Lecture Notes in Computer Science (2011) 332–336.

[17] A. Sordoni, Y. Bengio, J.-Y. Nie, Learning concept embeddings for query expansion by quantum entropy minimization, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

[18] Q. Li, J. Li, P. Zhang, D. Song, Modeling multi-query retrieval tasks using density matrix transformation, in: SIGIR, ACM, 2015, pp. 871–874.

[19] M. Xie, Y. Hou, P. Zhang, J. Li, W. Li, D. Song, Modeling quantum entanglements in quantum language models.

[20] M. K. Warmuth, D. Kuzmin, Bayesian generalized probability calculus for density matrices, Machine learning 78 (1-2) (2010) 63–101.

[21] J. Lafferty, C. Zhai, Document language models, query models, and risk minimization for information retrieval, in: SIGIR, ACM, 2001, pp. 111–119.

[22] C. Zhai, J. Lafferty, Model-based feedback in the language modeling approach to information retrieval, in: CIKM, ACM, 2001, pp. 403–410.

[23] A. Lvovsky, Iterative maximum-likelihood reconstruction in quantum homodyne tomography, Journal of Optics B: Quantum and Semiclassical Optics 6 (6) (2004) S556.

[24] M. A. Nielsen, I. L. Chuang, Quantum computation and quantum information, Cambridge university press, 2010.

[25] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: SIGIR, ACM, 2001, pp. 334–342.

[26] D. Gayo-Avello, A survey on session detection methods in query logs and a proposal for future evaluation, Information Sciences 179 (12) (2009) 1822–1843.

[27] D. He, A. Göker, Detecting session boundaries from web user logs, in: Proceedings of the BCS-IRSG 22nd annual colloquium on information retrieval research, 2000, pp. 57–66.

[28] T. Kramár, M. Bieliková, Detecting search sessions using document metadata and implicit feedback, in: WSCD 2012, 2012.

[29] B. Tan, X. Shen, C. Zhai, Mining long-term search history to improve search accuracy, in: SIGKDD international conference on Knowledge discovery and data, ACM, 2006, pp. 718–723.

[30] M. F. Porter, An algorithm for suffix stripping, Program: electronic library and information systems 14 (3) (1980) 130–137.

[31] T. Vu, D. Song, A. Willis, S. N. Tran, J. Li, Improving search personalisation with dynamic group formation.

[32] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, X. Cui, Modeling the impact of short-and long-term behavior on search personalization, in: SIGIR, ACM, 2012, pp. 185–194.

[33] V. Lavrenko, W. B. Croft, Relevance based language models, in: SIGIR, ACM, 2001, pp. 120–127.

[34] P. Zhang, D. Song, J. Wang, Y. Hou, Bias-variance decomposition of ir evaluation, in: SIGIR, ACM, 2013, pp. 1021–1024.

Highlights

1. An adaptive quantum language model for Information retrieval is proposed.
2. Users' dynamic information need is captured by the model.
3. Extensive experiments have shown its effectiveness.