Université de Montréal

Learning Representations for Information Retrieval

par Alessandro Sordoni

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Mars, 2016

Résumé

La recherche d'informations s'intéresse, entre autres, à répondre à des questions comme: est-ce qu'un document est pertinent à une requête ? Est-ce que deux requêtes ou deux documents sont similaires ? Comment la similarité entre deux requêtes ou documents peut être utilisée pour améliorer l'estimation de la pertinence ? Pour donner réponse à ces questions, il est nécessaire d'associer chaque document et requête à des représentations interprétables par ordinateur. Une fois ces représentations estimées, la similarité peut correspondre, par exemple, à une distance ou une divergence qui opère dans l'espace de représentation. On admet généralement que la qualité d'une représentation a un impact direct sur l'erreur d'estimation par rapport à la vraie pertinence, jugée par un humain. Estimer de bonnes représentations des documents et des requêtes a longtemps été un problème central de la recherche d'informations. Le but de cette thèse est de proposer des nouvelles méthodes pour estimer les représentations des documents et des requêtes, la relation de pertinence entre eux et ainsi modestement avancer l'état de l'art du domaine. Nous présentons quatre articles publiés dans des conférences internationales et un article publié dans un forum d'évaluation. Les deux premiers articles concernent des méthodes qui créent l'espace de représentation selon une connaissance à priori sur les caractéristiques qui sont importantes pour la tâche à accomplir. Ceux-ci nous amènent à présenter un nouveau modèle de recherche d'informations qui diffère des modèles existants sur le plan théorique et de l'efficacité expérimentale. Les deux derniers articles marquent un changement fondamental dans l'approche de construction des représentations. Ils bénéficient notamment de l'intérêt de recherche dont les techniques d'apprentissage profond par réseaux de neurones, ou *deep learning*, ont fait récemment l'objet. Ces modèles d'apprentissage élicitent automatiquement les caractéristiques importantes pour la tâche demandée à partir d'une quantité importante de données. Nous nous intéressons à la modélisation des relations sémantiques entre documents et requêtes ainsi qu'entre deux ou plusieurs requêtes. Ces derniers articles marquent les premières applications de l'apprentissage de représentations par réseaux de neurones à la recherche d'informations. Les modèles proposés ont aussi produit une performance améliorée sur des collections de test standard. Nos travaux nous mènent à la conclusion générale suivante: la performance en recherche d'informations pourrait drastiquement être améliorée en se basant sur les approches d'apprentissage de représentations.

Mots-clés: modèle de recherche, suggestion de requête, recherche ad-hoc, expansion de requête, théorie quantique, matrice de densité, vecteurs de mot, apprentissage supervisé, réseaux de neurones, apprentissage profond.

Summary

Information retrieval is generally concerned with answering questions such as: is this document relevant to this query? How similar are two queries or two documents? How query and document similarity can be used to enhance relevance estimation? In order to answer these questions, it is necessary to access computational representations of documents and queries. For example, similarities between documents and queries may correspond to a distance or a divergence defined on the representation space. It is generally assumed that the quality of the representation has a direct impact on the bias with respect to the true similarity, estimated by means of human intervention. Building useful representations for documents and queries has always been central to information retrieval research. The goal of this thesis is to provide new ways of estimating such representations and the relevance relationship between them. We present four articles that have been published in international conferences and one published in an information retrieval evaluation forum. The first two articles can be categorized as feature engineering approaches, which transduce a priori knowledge about the domain into the features of the representation. We present a novel retrieval model that compares favorably to existing models in terms of both theoretical originality and experimental effectiveness. The remaining two articles mark a significant change in our vision and originate from the widespread interest in deep learning research that took place during the time they were written. Therefore, they naturally belong to the category of representation learning approaches, also known as feature learning. Differently from previous approaches, the learning model discovers alone the most important features for the task at hand, given a considerable amount of labeled data. We propose to model the semantic relationships between documents and queries and between queries themselves. The models presented have also shown improved effectiveness on standard test collections. These last articles are amongst the first applications of representation learning with neural networks for information retrieval. This series of research leads to the following observation: future improvements of information retrieval effectiveness has to rely on representation learning techniques instead of manually defining the representation space.

Keywords: retrieval models, query suggestion, ad-hoc retrieval, query expansion, quantum theory, density matrix, word embeddings, supervised learning, neural network, deep learning, recurrent neural networks.

Contents

List of Figures

List of Tables

List of Abbreviations

DSSM	Deep Structured Semantic Models
ERR	Expected Reciprocal Rank
HRED	Hierarchical Recurrent Encoder-Decoder
IR	Information Retrieval
IDF	Inverse Document Frequency
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
LM	Language Model
MAP	Mean Average Precision
MRF	Markov Random Field
MRR	Marginal Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain
NN	Neural Network
QT	Quantum Theory
QLM	Quantum Language Model
QEM	Quantum Entropy Minimization
SGD	Stochastic Gradient Descent
TF	Term Frequency
VSM	Vector-Space Model
VN	Von Neumann

Acknowledgments

This thesis is dedicated to Sylvaine, David, Véronique, Emanuele, Giorgio and Guy, for their endless and generous friendship. I would like to especially thank my supervisor, Prof. Jian-Yun Nie, for his guidance, support and constant trust in my ideas. That was invaluable. I thank Yoshua Bengio for teaching me the value of doing good research and for his constant technical and moral support. I grew a lot by interacting with him. Finally I thank my parents, Massimo and Roberta, my brother Andrea and my grandmother Lina for making me understand the value of education and for their love throughout the thesis.

1 Introduction

In this thesis, our aim is to estimate more accurate representations of documents and queries and to adapt these representations for specific information retrieval tasks. Estimating representations for document and queries is central to information retrieval. We hope to advance the state of the art by proposing new solutions to this problem. To support our purpose, we present four articles that have been published in peer-reviewed international conferences and one published in a retrieval evaluation forum.

This chapter provides some background on the aspects of information retrieval we will be dealing with along with the general motivation of our work. In addition, we outline the content of the articles we present in this thesis. The following chapters introduce previous related work and the datasets we will use to support our experimental evaluation.

1.1 Research Context

An information retrieval (IR) process can be viewed as a user-based needle-in-a-haystack problem: a user seeking for information is typically faced with an enormous amount of varied and interconnected information items. A retrieval system is committed to support the user in the search process by returning the information items relevant to her information need. In this thesis, we focus on document search, for which information items are textual documents, such as web pages, and the information need is expressed by means of a textual query.

In its basic form, a retrieval system consists of an index structure, which provides fast access to the documents in the collection, and a retrieval model, which is an algorithm responsible for predicting the relevance of each document with respect to the user query. Predicting relevance is a complex task that may depend on

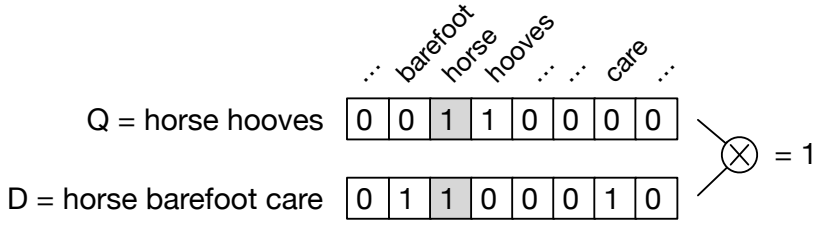


Figure 1.1 – In the vector space model (Salton et al., 1975), documents and queries are represented as vectors in a high-dimensional term space. Relevance is assumed to be correlated to the degree of terms matching between documents and queries. The assumption that term matching is important for retrieval is hard-coded into the model. The example uses a binary weighting corresponding to the presence of a term, but other choices are possible (Salton, 1986).

multiple factors including the quality of a document itself, the current geographical context and educational background of the user and on other documents the user has previously inspected during the search. In order to harness the complexity of relevance prediction, industrial search engines such as Google [1] and Bing [2] are multi-modular systems employing a number of retrieval models and additional strategies. For example, when the query is short or ambiguous, they may opt to explicitly diversify the search results to cover all possible aspects of the user query and to minimize the risk of user dissatisfaction. Moreover, they offer query autocompletion or suggestion services, promoting those reformulations that are more likely to lead to the documents the user is searching for.

In order to provide the aforementioned services, retrieval systems do need to come up with computational representations of documents and queries. In other words, they need to transform the textual input and other useful contextual information into a representation that can be processed by internal algorithms. For example, in the vector-space retrieval model (VSM) (Luhn, 1957; Salton et al., 1975; Salton and Buckley, 1988), documents and queries are represented as vectors in a high-dimensional vector-space. The dimensions of the vector-space may correspond to single terms, phrases or other indexing units, depending on the manual choices of the indexer. An entry of the document (query) vector is non-zero if the corresponding index unit appears in the document (query). The relevance of a document is estimated by computing a distance measure between the document vector and the query vector (see Figure 1.1).

1. http://www.google.ca
2. http://www.bing.com

Finding suitable query and document representations in order to perform retrieval has always been a central problem in IR. In general, most retrieval models associate both the document and the query to their corresponding representations and estimate the relevance based on the degree of matching between the representations. Some retrieval models, such as the early probabilistic Binary Independence Retrieval model (Robertson and Jones, 1976; Rijsbergen, 1979; Croft and Harper, 1979), drop the explicit query representation and directly estimate the probability of relevance given a document. Unlike the vector-space model, the probabilistic framework provides formal guidance in order to approximate relevance. However, the absence of a query representation makes it difficult to incorporate additional query evidence into the model (Metzler, 2011). The initial Language Modeling (Ponte and Croft, 1998) approach also sacrifices the query representation by interpreting the query as a sequence of samples from the document distribution. Later, Lafferty and Zhai (2001); Zhai (2008) generalize the LM approach by restoring an explicit query representation: both queries and documents are associated to n-gram distributions and relevance correlates to a probabilistic divergence between the distributions. The flexible model developed by Turtle and Croft (1989, 1991) provides another example in which queries and documents are explicitly represented, in this particular case by means of an inference network. In general, by having separate document and query representations, it is possible to integrate evidence about other queries that point to the same relevant document and about other documents that are relevant to the same query (Robertson et al., 1983; Robertson, 2003; Bodoff and Robertson, 2004). Research on retrieval models has been moving towards representational frameworks flexible enough to allow the integration of evidence about the query and the document in a principled way (Croft, 2002; van Rijsbergen, 2004; Melucci, 2008; Buccio et al., 2010).

Regardless of the specific representational framework, when task-oriented, query and document representations should be estimated in such a way that they make the task easier to succeed. For example, in the vector-space model, the ideal configuration would put the query representation close to its relevant documents and far away from all its non-relevant documents (Rocchio, 1971). In this case, relevance can be easily computed by calculating the distance between the query and the document representations. If the task is to suggest alternative queries to the user, this kind of property should hold for similar queries. Therefore, as a general

rule, the representation should account for those input characteristics, or features, that are important for the task and be invariant to, i.e. do not change for, those input traits that have no impact on the task.

Historically, retrieval models tried to enforce this principle by hard-coding into the representations those features that were known to correlate well with the task. This *feature engineering* approach is adopted by basic retrieval models, such as the vector-space and language models, which generally rely on the intuitive assumption that term or phrase matching correlates with relevance (see Figure 1.1). Another example of prior knowledge is the use of document term frequency (TF). The fact that the frequency of a query term in a document is important to decide whether a document is "about" the query is hard-coded into the representation itself by weighting the corresponding term feature. Curiously, the common use of term frequency coupled with inverse document frequency (IDF), i.e. the well-known *tf-idf* weighting scheme (Salton and Buckley, 1988), can be interpreted as an attempt to make the representation invariant to those terms that are less discriminative for relevance, i.e. the representations tend to be invariant to overly common terms having a low IDF value. More recent and complex non-bag-of-words (Metzler and Croft, 2005; Bendersky and Croft, 2012) and learning-to-rank (Li, 2011) techniques adopt a similar approach by engineering a feature space composed by various evidence and by learning a complex, composite function mapping the fixed feature space to the desired output.

One of the shortcomings of the previous approach is that it is usually unhandy or even impossible to enumerate all the features that could help in predicting relevance. In supervised *feature learning*, the representation features are automatically learned in order to maximize the success of the task, i.e. one learns both the mapping from the input to the feature space and from the features to the output. This is a relatively new avenue in information retrieval research and is based on the emerging representation learning techniques based on deep learning architectures. Deep learning refers to the layered structure of the learning algorithm, each layer corresponding to an increasingly abstract representation of the input data. The major advantage of this approach is that there is little to no prior knowledge involved. The learning algorithm is able to carve its own feature space and to detect the proper invariances in such a way that the discovered representations minimize the task error. This research direction allowed to obtain ground-breaking results in

computer vision and general natural language processing (Goodfellow et al., 2016) but has just started to be explored in IR (Shen et al., 2014; Severyn and Moschitti, 2015; Mitra, 2015; Zheng and Callan, 2015). One of the most prominent examples of feature learning is the estimation of *words embeddings* (Bengio et al., 2003; Mikolov et al., 2013), i.e. each word is associated to a vector encoding syntactic and semantic characteristics thereof. As a consequence, the distance between word vectors corresponds to a degree of relatedness between words. The dimensions of the representation space do not directly encode the a priori knowledge about word similarities, which would be impractical for large vocabularies, but rather correspond to semantic descriptors that are automatically learned to fit the task at hand.

Our view is that the latter techniques, eventually in combination with established retrieval approaches, are well-suited to learn high-quality representations for documents and queries and may set a new state of the art in IR. For example, when applied to relevance estimation, the learned feature space may encode semantic features of documents and queries thus moving retrieval models closer to the human relevance assessment process. The papers included in this thesis are stepping stones on a path that, departing from feature engineering methods, led us to the application of feature learning models for IR. Particularly, the first two articles are naturally based on the paradigm of feature engineering. The last two articles are an attempt to move general IR towards feature learning approaches.

1.2 Articles Outline

The first article has been published in the proceedings of the 2013 Quantum Interaction (QI) conference and is presented in Chapter 4. We analyze the representational assumptions made by well-known term matching retrieval models, such as VSM (Salton, 1986) and the Language Models (LM) (Ponte and Croft, 1998; Zhai, 2008) for IR. We unify the methodologies using the mathematical framework borrowed from quantum physics. Unlike previously thought, our study reveals that the approaches are rather complimentary. We highlight that it is possible to mix the strengths of both models.

The second article has been published in the proceedings of the 2013 SIGIR conference (Annual Conference of the Special Interest Group in Information Retrieval) and is presented in Chapter 5. It presents an operational instantiation of the idea presented in the previous chapter. Our Quantum Language Model (QLM) retrieval approach is a hybrid VSM/LM approach that represents documents and queries as matrices. The rich representation space allows us to consider both term-level matching and phrase-level matching without artificially extending the term space to account for phrases, as previously done in the literature. QLM achieves statistical significant improvements over strong non-bag-of-words models and establishes a new state-of-the-art amongst phrase-based retrieval models for web retrieval. We also include the results of our participation in the Text REtrieval Conference evaluation forum (TREC 2013) using the same model, with which good results have been obtained.

The third article has been published in the proceedings of the 2014 Association for the Advancement of Artificial Intelligence (AAAI) and is presented in Chapter 6. It marks our first exploration of feature learning models for IR. Contrarily to the first two articles, in which documents and queries are embedded into an artificially created term space, we estimate, by means of a novel algorithm, a feature space whose dimensions do not necessarily correspond to single terms but rather to semantic descriptors. Our learning algorithm is trained using a *query log*, i.e. a collection of query and relevant document pairs (as described in Section 3.3). Our objective function pushes a query representation near its relevant documents and far away from all its non-relevant documents. The resulting representations are used to improve ad-hoc retrieval in a query expansion setting.

The fourth article has been published in the proceedings of the 2015 Conference of Information Knowledge and Management (CIKM) and is presented in Chapter 7. We propose a way to estimate query representations for query suggestion through a deep learning based architecture. Unlike existing suggestion models exploiting query co-occurrence features, our system automatically learns which features are important for query suggestion and represents one of the first applications of deep learning to IR.

2 Related Works

This chapter provides an overview of the literature related to this thesis. Section 2.1 narrates a short history of vector-space, language and quantum-based retrieval models along with a highlight of their representational assumptions. We focus on both bag-of-words and term dependency models, which will provide the necessary basis to contextualize Chapter 4 and 5. In Section 2.2, we briefly introduce the foundational works on the field of representation learning and its deployment in IR, which is relevant to Chapters 6 and 7.

2.1 Retrieval Models

2.1.1 Vector Space Models

In the well-known vector space model (VSM) (Salton et al., 1975), documents and queries are represented in a vector space whose dimensions correspond to single terms. The coordinates for each document/query are determined by leveraging weight functions such as inverse document frequency (IDF) and term frequency (TF). The relevance score for a document corresponds to the inner product between the document and the query vectors. The assumed orthogonality between terms causes the exact matching problem, i.e. only documents that contain at least one query term have a non-null score. This is clearly a simplistic assumption due to linguistic phenomena such as synonymy.

Fagan (1987) and successively Mitra et al. (1997) try to incorporate phrases, such as multiword concepts, into the VSM. Phrases are considered additional dimensions in the representation space, orthogonal to their component terms. The score of a phrase in a document is the average of the TF-IDF (Salton and Buckley, 1988) weights of its component terms. The dependencies between phrases and component terms are either ignored or taken care of in an ad-hoc fashion by weighting

the constituent words more than the phrases. In these models, the ranking function boils down to a combination of scores from single terms and from phrases:

$$s(Q,D) = w_{term} \cdot \underbrace{s_{term}(Q,D)}_{\text{term score}} + w_{phr} \cdot \underbrace{s_{phr}(Q,D)}_{\text{phrase score}},$$

where w_{term}, w_{phr} are the combination weights for the term score and phrase score respectively. Phrases are considered as additional indexing units and their importance is adjusted in order to achieve weight normalization (Jones et al., 2000a), i.e. to compensate for the fact that the occurrence of single terms is counted twice, in the term score and in the phrase score. Addressing the weight normalization problem in new ways will be one of the main foci of Chapter 5.

Other attempts of incorporating phrases into a retrieval model belong to the literature of passage-retrieval with vector-space models (Kaszkiel et al., 1999; Kaszkiel and Zobel, 2001). The document is split into (non-)overlapping windows of contiguous words, i.e. passages, and a vector is built from each passage. The query vector is then compared to each passage vector and the obtained scores are aggregated for each document. Passage retrieval has potential advantages such as encoding proximity information. Proximity is a clear indicator of relevance, i.e. a document having a short passage with a lot of query words is more likely to be relevant than a document with no such passage (Kaszkiel et al., 1999).

One of the strengths of the VSM approach is the explicit definition of two separate representations, for the query and for the document. Keeping two separate representations gives the flexibility of integrating evidence both in the query and in the document (Zhai, 2007). In the VSM, this flexibility comes at the price of the heuristic flavor of the weight schemes and the multiplicity of existing scoring functions. Considering phrases as additional indexing units comes at the cost of having no guidance for the estimation of coordinates on such dimensions. Among the works that pinned these issues, Zobel and Moffat (1998) conclude that "no component or weight scheme was shown to be consistently valuable across all of the experimental domains" and "the measures do not form a space that can be explored in any meaningful way, other than by exhaustion".

2.1.2 Language Models

Language models (LM) for IR (Ponte and Croft, 1998) were at first seen as a solution to the heuristic flavor of vector space models weight schemes. In IR, a statistical language model ranks a document by the likelihood of a query given the probabilistic model associated to the document. Each document is associated to an unigram language model, i.e. $\theta_d = (\theta_{d1}, \ldots, \theta_{dV})$, where θ_{di} corresponds to the probability of observing word i in the vocabulary V. Therefore, the query likelihood writes as $p(q|\theta_d) = \prod_i p(q_i|\theta_d)$, leading to the multinomial language model. However, other choices are also possible (Ponte and Croft, 1998; Zhai, 2007; Bravo-Marquez et al., 2010). The assumption of word independence is always assumed by the factorization of the joint probability [1]. In Chapter 4, we will explore interesting links between the representational assumptions of independence posited by LM and VSM.

The advent of the LM approach opens new perspectives for integrating phrases into the retrieval model. Srikanth and Srihari (2002) claim that "the elegance of LM approach to IR facilitates a better representation of the dependencies between the constituent words of a phrase". Term dependencies such as those arising in phrases may be modeled as joint probabilities. Song and Croft (1999) take into consideration bigrams and trigrams. The scoring function considers whether the document can generate the n-grams appearing in the query. This approach turned out to lack flexibility because a document containing "retrieval of information" could also be relevant to the query "information retrieval". Srikanth and Srihari (2002) propose to relax the strong bigram assumption by modeling biterms. A biterm is defined as an unordered occurrence of two terms. These probabilities being difficult to compute, the authors propose three different heuristics. Although the model

1. Nevertheless, it is instantiated in different ways. The multinomial model assumes that every occurrence of a word, including the multiple occurrences of the same word, is independent. On the contrary, the multiple Bernoulli model assumes that the occurrences of different words are independent. The multiple-Bernoulli model makes a weaker independence assumption, but this is at the price of not being able to model multiple occurrences. Empirically, there has been some evidence that Multinomial outperforms multiple-Bernoulli (Metzler et al., 2004) but not for all tasks (Losada and Azzopardi, 2008) thus more work is clearly required in this direction (Tao and Zhai, 2007). Moreover, we would like to stress that a multinomial language model naturally embodies the assumption of words as atomic units of information by structuring terms as disjoint events. Disjoint events can be represented as orthogonal dimensions in a vector space, as in the VSM. This strong probabilistic constraint is not required in the Bernoulli and the Poisson model because they only assume terms to be stochastically independent.

performs better than n-grams LMs, its underlying probabilistic space become less clear. Nallapati and Allan (2002) try to capture useful dependencies at sentence level. A document is divided into sentences and independence is hypothesized among sentences rather than among terms. The dependency amongst terms in a sentence is calculated by using the Jaccard coefficient. The joint distribution over the sentence is approximated by building a maximum spanning tree in which the nodes represent the terms and the link represents bigram dependencies.

In general, these approaches only saw marginal improvements over the unigram LM at the expense of greatly increased computational complexity. A possible reason is that the models generally assume that any sequential pair of query terms is in a dependence relation. A number of authors recur to syntactical parsing of queries in order to overcome this issue (Srikanth and Srihari, 2003; Chelba and Jelinek, 2000; Lee et al., 2006; Maisonnasse et al., 2007; Gao et al., 2004). Srikanth and Srihari (2003) use syntactical parsing in order to divide the query into span of words, called concepts. They assume independence between concepts but bigram dependence between the words constituting query concepts. Gao et al. (2004) formulate a general dependence model retaining bigram and biterm models as special cases [2]. They posit the existence of a hidden linkage that represents the dependence between query terms by means of an acyclic planar graph. For a document to be retrieved, it has to contain not only the query words as in the classical bag-of-words model, but also the dependencies between query terms. The authors come to the interesting conclusion that the linguistic structure such as discovered by syntactic grammars does not probably reflect those dependencies that are needed for IR. New evidence supporting this conclusion has been provided by Bendersky et al. (2010).

2.1.3 Markov Random Fields

Despite the increased complexity of term dependencies models, the obtained improvements tend to be insignificant with respect to the unigram model, or at least not so significant as some other extensions solving exact matching problems by tackling synonymy and word relatedness, discussed next. Lavrenko (2004) advances that the information gained by integrating term dependencies may not be

2. This model has recently been reviewed by Maisonnasse et al. (2007). The authors give a simpler interpretation of (Gao et al., 2004), which must be preferred to the original one for it has shown to be equivalent in performance.

as large as it was initially thought. The Markov Random Field (MRF) approach for IR (Metzler and Croft, 2005) reports the first clear improvement for term dependencies models over bag-of-words baseline models. The MRF model is a general discriminative approach to ranking that exploits different types of evidence including proximity and n-grams occurrence. The evidence is combined in the scoring function in a log-linear way. Formally, the score of a document under the MRF model is given by:

$$s(Q, D) = \lambda_T \sum_{t \in Q} \log s_T(t, D) + \lambda_U \sum_{u \in Q} \log s_U(u, D) + \lambda_O \sum_{o \in Q} \log s_O(o, D),$$

where λ_T, λ_U and λ_B are the weights for the term score s_T, proximity (unordered) score s_U and n-gram (ordered) score s_O respectively. From a representational point of view, the MRF model marks an implicit turn-back towards the first VSM, where phrases, or unordered matches, are considered as additional dimensions and added to the final score. In our opinion, the success of MRF is due to multiple concurring factors: a) the interpolation coefficients λ are optimized towards the measure of interest, namely mean average precision (see Section 3.2.1); b) document weights are chosen using the frequency-based weight estimation principles inherited from language models, which provide robust performance, i.e. see (Zhai, 2008; Hazimeh and Zhai, 2015); c) the advent of large web collections, which highlighted the usefulness of capturing phrases. In Chapter 5, we obtain significative improvements over the MRF by switching to a truly new, more general representational framework.

2.1.4 Recent Term Dependency Models

Following the success of MRFs, a huge number of works have been dedicated in integrating proximity information of query terms in the document (Tao and Zhai, 2007; Cummins and O'Riordan, 2009; Lv and Zhai, 2009b; Svore et al., 2010; Cummins et al., 2010; Lu et al., 2014; Lu, 2015). Proximity can be seen as a "work-well-on-average" method that avoids the complexity to estimate the exact dependency between terms. Other methods made efforts to account for the importance of a phrase in characterizing the user information need (Shi and Nie, 2009, 2010; Cummins and O'Riordan, 2009; Song et al., 2008, 2009; Bendersky et al., 2010; Svore et al., 2010; Cummins et al., 2010; Bendersky and Croft, 2012; Hou

et al., 2013). Related to this line of work, Lioma et al. (2015) detect whether a phrase is compositional or non-compositional. Only non-compositional phrases are integrated into the MRF as useful term dependencies. Although not explored in this thesis, these methods may be applied to the model presented in Chapter 5. Eickhoff et al. (2015) present a new retrieval model based on statistical copulas. This model is principled and make use of sentence-level co-occurrence as a signal of dependence between terms. Similarly to the model presented in Chapter 5, this model doesn't heuristically mix a term score with a dependence score. Finally, a recent comparison of term dependencies models can be found in Huston and Croft (2014).

2.1.5 Quantum-Based Models

Most of the literature that has appeared since Van Rijsbergen's book (van Rijsbergen, 2004) tries to apply the new representation space offered by quantum theory (QT) to IR or to computational semantics. Widdows and Peters (2003) use vector spaces in order to model a geometry of word meaning. The authors represent word negation with the notion of orthogonality. By joining vector space and Boolean concepts, the model handles structured Boolean queries while providing a flexible framework for word sense disambiguation (WSD). A survey on vector space models of semantics is provided by Turney et al. (2010). Symonds et al. (2012) propose the Tensor Query Expansion (TQE), which uses ideas from the vector space model of meaning in order to mine candidate terms for query expansion. The method achieved interesting results on modest test collections but it is still unclear if it can handle current web datasets with millions of documents.

Melucci (2008) establishes a principled framework to address contextual information retrieval by bringing together vector space models and quantum probability. This work is motivated by the fact that an IR system should be context-aware thus providing an explicit representation of all the factors that could influence relevance computation, i.e. user preferences, location or search history. Documents are represented as vectors in the space and define a quantum probability distribution on the contextual factors, represented as subspaces. The author applies the model in a pseudo-relevance feedback scenario by building a relevance subspace. Although the approach being principled and quite general, documents representations do not exploit the full generality of the quantum probability space in the sense that

they are simply vectors and not density matrices. Nonetheless, this work can be seen as a precursor in the important questioning about what to observe from a document. The formalization of contextual factors is motivated by the need of observing more discriminative properties than single terms in order to characterize relevant documents. The work by Piwowarski et al. (2010) tests if acceptable performance for ad-hoc tasks can be achieved with a quantum approach to IR. Differently from Melucci (2008), the authors represent documents as subspaces and queries as density operators. The subspaces corresponding to the documents are estimated through passage-retrieval heuristics, i.e. a document is divided into passages and is associated to a subspace spanned by the vectors corresponding to the document passages. Different representations for the query density matrix are tested but none of them led to good retrieval performance. Successively, a number of works took inspiration from quantum phenomena in order to relax some common assumption in IR (Zuccon et al., 2010; Zhao et al., 2011). Zuccon et al. (2010) introduce interference effects into the Probability Ranking Principle in order to rank interdependent documents. Although this method achieves good results, it does not make principled use of the quantum probability space and cannot be considered as evidence towards the usefulness of the enlarged probabilistic space.

The intrinsic heuristic flavor in preceding approaches motivated some authors to provide evidence to the hypothesis that there exists an IR situation in which classical probabilistic IR fails and it is thus necessary to switch to a more general probabilistic theory. This is generally done by testing probabilistic invariants (Accardi, 1984) or Bell's inequality (Khrennikov, 2007). Melucci (2010) provides evidence for the necessity of adopting a quantum probability framework in IR by arguing that the "best" terms selected for query expansion show "non-classical" probabilistic behavior, in the sense that they violate Accardi's statistical invariant. Later, Melucci (2013) derives interesting links between the probability ranking principle and the problem of quantum detection. Similar attempts but in different contexts are the works by Bruza et al. (2009); Bruza and Cole (2005), Kitto et al. (2010), Aerts and Sozzo (2011) in computational semantics and cognitive science and by Busemeyer et al. (2011); Trueblood and Busemeyer (2011), Pothos and Busemeyer (2009) in decision theory. Cognitive science and human decision theory deal with user studies and consequently small datasets. Therefore, differently from the IR domain, computational issues are not of strict concern. Although being inspiring, it is not

clear how these studies are applicable to retrieval tasks and thus their utility for our purposes remains limited. Although not related to IR, the works by Tsuda et al. (2006), Warmuth and Kuzmin (2009) and Koolen et al. (2011) apply learning techniques to estimate density matrices. These works highly determined our general comprehension of the density matrix formalism. Finally, Melucci (2015) provides a recent review of the state-of-the-art in the domain.

2.1.6 Other Retrieval Models

In addition to the models described above, several other models and representations have been proposed in IR literature. Here we sketch a brief picture of some of them without providing all the details, as they are less related to our work. In the early Binary Independence Model (BIR) (Robertson and Jones, 1976) documents are represented as vectors of binary random variables, each corresponding to a word in the vocabulary. Therefore, an entry in the document vector is 1 if the corresponding word appears in the document and 0 otherwise. The model does not assign an explicit representation to the query but rather relies on the probabilities of each term given relevance, represented as a binary random variable. In this sense, it is difficult to integrate evidence about the query into the model. Robertson et al. (1983); Robertson (2003); Bodoff and Robertson (2004) try to overcome this problem by formulating a probabilistic model capable of integrating both query and document information. These works reinforce the intuition on how important it is to have separate document and query representations.

Overall, the BIR model had a profound impact on IR and generated a number of extensions, such as the Tree Dependence Model (van Rijsbergen, 1977) and the 2-Poisson Model (Robertson et al., 1980). The former tries to capture dependencies between document terms considered as conditionally independent in the BIR model (Cooper, 1995). The latter integrates information about the frequency of occurrence of a term by modeling the document as a vector of term frequencies drawn from a mixture of two Poisson distributions. The 2-Poisson model further inspired the successful BM25 weight scheme (Robertson, 2010) which is very simple to implement and does not suffer from data sparsity issues as previous models. The BM25 model can be viewed as a wise instantiation of the early vector-space model, i.e. it represents both documents and queries as vectors but chooses a

different weight scheme for terms occurring in documents and queries.

Turtle and Croft (1989) propose an inference network model for retrieval comprising document nodes, representation nodes, query nodes and a single information need node. Relevance is computed by applying the probabilistic inference while the link probabilities are estimated rather heuristically. Later, Metzler and Croft (2004) combine the inference network model with the statistical principles of the language modeling approach and give birth to the Indri search engine (Strohman et al., 2005), deploying the successful MRF model, presented in Section 2.1.3. We will not delve into the details of these models as they are less relevant to our work.

2.2 Representation Learning for IR

2.2.1 Matrix Factorization Methods

Latent Semantic Indexing The work on representation learning for IR starts with a generalization of the vector space model (GVSM) (Wong et al., 1985), in which documents and queries vectors pass through a linear transformation before computing the inner product, i.e. $s(Q, D) = \cos(A^T q, A^T d)$, where q, d are the document and query vectors and A is the linear transformation matrix. The matrix A contains information about term relationships. Each row of A is an embedding of a term in a latent space in which the orthogonality constraint between terms is relaxed. One question naturally arises about how to choose A. Deerwester et al. (1990) answer to this question a few years later by proposing the Latent Semantic Indexing (LSI) approach. LSI chooses A to be a low-rank approximation to the document-term matrix obtained by a truncated Singular Value Decomposition (SVD) [3]. The level of truncation corresponds to the dimensionality of the latent space. Synonymy is considered as semantic noise, masking the true concept behind the different word use. By reducing the dimensions of the term vector space, document and queries acquire a compressed representation that avoids the noise induced by synonymy thus becoming less prone to exact matching problems. Kontostathis and Pottenger (2006) show that the method also succeeds in capturing

3. An entry of the document-term matrix stores the frequency but also other weight functions can be used.

second order term co-occurrence information [4]. However, due to its complexity and low performance on standard TREC test collections (Atreya and Elkan, 2011), LSI has recently received modest attention from the IR community. In particular, Tipping and Bishop (1999) show that the method implicitly assumes that the entries of the document-term matrix are drawn from an isotropic Gaussian distribution. This may be harmful for count data and can thus undermine the effectiveness of the model.

Topic Models The application of dimensionality reduction in order to enhance latent patterns in the data stimulated a vast amount of research. Notably, it originated the topic modeling discipline. Its goal is to estimate latent dimensions, i.e. topics, which summarize well a collection of documents (Blei, 2012). In general, a topic is represented as a probability distribution over words, and documents are defined as combinations, probabilistic mixtures, of topic distributions. In this setting, a word is a vector in the topic space where each dimension is weighted by the conditional probability of belonging to that topic. Although being completely probabilistic, the link between topic models and linear algebra matrix factorization methods is strong (Ding et al., 2008). Among the best-known models, we cite the foundational works of Latent Dirichelet Allocation (LDA) (Blei et al., 2003) and Non-negative Matrix Factorization (Hoyer, 2004). In general, these models have shown to be very effective in IR ad-hoc tasks (Wei and Croft, 2006; Lu et al., 2010; Wang et al., 2013) as well as in a variety of other applications. In Sordoni et al. (2013), which is peripheral to this thesis, we modify the scoring function of (Wei and Croft, 2006) to account for dependencies between topics and achieved significant gains with respect to the baseline methods.

2.2.2 Neural Networks for IR

A huge amount of work has been published about neural networks and deep learning (see Goodfellow et al. (2016) for a recent account). The rise of neural network models in the NLP field notably began with the Neural Network Language Models (NLM) (Bengio et al., 2006). The authors first advanced the idea of explicitly learning word embeddings in order to boost the performance of statistical

4. A second order co-occurrence between two terms w_1 and w_3 holds when w_1 co-occurs with w_2 and w_2 co-occurs with w_3.

Language Modeling (LM) tasks. By exploiting word embeddings, it is possible to consider large n-gram contexts without suffering from data sparsity problems: semantic similarity between n-grams can be leveraged to achieve better generalization in predicting the next word. A notable amount of work followed these first approaches in order to lower their computational requirements, i.e. see (Morin and Bengio, 2005) or (Bengio, 2013) for a review. Recently, Mikolov et al. (2013) proposed the particularly successful Skip-Gram model offering fast unsupervised learning of word embeddings and accurate semantic resolution. Skip-Gram embeddings generated widespread interest from the community due to their versatility both as an effective out-of-the-box estimator and as a means to bootstrap more complex neural network models, i.e. (Serban et al., 2016; Severyn and Moschitti, 2015). The learning is completely unsupervised and exploits the distributional hypothesis, i.e. the embeddings of words that occur in similar contexts in the training corpus should be similar in the semantic space. In this sense, the estimation exploits local co-occurrence instead of global document-level co-occurrence. A global embedding estimation method has been proposed (Pennington et al., 2014). Levy and Goldberg (2014) shed light on the differences between local and global co-occurrence information and linked these embeddings models with the matrix factorization methods reviewed in the previous section. Overall, Skip-Gram determined the widespread use of word embeddings in a variety of fields.

The application of representation learning with neural networks is growing in IR. The Supervised Semantic Indexing (SSI) model (Bai et al., 2009) first practically exploited the idea of IR-oriented learning of embedding spaces. SSI learns a low-dimensional linear transformation by aligning Wikipedia documents to their titles. Differently from LSI, the estimation of the linear transformation is supervised and aims to minimize the ranking error. In Chapter 6, we contrast this method to our proposal. A work similar to SSI is the semantic hashing method, in which the representations are obtained through an auto-encoder Salakhutdinov and Hinton (2009). The Deep Semantic Structured Model (DSSM) (Huang et al., 2013) signed the first application of more complex neural network models for ad-hoc IR. The DSSM uses two deep neural networks, estimating the representation of a document and for a query. By exploiting proprietary query logs, the networks are trained to maximize the similarity between a user query and the clicked document title. Differently from SSI, documents and queries representations are obtained by a highly

non-linear transformation. However, the input of the deep forward network is still a bag-of-words. Shen et al. (2014) relax the bag-of-word assumption by employing a convolutional structure and propose the convolutional DSSM (CDSSM). The DSSM model has been used by Mitra (2015) to encode query similarities in the context of query auto-completion. Grbovic et al. (2015) embed queries using the Skip-Gram model and exploit the semantic space for query rewriting in sponsored search. Severyn and Moschitti (2015) propose a deep convolutional neural-network to rank short text pairs where elements of the pair are sentences. Recently, Zheng and Callan (2015) used continuous word embeddings estimated with the Skip-Gram model to determine the weight of query terms in a classical LM retrieval model.

3 Evaluation

This chapter describes the evaluation framework used throughout the thesis. Section 3.1 describes the anatomy of the datasets we used to conduct the experiments. Section 3.2 presents the evaluation metrics, their properties in capturing different aspects of system performance and the statistical significance tests we used in the thesis. Section 3.3 describes the anatomy of a query log and the Wikipedia dump we used in our experiments.

3.1 TREC Corpora

The Text REtrieval Conference (TREC) [1] series produced the test corpora used in this thesis. The first edition of TREC was held in 1992. Since then, TREC supported the IR community with a variety of open test corpora that provide a means to automatic system evaluation and thus are a fundamental resource in the advancement of the research in the field. TREC corpora consist of document collections, test topics and their relevance judgments. Each year, a number of participants take part to a TREC competition by submitting the runs of their retrieval systems on a provided set of topics and a document collection. For each run, the top results from a set of runs are combined to form the pool of documents to judge for relevance. Next, TREC assessors judge the pool with either binary or graded relevance judgments. The documents not appearing in the pool are not considered relevant.

3.1.1 Document Collections and Topics

We employ diversified retrieval collections to extensively test the capabilities of our models. We use both newswire and web test corpora. The newswire cor-

1. http://trec.nist.gov/

Name	Content	# Docs	Topic Numbers
SJMN	Newswire	90,257	51-150
TREC7-8	Newswire	528,155	351-450
WT10g	Web	1,692,096	451-550
ClueWeb-B	Web	50,220,423	51-200

Table 3.1 – The TREC collections used for evaluation.

Grade	Label	Description
4	Nav	This page represents a home page of an entity directly named by the query; the user may be searching for this specific page or site.
3	Key	This page or site is dedicated to the topic; authoritative and comprehensive, it is worthy of being a top result in a web search engine.
2	HRel	The content of this page provides substantial information on the topic.
1	Rel	The content of this page provides some information on the topic, which may be minimal; the relevant information must be on that page, not just promising-looking anchor text pointing to a possibly useful page
0	Non	The content of this page does not provide useful information on the topic, but may provide useful information on other topics, including other interpretations of the same query
-2	Junk	This page does not appear to be useful for any reasonable purpose; it may be spam or junk

Table 3.2 – Graded relevance scale for ClueWeb-B.

pora SJMN and TREC7-8 contain news articles from different sources (e.g., San Jose Morning News, Financial Times or LA Times). WT10G and ClueWeb-B contain web pages. At the time of our experiments, ClueWeb09 was the largest Web collection available to the IR researchers containing approximately one billion documents. In this work, we only use the Category-B of the corpus which contains about 50 million documents with the highest crawl priority.

3.1.2 Relevance Judgments

The relevance judgments of a TREC corpus are binary, i.e. a document is either relevant or non-relevant, or graded. For newswire collections, binary judgments are

used. Web collections use graded relevance judgments ranging from -2 to 4. Table 3.2 describes the meaning of the graded relevance judgments. The graded scale better captures the usefulness of a document to the user query and enables the computation of the refined performance metrics we introduce next. Note that relevance grades can be promptly used to calculate binary effectiveness measures, i.e. we treat 1/2/3/4 as relevant and grades 0/-2 as non-relevant.

3.2 Metrics

One can compute the effectiveness of a retrieval system by comparing a set of ranked documents with their relevance grade. For example, it is possible to compute the standard precision/recall curves. However, especially in web search, users might only be interested in examining the result list until a certain cutoff k is reached. Therefore, most of the measures presented next are computed at different cutoffs k. In general, TREC evaluates at $k = 10, 20$.

3.2.1 Binary Metrics

P@k One of the simplest strategies to compute the effectiveness of a system is to measure the proportion of relevant documents up to the k-th position in the ranking list. This measure is called *precision* at k, abbreviated $P@k$, and is defined as:

$$P@k = \sum_{i=1}^{k} \frac{r(D_i)}{k}, \tag{3.1}$$

where $r(D_i) \in \{0, 1\}$ indicates the binary relevance score for document D_i in the returned ranked list.

AP The *average precision* takes into account both precision and recall by computing an average of the precision at different cutoffs k, k being chosen as the positions where recall increases, i.e. the positions of relevant documents. Formally, AP is defined as:

$$AP = \sum_{i:r(D_i)=1} \frac{P@i}{|\mathcal{R}|}, \tag{3.2}$$

where $|\mathcal{R}|$ is the number of relevant documents. AP can be considered as an approximation of the area under the precision/recall curve. Usually, average precision is computed over a ranked list of 1000 documents.

MAP The *mean average precision* is simply defined as the mean AP over the set of evaluation topics $\mathcal{T}$:

$$MAP = \sum_{t \in \mathcal{T}} \frac{AP(t)}{|\mathcal{T}|}, \tag{3.3}$$

MRR In tasks such that there is only one relevant document, simply computing the *reciprocal rank* of the relevant document is a reliable way to measure effectiveness of competing systems. Formally:

$$MRR = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{i_t}, \tag{3.4}$$

where i_t is the rank of the relevant document for topic t. We use this measure in Chapter 7 to evaluate the effectiveness of our query suggestion system.

3.2.2 Graded Metrics

NDCG@k The *normalized discounted cumulative gain* is a metric using graded relevance judgment which was first proposed by Järvelin and Kekäläinen (2002). The gain of a document in the result list is obtained by discounting its relevance score with the logarithm of its rank. The discounted gain of each document in the result list is accumulated to form the discounted cumulative gain (DCG):

$$DCG@k = \sum_{i=1}^{k} \frac{2^{r(D_i)-1}}{\log_2(i+1)}, \tag{3.5}$$

where $r(D_i)$ is the relevance grade of document D_i. The DCG depend on the relevance scores available for each topic. In order to normalize the performance scores across topics, one divides the DCG score by the DCG obtained by the ideal ordering of documents in the ranked list. Formally:

$$NDCG@k = \frac{DCG@k}{IDCG@k}, \tag{3.6}$$

where IDCG@k is the DCG of the optimal ranking list.

ERR@k The *expected reciprocal rank* is based on the cascade user browsing model Craswell et al. (2008) for which the user examines the result list and stops at the first document that satisfy the query. Differently from NDCG, which rewards ranked lists that show highly relevant documents in the top ranks, the ERR discounts the relevance score of a document if it appears after a highly relevant document. Formally:

$$ERR@k = \sum_{i=1}^{k} \frac{P_i}{i} \prod_{j=1}^{i-1} (1 - P_j) \tag{3.7}$$

where $P_i = \frac{2^{r(D_i)}-1}{2^{\max(r)}}$ is a normalized relevance grade and $\max(r)$ is the maximum relevance score.

3.2.3 Statistical Significance

Given two systems A and B, the goal of statistical significance testing is to bound the uncertainty on the difference of performance between the two systems. The null hypothesis is that the mean of the distribution of the differences in performance between the two systems A and B is zero, i.e. the two runs are identical. Then, the statistical significance test computes a p-value, which is the probability of obtaining the observed difference in performance, assuming that the null hypothesis is true. Low p-values reinforce the confidence of the experimenter in rejecting the null hypothesis, i.e. the difference between the two systems does not depend on inherent noise in the evaluation. Several statistical significance tests have been used in the IR literature. Wilcoxon signed rank test and the Student t-test are notorious examples. Following the recommendations of Smucker et al. (2007), we use both t-test and the two-sided Fisher randomization test with 25,000 permutations evaluated at $\alpha < 0.05$.

3.3 External Resources

Historically, information retrieval models benefited from the use of *external* resources which are independent of the document collection (Bendersky and Croft,

Anchor Text	Title of the linked page
disorder of neural development	neurodevelopmental disorder
genetics of autism	heritability of autism
agents that cause birth defects	teratology
embryonic stage	human embryogenesis

Table 3.3 – Each line of the anchor-log consists of an anchor text and the title of the link.

User ID	Query	Date	Time	#	URL
3,496,052	cat	2006-03-01	00:01:09		
3,489,966	tropical rain forest	2006-03-01	00:01:09		
1,270,972	head hunters	2006-03-01	00:01:10	2	www.headhunters.com
1,270,972	head hunters	2006-03-01	00:01:10	3	www.planetquake.com
2,622,800	singlesnet	2006-03-01	00:01:11	1	www.singlesnet.com
1,346,136	douglas county libary	2006-03-01	00:01:13		
465,778	google	2006-03-01	00:01:13	1	www.google.com

Table 3.4 – Each line of the AOL query log contains a user ID, the textual query, the date and time of submission, the position of the clicked document and its URL.

2012; Kotov and Zhai, 2012; Liu et al., 2014). In this thesis, we consider two typical sources of information to support our systems: the English Wikipedia dumps of July 8th, 2013 and the query log data of AOL released in 2006 (Pass et al., 2006).

In Chapter 6, we use the English Wikipedia dump to build an *anchor-log* to mine term relationships for query expansion. An anchor-log is a paired corpus in which the first entry is an anchor text and the second entry is the title of the page referenced by the anchor (see Table 3.4). For query reformulation purposes, Dang and Croft (2010) show that an anchor log can bring similar performance to a corpus composed of query and clicked document title extracted from real proprietary query log. Overall, our anchor log contains 14,358,573 pairs.

In Chapter 7, we exploit the AOL query log to build the training corpus for our query suggestion system. The AOL query log contains 32,777,610 queries submitted by 657,426 anonymous users between March 1st and May 31, 2006. The structure of AOL query log is reported in Table 3.4. Due to our requirements for query suggestion, we process the query log to carve out a *session-log*, a dataset in which each entry is a sequence of queries by the same user such that each two consecutive queries in the sequence have been submitted within a predefined time frame (usually

Query Sessions
lulac - second language acquisition theory
houston texas - newspaper houston chronicle - galena park independent school district
fox news - fox 16 news rock ar
walmart best buy - walmart circuit city

Table 3.5 – Each line of the session log is a sequence of queries that the same user submitted within a time frame, usually 30 minutes. In the table, we use the symbol "-" mark the end of each query.

30 minutes). An example of the session-log we use in our experiments is given in Table 3.5. Further statistics of the session log are reported in Chapter 7.

3.4 Summary

Each TREC corpus includes a document collection, a set of topics and a set of relevance judgments. In this thesis, we will use SJMN and TREC 7-8 as newswire collections and ClueWeb09-B and WT10G as web collections. In addition, we presented the graded and binary effectiveness measures we will recur to evaluate our systems. MAP, NDCG and ERR will be used in Chapter 5 and 6, while MRR will be used in Chapter 7. Finally, we introduced the Wikipedia anchor-log and our AOL session-log that will be employed respectively in Chapter 6 and 7.

4 Looking at Vector-Space and Language Models for IR using Density Matrices

Prologue

Article Details

Looking at Vector Space and Language Models for IR Using Density Matrices. Alessandro Sordoni, Jian-Yun Nie. *Proceedings of Quantum Interaction (QI '13)*, pp. 147-159.

Context

At the time we wrote this article, most of the literature investigating the relationship between Quantum Theory (QT) and Information Retrieval (IR) was dedicated to study wether purely quantum effects such as interference and entanglement could shed new light on classical IR problems (Sordoni et al., 2013; Zuccon et al., 2010; Melucci and Rijsbergen, 2011). A few works were focused to interpret existing IR models using the mathematical framework given by QT (Melucci, 2013). Similarly, we are interested in analyzing whether the mathematics of density matrices could be used to interpret and generalize well-known retrieval models.

Contributions

The main contribution of this article is to propose a representational analysis of the well-known Language Modelling (LM) and Vector Space Model (VSM) approaches under the unifying formalism of density matrices. Our analysis reveals that the two approaches are complementary and new retrieval models may be built upon this complementarity. Notably, this article provided ground for the development of the Quantum Language Model (QLM), a new retrieval model that will be presented in Chapter 5.

Glossary

We provide a short glossary of the concepts we will be using. These will be discussed in more details throughout the article.

Ket/Bra In quantum theory, complex vectors are represented using the Dirac's notation. Therefore, a column vector $u \in \mathbb{H}^n$, where $\mathbb{H}^n$ is a vector-space in n dimensions, is denoted using a *ket*, $|u\rangle$. Its transpose is denoted using a *bra*, $\langle u|$.

Dyad/Projector A dyad is a unit-rank projection matrix. Given a ket $|u\rangle$, the dyad $|u\rangle\langle u| \in \mathbb{H}^{n\times n}$ is a projector onto the ray $|u\rangle$.

Quantum State In quantum theory, the state of a system, i.e. a particle or, in the case of information retrieval, a document or a query, is described by a ket $|u\rangle$.

Superposition We say that a system is in a *superposition* of states if it is described by a ket $|s\rangle = \alpha|u\rangle + \beta|v\rangle$, where $|u\rangle, |v\rangle$ are the components of the superposition and α, β their weights. Superposition is peculiar in quantum theory and can be viewed as the system being both in state $|u\rangle$ and in state $|v\rangle$.

Density Matrix/Mixed State Unlike superposition, a *density matrix* expresses uncertainty about the actual state of the system. It is a symmetric, positive-definite matrix of trace equal to one. If a system is in state $|u\rangle$ with probability α and in state $|v\rangle$ with probability $\beta = 1 - \alpha$, it can be described by the density matrix $S = \alpha|u\rangle\langle u| + \beta|v\rangle\langle v|$. When the state of a system can be expressed by means of a density matrix, we say that the system is in a *mixed state*.

4.1 Introduction

Information Retrieval (IR) has nowadays become the focus of a multidisciplinary research, combining mathematics, statistics, philosophy of language and of the mind and cognitive sciences. In addition to these, it has been recently argued that IR researchers should be looking into particular concepts borrowed from physics. Particularly, it was first evoked in 2004 in Van Rijsbergen's pioneering manuscript

"The Geometry of Information Retrieval" (van Rijsbergen, 2004) that Quantum Theory principles could be beneficial to IR.

Despite Quantum Theory (QT) being an extremely successful theory in a number of fields, the idea of giving a quantum look to Information Retrieval could be at first classified as unjustified euphoria. However, the main motivation for this big leap is found in the powerful mathematical framework embraced by the theory which offers a generalized view of probability measures defined on vector spaces. Events correspond to subspaces and generalized probability measures are parametrized by a special matrix, usually called *density matrix* or *density operator*. From an IR point of view, it is extremely attractive to deal with a formalism which embraces probability and geometry, those being two amongst the pillars of modern retrieval models. Even if we believe that an unification of retrieval approaches would be out-of-reach due to the intrinsic complexity of modern models, the framework of QT could give interesting overlooks and change of perspective thus fostering the design of new models. The opening lines of van Rijsbergen (2004) perfectly reflect this interpretation: "It is about a way of looking, and it is about a formal language that can be used to describe the objects and processes in Information Retrieval". To this end, the last chapter of Van Rijsbergen's book is mainly dedicated to a preliminary analysis of IR models and tasks by means of the language of QT. Amongst others, the author deals with coordinate level matching and pseudo-relevance feedback.

Since then, the methods that stemmed from Van Rijsbergen's initial intuition provided only limited evidence about the real usefulness and effectiveness of the framework for IR tasks (Piwowarski et al., 2010; Zhao et al., 2011; Zuccon et al., 2011). Several proposed approaches took inspiration from the key notions of the theory such as superposition, interference or entanglement. In Zuccon et al. (2010), the authors use interference effects in order to model document dependence thus relaxing the strong assumption imposed by the probability ranking principle (PRP). An alternative solution to this problem has been proposed in Zhao et al. (2011), in which a novel reranking approach is proposed using a probabilistic model inspired by the notion of quantum measurement. In Piwowarski et al. (2010), the authors represent documents as subspaces and queries as density matrices. However, both documents and queries are estimated through passage-retrieval like heuristics, i.e. a document is divided into passages and is associated to a subspace spanned by

the vectors corresponding to document passages. Different representations for the query density matrix are tested but none of them led to good retrieval performance. In Sordoni et al. (2013), the authors work out an explicit interference formula in a topic model setting. Although improvements are obtained over the baseline model, the ad-hoc application of the interference formula does not provide solid evidence towards the usefulness of the theory itself.

In order to give a stronger theoretical status to QT as a necessary or more general theory for IR, some authors step back into more theoretical considerations exposing potential improvements achievable over state-of-the-art models (Widdows and Peters, 2003; Melucci, 2010, 2013; Piwowarski et al., 2012). In Melucci (2013), the author shows how detection theory in QT offers a generalization of the Neyman-Pearson Lemma (NPL), which is shown to be strictly linked to the PRP. Dramatic potential improvements could be obtained by switching to such more general framework. Widdows and Peters (2003) observed that the Vector Space Model (VSM) lacked a logic like the Boolean model. Through the formalism for quantum logic illustrated in Birkhoff and Neumann (1936), the author defines a geometry of word meaning by expressing word negation based on the notion of orthogonality. Recently, Melucci and Rijsbergen (2011) offered a comprehensive review of QT methods for IR along with some insightful thoughts about possible reinterpretations of general IR methods – such as LSI (Deerwester et al., 1990) – from a quantum point of view. This paper shares the main purpose of the latter works.

In the ending section of his book, Van Rijsbergen calls for a reinterpretation of the Language Modeling (LM) approach for IR by means of the quantum framework. To our knowledge, such an interpretation has not been presented yet in the literature and this work can be considered as a first attempt to fill this gap. We provide a theoretical analysis of both LM and the VSM approach from a quantum point of view. In both models, documents and queries can be represented by means of density matrices. A density matrix is shown to be a general representational tool capable of leveraging capabilities of both VSM and LM representations thus paving the way for a new generation of retrieval models. As a conclusion, we analyze the possible implications suggested by our findings.

4.2 Quantum Probability and Density Matrices

In QT, the probabilistic space is naturally encapsulated in a complex vector space, specifically a Hilbert space, noted $\mathbb{H}^n$. We adopt the notation $|e_1\rangle, \ldots, |e_n\rangle$ [1] to denote the standard basis vectors in $\mathbb{H}^n$. In QT, events are no more defined as subsets but as subspaces, more specifically as projectors onto subspaces. Given a ket $|u\rangle$, the projector $|u\rangle\langle u|$ onto $|u\rangle$ is an elementary event of the quantum probability space, also called *dyad*. A dyad is always a projector onto a 1-dimensional space. Generally, a unit vector $|v\rangle = \sum_i v_i |u_i\rangle$, $v_i \in \mathbb{H}$, $\sum_i |v_i|^2 = 1$, is called a *superposition* of the $|u_i\rangle$ where $|u_1\rangle, \ldots, |u_n\rangle$ form an orthonormal basis for $\mathbb{H}^n$.

A density matrix ρ is a symmetric positive semi-definite matrix of trace one. In QT, a density matrix defines the state of a system (a particle or an ensemble of particles) under consideration. Gleason's famous theorem (Gleason, 1957) ensures that a density matrix is the unique way of defining quantum probability measures through the mapping $\mu(|u\rangle\langle u||\rho) = \mathrm{tr}\,(\rho|u\rangle\langle u|)$. The measure μ ensures that $\forall |u\rangle, \mu(|u\rangle\langle u||\rho) \geq 0$. This is because, $\mathrm{tr}\,(\rho|u\rangle\langle u|) = \langle u|\rho|u\rangle \geq 0$ because ρ is positive semi-definite. Moreover, if $|u_1\rangle, \ldots, |u_n\rangle$ form an orthonormal system for $\mathbb{H}^n$, the probabilities for the dyads $|u_i\rangle\langle u_i|$ sum to one, i.e. they can be understood as disjoints events of a classical sample space. Given that $\sum_i |u_i\rangle\langle u_i| = I_n$, the identity matrix, we have $\sum_i \mathrm{tr}\,(\rho|u_i\rangle\langle u_i|) = \mathrm{tr}\,(\rho \sum_i |u_i\rangle\langle u_i|) = \mathrm{tr}\,(\rho) = 1$. Therefore, for orthogonal decompositions of the vector space [2], a quantum probability measure μ reduces to a classical probability measure.

Any classical discrete probability distribution can be seen as a mixture over n elementary points, i.e. a parameter $\theta = (\theta_1, \ldots, \theta_n)$, $\theta_i \geq 0$, $\sum_i \theta_i = 1$. The density matrix is the straightforward generalization of this idea by considering a mixture over orthogonal dyads [3], i.e. $\rho = \sum_i v_i |u_i\rangle\langle u_i|$, $v_i \geq 0$, $\sum_i v_i = 1$. Given a density matrix ρ, one can find the components dyads by taking its eigendecomposition

1. The Dirac notation establishes that $|u\rangle$ denotes a unit norm vector in $\mathbb{H}^n$ and $\langle u|$ its conjugate transpose.

2. In a more general formulation of the theory, a quantum probability measure reduces to a classical probability measure for any set $\mathcal{M} = \{M_i\}$ of positive operators M_i such that $\sum_i M_i = I_n$. The set $\mathcal{M}$ is called Positive-Operator Valued Measure (POVM) (Nielsen and Chuang, 2010). Therefore, the properties reported in this paper which apply to a complete set of mutually orthogonal projectors equally hold for a general POVM.

3. In general, the dyads in the mixture don't need to be orthogonal. However, in this case, the coefficients v_i cannot be easily interpreted as the probabilities assigned by the density matrix to each dyad.

and building a dyad for each eigenvector. We note such decomposition by $\rho = R\Lambda R^\dagger = \sum_{i=1}^{n} \lambda_i |r_i\rangle\langle r_i|$, where $|r\rangle_i$ are the eigenvectors and λ_i their corresponding eigenvalues. This decomposition always exists for density matrices (Nielsen and Chuang, 2010). Note that the vector of eigenvalues $\lambda = (\lambda_1, \ldots, \lambda_n)$ belongs to the simplex of classical discrete distributions over n points. If the distribution λ lies at a corner of the multinomial simplex, i.e. $\lambda_i = 1$ for some i, then the resulting density matrix consists of a single dyad and is called *pure state*. In the other cases, the density is called *mixed state*.

Conventional probability distributions can be represented by diagonal density matrices. In this case, a classical sample space of n points corresponds to the set of projectors onto the standard basis $\{|e_1\rangle\langle e_1|, \ldots, |e_n\rangle\langle e_n|\}$. Hence, the density matrix corresponding to the multinomial parameter θ above can be represented as a mixture, $\rho_\theta = \mathrm{diag}(\theta) = \sum_i \theta_i |e_i\rangle\langle e_i|$. As an example, the density matrix ρ_θ below corresponds to a classical probability distribution with $n = 2$, σ is a pure state and ρ is a general quantum density, a mixed state:

$$\rho_\theta = \frac{1}{2}|e_a\rangle\langle e_a| + \frac{1}{2}|e_b\rangle\langle e_b| = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

$$\sigma = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix},$$

$$\rho = \begin{pmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{pmatrix}.$$

4.3 Looking at Language Models

In the Language Modeling approach to IR, each document is usually assigned a unigram language model $\theta_d = (\theta_{d1}, \ldots, \theta_{dn})$, i.e. a categorical distribution over the vocabulary sample space $\mathcal{V}$ (of size n), $w \in \mathcal{V}$, $p(w|\theta_d) = \theta_{dw}$ (Zhai, 2007). A query is represented as a sequence of terms $\{q_1, \ldots, q_m\}$, sampled i.i.d. (independent and identically distributed) from the document model. The score for a document is obtained by computing the likelihood for the query to be generated

by the corresponding document model:

$$L(\{q_1, \ldots, q_m\}|\theta_d) = \prod_{i=1}^{m} p(q_i|\theta_d). \tag{4.1}$$

This scoring function is generally called Query Likelihood (QL). On the other hand, Kullback-Leibler (KL) divergence models can be seen as a generalization of QL models introduced in order to facilitate the use of feedback information in Language Modeling framework (Zhai, 2007). In KL-divergence models, both documents and queries are assigned to unigram language models. The score for a document is calculated as the negative query to document KL-divergence:

$$-\mathrm{KL}(\theta_q \| \theta_d) = -\sum_w \theta_{qw} \log \frac{\theta_{qw}}{\theta_{dw}}.$$

4.3.1 Query Likelihood View

As presented in Section 4.2, conventional probability distributions can be seen as diagonal density matrices. A straightforward quantum interpretation of the QL scoring function can be obtained by associating a diagonal density matrix to each document and consider a query as a sequence of dyads. Formally, we associate the vocabulary sample space to the orthogonal set of projectors on the standard basis, $\mathcal{E} = \{|e_1\rangle\langle e_1|, \ldots, |e_n\rangle\langle e_n|\}$. The density matrix ρ for a document is a mixture over $\mathcal{E}$ whose vector of weights corresponds to the parameters θ_d. Therefore, $\rho = \mathrm{diag}(\theta_d) = \sum_i \theta_{di}|e_i\rangle\langle e_i|$. It is straightforward to show that restricted to $\mathcal{E}$, μ_ρ generates the same statistics as $p(\cdot|\theta_d)$, i.e. $\forall w \in \mathcal{V}$:

$$\mu(|e_w\rangle\langle e_w||\rho) = \mathrm{tr}\left(\rho|e_w\rangle\langle e_w|\right) = \sum_i \theta_{di}\,\mathrm{tr}\left(|e_i\rangle\langle e_i||e_w\rangle\langle e_w|\right) = \theta_{dw} = p(w|\theta_d).$$

In the query likelihood view, the query is represented as an i.i.d. sample of word events. As word events correspond to projectors onto the standard basis, we represent a query as a sequence of i.i.d. [4] quantum events belonging to $\mathcal{E}$, $\{|e_{q_1}\rangle\langle e_{q_1}|, \ldots, |e_{q_m}\rangle\langle e_{q_m}|\}$. Therefore, the score for a document is computed by the

4. In quantum physics, the meaning of i.i.d. can be associated to the physical notion of measurement. If a density matrix ρ represents the state of a system, an i.i.d. set of m quantum events is obtained by performing a measurement on m different copies of ρ and by recording the outcomes.

following product:

$$L(\{|e_{q_1}\rangle\langle e_{q_1}|, \ldots, |e_{q_m}\rangle\langle e_{q_m}|\}|\rho) = \prod_{i=1}^{m} \mu(|e_{q_i}\rangle\langle e_{q_i}||\rho) = \prod_{i=1}^{m} p(q_i|\theta_d), \tag{4.2}$$

which indeed corresponds to the classical QL scoring function. However, we shall stress out an important point about the equation above. If the projectors included in the query sequence are mutually orthogonal (as above), the calculation above behaves as a proper classical likelihood, i.e. the sum of the likelihoods of all possible samples of length m is one. On the contrary, the product cannot be considered as a classical likelihood because quantum probabilities for arbitrary events does not need to sum to one. Further considerations on these issues will be made in Section 6.

4.3.2 Divergence View

The KL scoring function computes a divergence between a query language model θ_q and document language model θ_d. In QT, the KL-divergence is a special case of a more general divergence function acting on density matrices called Von-Neumann (VN) Divergence. Note $\rho = \sum_i \lambda_i |r_i\rangle\langle r_i|$, and $\sigma = \sum_i \zeta_i |s_i\rangle\langle s_i|$ the eigendecompositions of two arbitrary density matrices. In the following, the log function applied to a matrix refers to the matrix logarithm, i.e. the natural logarithm applied to the matrix eigenvalues, $\log \rho = \sum_i \log \lambda_i |r_i\rangle\langle r_i|$. The VN divergence writes as:

$$\mathrm{VN}(\rho\|\sigma) = \mathrm{tr}\left(\rho(\log\rho - \log\sigma)\right) = \sum_i \lambda_i \log \lambda_i - \sum_{i,j} \lambda_i \log \zeta_j |\langle r_i|s_j\rangle|^2.$$

This divergence quantifies the difference in the eigenvalues as well as in the eigenvectors of the two density matrices Tsuda et al. (2006).

In order to see how the classical KL retrieval framework is recovered, we assign a density matrix to the query very similarly to what has been done for a document. Precisely, ρ_q and ρ_d are diagonal density matrices such that $\rho_d = \sum_i \theta_{qi} |e_i\rangle\langle e_i|$ and $\rho_d = \sum_i \theta_{di} |e_i\rangle\langle e_i|$. As ρ_q (ρ_d) is diagonal in the standard basis, its eigenvalues

correspond to θ_q (θ_d), thus:

$$\mathrm{VN}(\rho_q \| \rho_d) = \sum_i \theta_{qi} \log \theta_{qi} - \sum_{i,j} \theta_{qi} \log \theta_{dj} |\langle e_i | e_j \rangle|^2 = \sum_i \theta_{qi} \log \frac{\theta_{qi}}{\theta_{di}},$$

which corresponds to the KL divergence. As conventional probability distributions correspond to diagonal density matrices, their eigensystem is fixed to be the identity matrix. Intuitively, KL divergence captures the dissimilarities in the way they distribute the probability mass on that eigensystem, i.e. by their eigenvalues.

4.4 Looking at the Vector Space Model

In this section, we are attempting to look at the VSM (Salton and Buckley, 1988) in a new way. In its original formulation, no probabilistic interpretation could be given because of the lack of an explicit link between vector spaces and probability theory (Wong and Yao, 1995). In the model, documents and queries are represented in the non-negative part of the vector space $\mathbb{R}^n_+$, where n is the number of terms in the collection vocabulary. In VSM, each term corresponds to a standard basis vector. The location of each object in the term space is defined by term weights (i.e. *tf*, *idf*, *tf-idf*) on each dimension. Similarity between documents and queries are computed through a vector similarity score $q^\top d$, where q, d are the vector representations of the query and the document. In Salton and Buckley (1988), the authors show that normalizing document vectors is important to reduce bias introduced by variance on document lengths. By normalizing both document vector and query vector, the similarity score reduces to the cosine similarity between the two vectors, which is an effective similarity measure (Zobel and Moffat, 1998). Denote $|q\rangle, |d\rangle \in \mathbb{R}^n_+$, the normalized ($\|\cdot\|_2$) query vectors. Documents can thus be safely ranked by decreasing cosine $\langle q|d\rangle \in [0, 1]$, which cannot be negative because the ambient space is $\mathbb{R}^n_+$. [5]

5. In this paper, we do not explicitly take into account situations in which the vectors could contain negative entries. For example, this could easily happen after the application of Rocchio's algorithm (Rocchio, 1971) in feedback situations or by reducing the dimensionality of the vector space by LSI (Deerwester et al., 1990). Besides the historically encountered difficulties in the interpretation of such negative entries (Hofmann, 2001), in these particular cases, the rank equivalence situations discussed here may not hold. We argue that ignoring these situations causes no

4.4.1 Query Likelihood View

In this interpretation of the VSM, each document is associated to a probabilistic "model" in the same spirit of the Language Modeling approach. We define a density matrix ρ for the document as $\rho_d = |d\rangle\langle d|$, which is a pure state, i.e. its mixture weights are concentrated onto the projector $|d\rangle\langle d|$. Note that this density matrix does not have a statistical meaning. It has been determined by merely normalizing heuristic weighing schemes and it cannot be related to a statistical estimators such as Maximum Likelihood (MLE).

A query can be represented as the quantum event corresponding to the subspace spanned by $|q\rangle$. This subspace naturally corresponds to the dyad $|q\rangle\langle q|$. Hence, a query can be seen as the sequence of quantum events of length one $\{|q\rangle\langle q|\}$. In this setting, the likelihood given the document model is calculated by:

$$L(\{|q\rangle\langle q|\}|\rho_d) = \mu(|q\rangle\langle q||\rho_d) = \mathrm{tr}(\rho_d|q\rangle\langle q|) = \mathrm{tr}(\langle q|d\rangle\langle d|q\rangle) = |\langle q|d\rangle|^2, \qquad (4.3)$$

The above calculation shows that the quantum "likelihood" assigned to the event $|q\rangle\langle q|$ by the density ρ_d is the square of the cosine similarity between the query and the document. When restricted to the non-negative domain, the square function is a monotonic, increasing transformation. This means that $\mu(|q\rangle\langle q||\rho_d) \overset{rank}{=} \langle q|d\rangle$, i.e. the two formulations lead to the same document ranking.

4.4.2 Divergence View

According to the original VSM, queries and documents should share the same representation and the scoring function should be a distance measure between these representations. In the previous formalization, this initial paradigm seems apparently lost. The following alternative quantum interpretation of the VSM is perhaps closer to the original vision of the model. We associate a density matrix both to the document and to the query. Specifically, those density matrices would be pure states, projectors onto the corresponding vectors, i.e. $\rho_d = |d\rangle\langle d|$, $\rho_q = |q\rangle\langle q|$. It turns out that computing the Fidelity measure (Nielsen and Chuang, 2010) between

harm to the generality of our conclusions on the need of an enlarged representation space.

density matrices produces a ranking function equivalent to cosine similarity:

$$\mathrm{Fid}(\rho_q, \rho_d) = \mathrm{tr}\left(\sqrt{\sqrt{\rho_q}\rho_d\sqrt{\rho_q}}\right) = \mathrm{tr}\left(\sqrt{|q\rangle\langle q|d\rangle\langle d|q\rangle\langle q|}\right) = |\langle q|d\rangle|\mathrm{tr}\left(\rho_q\right) = |\langle q|d\rangle|$$

obtained by noting that ρ_q is a projector thus $\sqrt{\rho_q} = \rho_q$, and $\mathrm{tr}\left(\rho_q\right) = 1$. As $|q\rangle, |d\rangle \in \mathbb{R}^n_+$, ranking by Fidelity measure is equivalent to ranking by cosine similarity, thus $\mathcal{F}(\rho_q, \rho_d) \overset{rank}{=} \langle q|d\rangle$.

4.5 A joint analysis

In this section, we will try to summarize the commonalities and the differences arising from the quantum formalizations of the two models given in the preceding sections. The following analysis is succinctly reported in Table 4.1. As a starting point, we shall note that the ambient space for both models is the Hilbert space $\mathbb{H}^n$, where n is the size of the collection vocabulary. Each standard basis vector $\mathcal{E} = \{|e_1\rangle, \ldots, |e_n\rangle\}$ is associated to a word event. Therefore, the vocabulary sample space corresponds to the set of projectors onto the standard basis vectors $\{|e_i\rangle\langle e_i|\}_{i=1}^n$.

4.5.1 Query Likelihood View

In query likelihood interpretations, the query is represented as a sequence of i.i.d. dyads. In the VSM, the sequence contains one dyad corresponding to the projector onto the query vector $\{|q\rangle\langle q|\}$. On the contrary, in the LM approach the sequence contains a dyad for each classical word event, i.e. $\{|e_{q_1}\rangle\langle e_{q_i}|\}_{i=1}^m$.

Besides the number of dyads included in the sequence, a major difference distinguishes the two formalizations. Contrary to probabilistic retrieval models such as LM, a query is not considered as a sequence of independent classical word events but as a single event and a particular kind thereof. The query event is a *superposition* of word events. This can be seen because the vector $|q\rangle$ can be expressed, up to normalization, as $|q\rangle = \sum_w f(w)\,|e_w\rangle$ where $f(w)$ is the weight for term w in the query vector. This kind of event cannot be expressed using set theoretic operations neither it has a clear classical probabilistic interpretation: it does not belong to $\mathcal{E}$

Query Likelihood View			
	Query	Document	Scoring
VSM	$\{\|q\rangle\langle q\|\}$	$\rho_d = \|d\rangle\langle d\|$	$\mu(\|q\rangle\langle q\|\|\rho)$
LM	$\{\|e_{q_1}\rangle\langle e_{q_1}\|, \ldots, \|e_{q_m}\rangle\langle e_{q_m}\|\}$	$\rho_d = \sum_w \theta_{dw} \|e_w\rangle\langle e_w\|$	$\prod_i \mu(\|e_{q_i}\rangle\langle e_{q_i}\|\|\rho_d)$
Divergence View			
VSM	$\rho_q = \|q\rangle\langle q\|$	$\rho_d = \|d\rangle\langle d\|$	$\text{Fid}(\rho_q, \rho_d)$
LM	$\rho_q = \sum_w \theta_{qw} \|e_w\rangle\langle e_w\|$	$\rho_d = \sum_w \theta_{dw} \|e_w\rangle\langle e_w\|$	$-\text{VN}(\rho_q \| \rho_d)$

Table 4.1 – Density-matrix based interpretation of query-likelihood and divergence view to retrieval. In the query-likelihood view the query is a "quantum" event (a dyad, or a collection thereof), a document is a general density matrix and the scoring is the quantum likelihood of the query event given the document matrix. In the divergence view, both documents and queries are density matrices and the scoring corresponds to a divergence defined on the manifold of density matrices.

thus it can only be justified in the quantum probabilistic space. Arguing further, we would say that, in the case of VSM, term weighting methods aim at estimating the "best" query event, i.e. the event which is the most representative for the information need of the user. Intuitively, if a single choice would be given to us on what to observe, we would rather be observing in the "direction" of important words in the query.

It follows from the considerations above that VSM creates query representations by accessing the whole projective space through appropriate choices of $f(w)$. On the contrary, LM "sees", and consequently can handle, only events from the classical sample space $\mathcal{E}$. However, the principled probabilistic foundations of the model give the flexibility of adding an arbitrary number of such events in the sequence, thus refining query representation [6]. In the next section, this kind of duality between VSM and LM approaches will be strengthened by analyzing the properties of the density matrices used in the two models.

Before continuing, we shall make one last consideration about the "likelihood" written in Eq. 4.2. This equation and its corresponding maximization algorithm have already been proposed by Lvovsky et al. (Lvovsky, 2003) in Quantum Tomog-

6. This is indeed the practice of Query Expansion (QE), see for example (Carpineto and Romano, 2012).

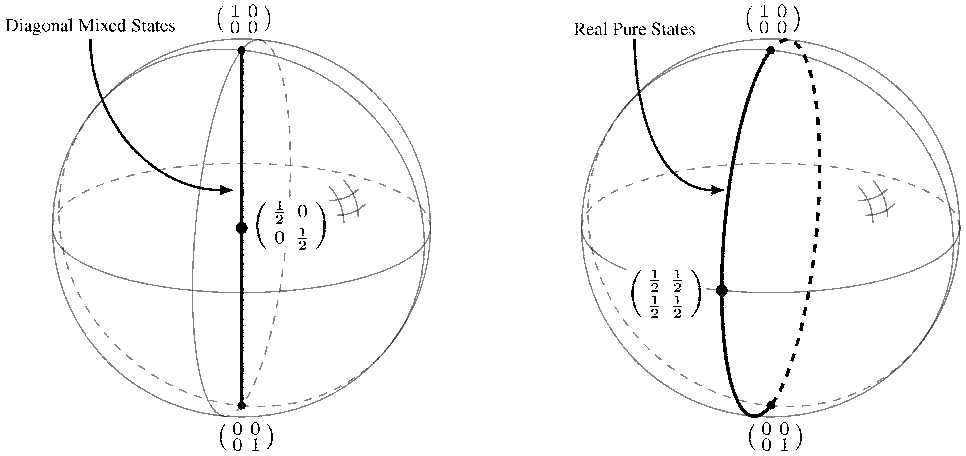


Figure 4.1 – The set $\mathcal{D}^2$ visualized using the Bloch sphere parametrization. We highlight in black the regions of $\mathcal{D}^2$ used by LM (to the left) and VSM (to the right).

raphy applications in order to achieve a Maximum Likelihood Estimation (MLE) of a density matrix. Eq. 4.2 reduces to a classical likelihood if and only if the projectors in the sequence are picked from the same eigensystem. Therefore, the product in its general form cannot be understood as a proper likelihood. We believe that it would be interesting to focus future research in finding a proper likelihood formulation in the quantum case that would enable principled statistical estimation and Bayesian inference (see Warmuth and Kuzmin (2009) for a recent attempt in formulating a Bayesian calculus for density matrices).

4.5.2 Divergence View

In the divergence view, a density matrix is associated both to the document and to the query and the scoring function is a divergence defined on the set $\mathcal{D}^n$ of $n \times n$ density matrices. Valuable insights can be provided by noting that the models gain access to different regions within $\mathcal{D}^n$. As an example, in Fig. 4.1, we plot the set $\mathcal{D}^2$ using the Bloch parametrization (Nielsen and Chuang, 2010). Highlighted in black are the regions of the space used by LM (left) and VSM (right). Distinct regions are likely to denote different representational capabilities.

In the case of LM, density matrices are restricted to be diagonal, i.e. mixtures over the identity eigensystem. For two density matrices to be different, one has to modify the distribution of the eigenvalues. Therefore, LM ranks based upon differences in the eigenvalues between density matrices. The picture of the VSM

approach appears as the perfect dual of the preceding situation. Query and documents are represented by *pure states*, i.e. dyads. Whatever the dimensionality of the Hilbert space, the mixture weights of these density matrices are concentrated onto a single projector. In order to be different, density matrices must be defined over different eigensystems. Therefore, VSM ranks based on the difference in the eigensystem between query and document density matrices.

The set of diagonal density matrices is represented in Fig. 4.1 (left). Any two antipodal points on the surface of the sphere correspond to a particular eigensystem. Diagonal density matrices are restricted to the identity eigensystem. However, they can delve inside the sphere by spreading the probability mass across their eigenvalues. The black circle in Fig. 4.1 (right) highlights pure states with real positive entries. These naturally lie on the surface of the Bloch sphere.

In summary, the VSM restriction to pure states leaves free choice on the eigensystem while fixing the eigenvalues. Conversely, by restricting density matrices to be diagonal, i.e. classical probability distributions, LM leaves free choice on the eigenvalues while fixing the eigensystem. Leveraging both degrees of freedom by employing the machinery of density matrices seems to be a natural step in order to achieve more precise representation for documents and queries. VSM and LM also differ in the choice of scoring functions. The former uses the Fidelity measure which is a metric on $\mathcal{D}^n$. The latter uses an asymmetric divergence on $\mathcal{D}^n$. More insights into these differences are given in the next section, where we try to contextualize our considerations by referring to common IR issues and concepts.

4.6 A Joint Interpretation and Perspectives

In Zhai (2007), the author presents KL divergence models as "essentially similar to the vector-space model except that text representation is based on probability distributions rather than heuristically weighted term vectors". The analysis done in the previous section extends this remark and highlights how VSM and LM leverage very different degrees of freedom by allocating different regions in $\mathcal{D}^n$. However, no clue is given about what should be the meaning of the eigensystems and the eigenvalues from an IR point of view, nor why controlling both could be useful for

IR. We will try to give some perspective for the potential usefulness of the enlarged representation space.

In basic bag-of-words retrieval models such as LM or VSM, terms are assumed to be unrelated, in the sense that each term is considered to be an atomic unit of information. To enforce this view, LM associates to each term a sample point and the VSM a dimension in a vector space. Our analysis showed that sample points correspond to dimensions in a vector space. The heritage left by LSI (Deerwester et al., 1990) suggests that a natural interpretation for such dimensions is to consider them as *concepts*. In this work, we interpret projectors onto directions as concepts. Because terms are considered as unrelated, the projectors onto the standard basis $|e_1\rangle\langle e_1|, \ldots, |e_n\rangle\langle e_n|$ in $\mathbb{H}^n$ form a *conceptual basis* in which each term labels its own underlying concept. [7]

From this point of view, LM builds representations of queries and documents by expressing uncertainty on which concept chosen from the standard basis represents the information need. On the contrary, VSM does not have the flexibility of spreading probability weights. However, it can represent documents and queries by a unique but arbitrary concept. In VSM, the similarity score is computed by comparing how similar the query concept is to the document concept. In this picture, the cosine similarity reveals to be a measure of relatedness between concepts. In LM, the score is not at all computed on concept similarity, but by considering how the query and the document spread uncertainty on the same conceptual basis.

In order to see how this all could be instantiated, let us suppose that compound phrases such as "computer architecture" express a different concept than "computer" and "architecture" taken separately. Modelling interactions between terms has been a longstanding problem in IR (Gao et al., 2004). We conjecture that a very natural way to handle such cases stems from our analysis. Assume that both "computer" and "architecture" are associated to their corresponding single term concepts, i.e. $|e_c\rangle\langle e_c|$, $|e_a\rangle\langle e_a|$. The concept expressed by the compound could be associated to a superposition event $|k_{ca}\rangle\langle k_{ca}|$ where $|k_{ca}\rangle = f(c)|e_c\rangle + f(a)|e_a\rangle$ and f is a weight function (assuming normalization) expressing how compound and single term con-

7. In Melucci (2008), *each* basis of a vector space is considered as describing a *contextual property* and the vectors in the basis as *contextual factors*. We prefer not to adopt such interpretation for two reasons: (1) in this paper, classical sample spaces are exclusively associated to orthonormal basis and (2) we believe that referring to concepts leads to a more general formulation, better tailored to our needs.

cepts are related. In this setting, the enlarged representation space turns out to be useful in order to express uncertainty on this set of concepts. The next step is to build a density matrix associated both to a query and to a document assigning uncertainty to both single term concepts $|e_c\rangle\langle e_c|, |e_a\rangle\langle e_a|$ and compound concepts $|k_{ca}\rangle\langle k_{ca}|$. This could be done, for example, by leveraging quantum estimation methods such as described in (Lvovsky, 2003). As we have pointed out before, the VN divergence could be the suitable scoring function in order to take into account both divergences in uncertainty distribution and concept similarities.

The considerations made so far did not necessitate of the whole machinery of complex vector spaces. We do not have a practical justification for the usefulness of vector spaces defined over the complex fields (see Zuccon et al. (2011) for a discussion on these issues). However, we speculate that these could bring improved representational power and thus remains an interesting direction to explore.

4.7 Conclusion

In this work, we showed how VSM and LM can be considered dual in how they allocate the representation space of density matrices and in the nature of their scoring functions. In our interpretation, VSM adopt a symmetric scoring function which measures the concept similarity. LM fixes the standard conceptual basis and scores documents against queries based on how they spread the probability mass on such basis. We argued that leveraging both degrees of freedom could lend a more precise representations of documents and queries and could be especially effective in modelling compound concepts arising from phrasal structures.

5 Modeling Term Dependencies with Quantum Language Models

Prologue

Article Details

Modeling Term Dependencies with Quantum Language Models for IR. Alessandro Sordoni, Yoshua Bengio and Jian-Yun Nie. *Proceedings of the 36th SIGIR conference (SIGIR '13)*, pp. 1319-1327.

Personal Contribution The ideas, the writing and the experiments in this article are my own. Yoshua Bengio provided general considerations to enhance the clarity of the article and gave several insights on possible future directions.

Context

The previous chapter provided us with the insight that new retrieval models may be built by means of a more general representational space. In this article, we present a new retrieval model that stems from the conclusions of the previous chapter. Our model estimates representations for documents and queries based on the presence of single terms and multi-word phrases. Until its formulation in (Metzler and Croft, 2005), the Markov Random Field Model (MRF) for IR held the state-of-the-art performance for term dependency models. MRFs consists in a scoring function mixing unigram scores with higher-order phrase scores. A major concern with this approach is that words and phrases are considered independently, i.e. phrases are associated to additional dimensions in the representation space. Clearly, phrases are not independent of their component words: as stated by Jones et al. (1998), "they represent a particular example of an extreme form of dependence between indexing units: the phrase *entails* the presence of single words". Differently from previous approaches, we propose to estimate the representations of documents

and queries by considering the interdependency between terms and phrases. We reproduce the article here as it appeared in SIGIR in its original form. In addition, we report the additional results obtained from our participation to the TREC 2013 Web Track conference (Sordoni et al., 2013).

Contributions

The contribution of this article is two-fold. First, we propose a novel application of quantum probability to IR and we show that significant improvements over a strong baseline bag-of-words model and a strong non bag-of-words model. Second, we show that the new phrase representation allows to specify the relationship between the phrase and its component terms. In our model, the phrasal information is not integrated in the scoring phase, but in the estimation phase.

Recent Developments

Our article can be considered as the first experimental evidence of the usefulness for IR of the mathematical framework of QT (Balkir et al., 2016). A number of extensions of the model have been proposed. For example, Li et al. (2015) proposes a Session-based Quantum Language Model (SQLM) that deals with multi-query session search task. Xie et al. (2015) uses QLMs to integrate term dependencies as quantum entanglements. This work has also provided ground for our next article, in which we will show how it is possible to learn, in an unsupervised way, the QLM projectors for each term (Sordoni et al., 2014).

5.1 Introduction

The quest for the effective modeling of term dependencies has been of central interest in the information retrieval (IR) community since the inception of first retrieval models. However, the gradual shift towards non bag-of-words models is strewn with modeling difficulties. One of the central problems is to find an effective way of representing and scoring documents based on such dependencies. As pointed out by Gao et al. (2004), dependencies can be handled in two ways.

The first approach is to extend the dimensionality of the representation space. In early geometrical retrieval models such as the Vector Space Model (VSM), dependencies arising from phrases (compound terms) are represented by defining additional dimensions in the space, i.e. both the phrase and its component single terms are regarded as representation features Fagan (1987); Mitra et al. (1997); Salton et al. (1974). For example, *computer architecture* is considered as disjoint from *computer* and *architecture*, which is a strong modeling assumption, and does not take advantage of the semantic relation that generally exists between a compound phrase and its component terms.

The second approach is more principled in such that simple terms are kept as representational units and term dependencies are modeled statistically as joint probabilities, i.e. $p(computer, architecture)$. Proposed dependence models such as n-gram Language Model (LM) for IR (Song and Croft, 1999), bi-term LM (Srikanth and Srihari, 2002) or the dependence LM (Gao et al., 2004) adopt such a representation. However, the gain from integrating dependencies was smaller than hoped (Zhai, 2007) and it came with higher computational costs due to dependency parsing or n-gram models (Lee et al., 2006; Song and Croft, 1999), or unsupervised iterative methods for estimating the joint probability (Gao et al., 2004).

Recently, non bag-of-words models such Markov random field (MRF) (Metzler and Croft, 2005), quasi-synchronous dependence model (Park et al., 2011) and the query hypergraph model (Bendersky and Croft, 2012) have been proposed. Most of these retrieval models take a log-linear form, which offers a very flexible way of taking into account term dependencies by integrating different sources of evidence, such as proximity heuristics and exact matching. However, the LM is used as a black box to estimate single-term and compound-term influences separately and then the model combines them to compute the final score. We believe that, from a representational point of view, these models have implicitly made a turn back to the first VSM approach in the sense that the dependencies are assumed to represent additional concepts, i.e. atomic units for the purpose of document and query representation, thus disjoint from the component terms (Bendersky et al., 2011; Bendersky and Croft, 2012). This choice indeed allows for flexible scoring functions. However, the retrieval model boils down to a combination of scores obtained separately from matching single terms and from matching compound dependencies. This is the main cause of the weight-normalization problem (Jones et al., 2000b;

Gao et al., 2004) which is that a dependency may be counted twice, as a compound and as component terms. In the context of phrases Jones et al. (2000b) state: "*the weight of the phrase should reflect not the increased odds of relevance implied by its presence as compared to its absence, as a whole unit, but the increased odds compared to the presence of its components words*". When integrating the evidence, the weights for the combination are usually estimated by optimizing a retrieval measure such as Mean Average Precision (MAP). In this sense, a principled probabilistic interpretation of these models is difficult.

The pioneering work by van Rijsbergen (2004) officially formalized the idea that Quantum Theory (QT) could be seen as a "*formal language that can be used to describe the objects and processes in information retrieval*". The idea of QT as a framework for manipulating vector spaces and probability is appealing. However, the methods that stem from this initial intuition provided only limited evidence about the usefulness and effectiveness of the framework for IR tasks. For example, Piwowarski et al. (2010) test if acceptable performance for ad-hoc tasks can be achieved with a quantum approach to IR. The authors represent documents as subspaces and queries as density operators. However, both documents and queries representations are estimated through passage-retrieval like heuristics, i.e. a document is divided into passages and is associated to a subspace spanned by the vectors corresponding to document passages. Different representations for the query density matrix are tested but none of them led to good retrieval performance. Successively, a number of works took inspiration from quantum phenomena in order to relax some common assumption in IR (Zhao et al., 2011; Zuccon et al., 2010). Zuccon et al. (2010) introduce interference effects into the Probability Ranking Principle (PRP) in order to rank interdependent documents. Although this method achieves good results, it does not make principled use of the quantum probability space and cannot be considered as evidence towards the usefulness of the enlarged probabilistic space. In general, these methods made heuristic use of the concepts of the theory and no clear probabilistic interpretation can be given.

The intrinsic heuristic flavor in preceding approaches motivated some authors to provide evidence to the hypothesis that there exists an IR situation in which classical probabilistic IR fails, or it is severely limited, and it is thus necessary to switch to a more general probabilistic theory (Warmuth and Kuzmin, 2009; Melucci and Rijsbergen, 2011; Melucci, 2013). Although these works are theoretically grounded

and heavily influenced our general vision of the theory, no clue is given on how to operationalize such results in real-world applications.

In this paper, we propose a novel retrieval framework for modeling term dependencies based on the probabilistic calculus offered by QT. In our model, both single terms and compound dependencies are mathematically modeled as projectors in a vector space, i.e. elementary events in an enlarged probabilistic space. In particular, a compound dependency is represented as a *superposition* event which is a special kind of projector that is neither disjoint from its component terms, nor a joint event. Documents and queries are represented as a sequence of projectors associated to a Quantum Language Model (QLM), encapsulated in a particular matrix. The scoring function is a divergence between query and document QLMs. We will show that our model is a generalization of classical unigram LMs. To our knowledge, this work can be seen as the first work to use the quantum probabilistic calculus in order to achieve improvements over state-of-the-art models.

5.2 A Broader View on Probability

5.2.1 The Quantum Sample Space

In quantum probability, the probabilistic space is naturally encapsulated in a vector space, specifically a Hilbert space, noted $\mathbb{H}^n$, but for the sake of simplicity, in this paper we limit ourselves to finite real spaces, noted $\mathbb{R}^n$. We will be using Dirac's notation restricted to the real field, for which a unit vector $u \in \mathbb{R}^n$, $\|u\|_2 = 1$ and its transpose $u^\top$ are respectively written as a *ket* $|u\rangle$ and a *bra* $\langle u|$. Using this notation, the projector onto the direction u writes as $|u\rangle\langle u|$. The inner product between two vectors writes as $\langle u|v\rangle$. Moreover, we note by $|e_i\rangle$ the elements of the standard basis in $\mathbb{R}^n$, i.e. $|e_i\rangle = (\delta_{1i}, \ldots, \delta_{ni})^\top$, where $\delta_{ij} = 1$ iff $i = j$.

Events are no more defined as subsets but as subspaces, more specifically as projectors onto subspaces (Nielsen and Chuang, 2010; Warmuth and Kuzmin, 2009). Given a 1-dimensional subspace spanned by a ket $|u\rangle$, the projector onto the unit norm vector $|u\rangle$, $|u\rangle\langle u|$, is an elementary event of the quantum probability space, also called a *dyad*. A dyad is always a projector onto a 1-dimensional space. Given the bijection between subspaces and projectors, it is correct to state that $|u\rangle$ is

itself an elementary event. For example, if $n = 2$, the quantum elementary events $|e_1\rangle = (1, 0)^\top$, $|f_1\rangle = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^\top$, can be represented by the following dyads:

$$|e_1\rangle\langle e_1| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, |f_1\rangle\langle f_1| = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}. \tag{5.1}$$

Generally, any ket $|v\rangle = \sum_i v_i|u_i\rangle$ is called a *superposition* of the $\{|u_i\rangle\}$ where $\{|u_1\rangle, \ldots, |u_n\rangle\}$ form an orthonormal basis. In order to see the generalization that is taking place, one has to consider that in $\mathbb{R}^n$ there is an infinite number of vectors even if the dimension n is finite. Hence, contrary to the classical case, an infinite number of elementary events can be defined.

5.2.2 Density Matrices

A quantum probability measure μ is the generalization of a classical probability measure such that (i) for every dyad $|u\rangle\langle u|$, $\mu(|u\rangle\langle u|) \in [0, 1]$ and (ii) it reduces to a classical probability measure for any orthonormal basis $\{|u_1\rangle, \ldots, |u_n\rangle\}$, i.e. $\sum_i \mu(|u_i\rangle\langle u_i|) = 1$. Gleason's Theorem (Gleason, 1957) states that, for any real vector space with dimension greater than 2, there is a one-to-one correspondence between quantum probability measures μ and *density matrices* ρ. The form of this correspondence is given by:

$$\mu_\rho(|v\rangle\langle v|) = \mathrm{tr}\,(\rho|v\rangle\langle v|). \tag{5.2}$$

A real density matrix is symmetric, $\rho = \rho^\top$, positive semidefinite, $\rho \geq 0$, and of trace 1, $\mathrm{tr}\,\rho = 1$ [1]. From now on, the set of $n \times n$ real density matrices would be noted $\mathcal{S}^n$.

By Gleason's theorem, a density matrix can be seen as the proper quantum generalization of a classical probability distribution. It assigns a quantum probability to each one of the infinite dyads. For example, the density matrix:

$$\rho = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, \tag{5.3}$$

assigns probabilities $\mathrm{tr}\,(\rho|e_1\rangle\langle e_1|) = 0.5$ and $\mathrm{tr}\,(\rho|f_1\rangle\langle f_1|) = 1$. Hence, the event

1. The trace is equal to the sum of the diagonal terms in a matrix.

$|f_1\rangle\langle f_1|$ is certain and still there is non-classical uncertainty on $|e_1\rangle\langle e_1|$. Only if $\{|u_1\rangle, \ldots, |u_n\rangle\}$ form an orthonormal system of $\mathbb{R}^n$ can the dyads $|u_i\rangle\langle u_i|$ be understood as disjoints events of a classical sample space, i.e. their probabilities sum to one. The relation that ties $|e_1\rangle\langle e_1|$ and $|f_1\rangle\langle f_1|$ is purely geometrical and cannot be expressed using set theoretic operations.

Any classical discrete probability distribution can be seen as a mixture over n elementary points, i.e. a parameter $\theta = (\theta_1, \ldots, \theta_n)$, where $\theta_i \geq 0$ and $\sum_i \theta_i = 1$. The density matrix is the straightforward generalization of this idea by considering a mixture over orthogonal dyads $\rho = \sum_i \upsilon_i |u_i\rangle\langle u_i|$ where $\upsilon_i \geq 0$ and $\sum_i \upsilon_i = 1$. Given a density matrix ρ, one can find the components dyads by taking its eigendecomposition and building a dyad for each eigenvector. We note such decomposition by $\rho = R\Lambda R^\top = \sum_{i=1}^n \lambda_i |r_i\rangle\langle r_i|$, where $|r_i\rangle$ are the eigenvectors and λ_i their corresponding eigenvalues. This decomposition always exists for density matrices (Nielsen and Chuang, 2010).

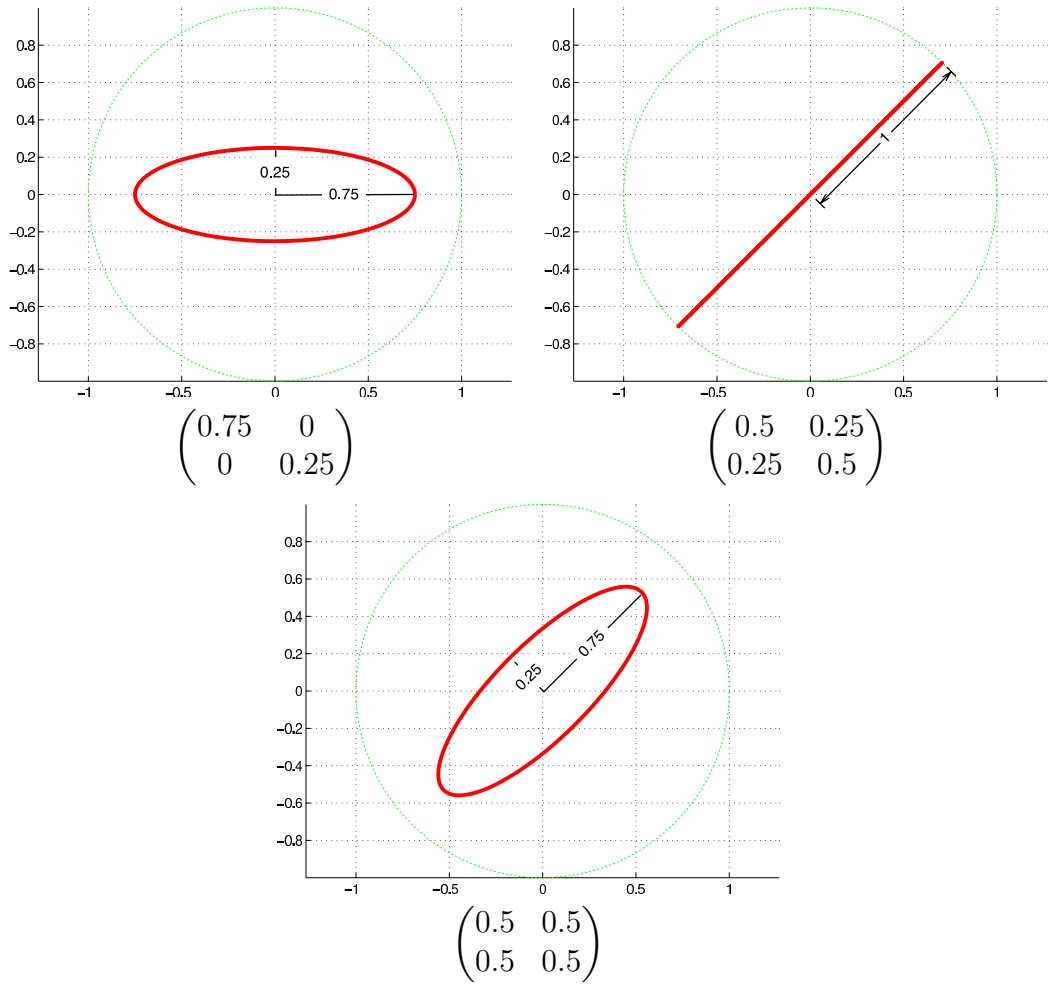
Conventional probability distributions can be represented by diagonal density matrices. The sample space corresponds to the standard basis $\mathcal{E} = \{|e_i\rangle\langle e_i|\}_{i=1}^n$. Hence, the density matrix corresponding to the parameter θ above can be represented as a mixture over $\mathcal{E}$, i.e. $\rho_\theta = \text{diag}(\theta) = \sum_i \theta_i |e_i\rangle\langle e_i|$. Consider a vocabulary of two terms $\mathcal{V} = \{a, b\}$. A unigram language model $\theta = (0.75, 0.25)$ defined on $\mathcal{V}$ is represented by:

$$\rho_\theta = \frac{3}{4}|e_a\rangle\langle e_a| + \frac{1}{4}|e_b\rangle\langle e_b| = \begin{pmatrix} 0.75 & 0 \\ 0 & 0.25 \end{pmatrix}.$$

Hence, term projectors are orthogonal, i.e. terms correspond to disjoint events. For example, the probability of the term a is computed by $\text{tr}\,(\rho_\theta |e_a\rangle\langle e_a|) = 0.75$. As conventional probability distributions are restricted to the identity eigensystem, they differ in their eigenvalues, which correspond to diagonal entries. On the contrary, general density matrices can differ also in the eigensystem. For example, the density matrix ρ of Eq. 5.3 has eigenvector $|f_1\rangle = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^\top$ with eigenvalue 1 and the eigenvector $|f_2\rangle = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})^\top$ with eigenvalue 0. Hence, it can be represented as a one-element mixture containing the projector $\rho = |f_1\rangle\langle f_1|$. When the mixture weights are concentrated into a single projector, the corresponding density matrix is called *pure state*. Otherwise, it is called *mixed state*.

When defined over $\mathbb{R}^n$, density matrices can be seen as ellipsoids, i.e. defor-

mations of the unit sphere (Figure 5.1) (Warmuth and Kuzmin, 2009). Classical probability distributions, i.e. diagonal density matrices, are ellipsoids stretched along the identity eigensystem. As quantum probability has access to an infinite number of eigensystems, the ellipsoid can be "rotated", i.e. defined on a different eigensystem. In this work, we will use this additional feature in order to build a more reliable representation of documents and queries taking into account more complex information than single terms.



$$\begin{pmatrix} 0.75 & 0 \\ 0 & 0.25 \end{pmatrix} \qquad \begin{pmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{pmatrix} \qquad \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

Figure 5.1 – The ellipses depict the action of the density matrices on the 2-D circle, i.e. $\{\rho|u\rangle : |u\rangle \in \mathbb{R}^2\}$. The eigenvalues of ρ define how much each ellipse is stretched along the corresponding eigenvectors. To the left, ρ corresponds to a classical probability distribution. To the center, a general density matrix for which we vary both the eigenvalues and the eigensystem. To the right, ρ is a pure state, thus the ellipse degenerates along the eigenvector corresponding to its unit eigenvalue.

5.3 Quantum Language Models

The approach Quantum Language Modeling (QLM) retains the classical Language Modeling for IR as a special case. Hereafter, we will present in details the quantum counterpart of unigram language models. Although it is not explicitly developed in this paper, we argue that arbitrary n-gram models could be modeled as well.

5.3.1 Representation

In classical bag-of-words language models, a document d is represented by a sequence of i.i.d. term events, i.e. $\mathcal{W}_d = \{w_i : i = 1, \ldots, N\}$, where N is the document length. Each w_i belongs to a sample space $\mathcal{V}$, corresponding to the vocabulary, of size n. It is assumed that such sequences correspond to a sample from an unknown distribution θ over the vocabulary $\mathcal{V}$, for which we want to gain insight.

A quantum language model assigns quantum probabilities to arbitrary subsets of the vocabulary. It is parametrized by an $n \times n$ density matrix ρ, $\rho \in \mathcal{S}^n$, where n is the size of the vocabulary $\mathcal{V}$. In QLM, a document d is considered as a sequence of M quantum events associated with a density matrix ρ:

$$\mathcal{P}_d = \{\Pi_i : i = 1, \ldots, M\}, \tag{5.4}$$

where each Π_i is a general dyad $|u\rangle\langle u|$ and represents a subset of the vocabulary. Note that the number of dyads M can be different from N, the total number of terms in the document. The sequence $\mathcal{P}_d$ is constructed from the observed terms $\mathcal{W}_d$: we have to define how to map subsets of terms to projectors. Separating the observed text from the observed projectors constitutes the main flexibility of our model. In what follows, we define a way of mapping single terms and arbitrary dependencies to quantum elementary events. Formally, we seek to define a mapping $m : \mathcal{P}(\mathcal{V}) \to \mathcal{L}(\mathbb{R}^n)$, where $\mathcal{P}(\mathcal{V})$ is the powerset of the vocabulary and $\mathcal{L}(\mathbb{R}^n)$ is the set of dyads on $\mathbb{R}^n$. As an initial assumption, we set $m(\emptyset) = \mathbb{O}$, where $\mathbb{O}$ is the projector onto the zero vector.

Representing Single Terms

In Section 5.2.2, we showed that unigram sample spaces can be represented as the set of projectors on the standard basis $\mathcal{E} = \{|e_i\rangle\langle e_i|\}_{i=1}^{n}$ and unigram language models can be represented as mixtures over $\mathcal{E}$, i.e. diagonal matrices. Therefore, a straightforward mapping from single terms to quantum events is:

$$m(\{w\}) = |e_w\rangle\langle e_w|, \tag{5.5}$$

where $w \in \mathcal{V}$. This choice associates the occurrence of each term to a dyad $|e_w\rangle\langle e_w|$, and these dyads form an orthonormal basis. Hence, occurrences of single terms are still represented as disjoint events. Consider $n = 3$ and $\mathcal{V} = \{$*computer*, *architecture*, *games*$\}$. If $\mathcal{W}_d = \{$*computer*, *architecture*$\}$ and one applies m to each of the terms, the sequence of corresponding projectors is $\mathcal{P}_d = \{\mathcal{E}_{computer}, \mathcal{E}_{architecture}\}$ where $\mathcal{E}_w = |e_w\rangle\langle e_w|$:

$$\mathcal{E}_{computer} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathcal{E}_{architecture} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{5.6}$$

Note that if we decide to observe only single terms, $\mathcal{P}_d$ turns out to be the quantum counterpart of classical observed terms $\mathcal{W}_d$, i.e. $M = N$.

Representing Dependencies

In this paper, by dependency, we mean a relationship linking two or more terms and we represent such an entity abstractly by a subset of the vocabulary, i.e. $\kappa = \{w_1, \ldots, w_K\}$. We define the following mapping for an arbitrary dependency κ:

$$m(\kappa) = m(\{w_1, \ldots, w_K\}) = |\kappa\rangle\langle\kappa|, \ |\kappa\rangle = \sum_{i=1}^{K} \sigma_i |e_{w_i}\rangle, \tag{5.7}$$

where the coefficients $\sigma_i \in \mathbb{R}$ must be chosen such that $\sum_i \sigma_i^2 = 1$, in order to ensure the proper normalization of $|\kappa\rangle$. The well-defined dyad $|\kappa\rangle\langle\kappa|$ is a *superposition* event. As we showed in Section 5.2.2, superposition events are justifiable only in the quantum probabilistic space. They are neither disjoint from their constituents $|e_{w_i}\rangle\langle e_{w_i}|$ nor do they solely constitute joint events in the sense of n-grams: here,

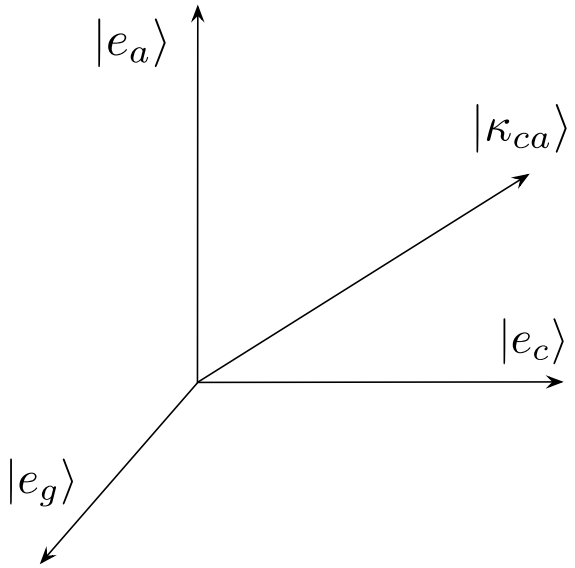


Figure 5.2 – The dependency κ_{ca} is modeled as a projector onto $|\kappa_{ca}\rangle$, i.e. as a superposition event.

the compound dependency is not considered as an additional entity, as done in previous models (Mitra et al., 1997; Metzler and Croft, 2005; Bendersky et al., 2011; Bendersky and Croft, 2012). The proposed mapping allows for the representation of relationships within a group of terms by creating a new quantum event in the same n-dimensional space.

In addition, superposition events come with a flexible way in quantifying how much evidence the observation of dependency κ brings to its component terms. This is achieved by changing the distribution of the σ_i: if one wants to attempt a classical interpretation, the σ_i can be viewed as relative pseudo-counts, i.e. observing $|\kappa\rangle\langle\kappa|$ adds fractional occurrence to the events of its component terms $|e_{w_i}\rangle\langle e_{w_i}|$. To our knowledge, until now this feature has been only modeled heuristically, or not modeled at all. In our framework, it fits nicely in the quantum probabilistic space by specifying how a compound dependency event and its constituent single terms events are related.

As an example, one could model the compound dependency between *computer* and *architecture*, $\kappa_{ca} = \{computer, architecture\}$, by the dyad $\mathcal{K}_{ca} = |\kappa_{ca}\rangle\langle\kappa_{ca}|$, where $|\kappa_{ca}\rangle = \sqrt{2/3}|e_c\rangle + \sqrt{1/3}|e_a\rangle$ (Figure 5.2). With respect to the example

taken above, the event is represented by the matrix:

$$\mathcal{K}_{ca} = \begin{pmatrix} \frac{2}{3} & \frac{\sqrt{2}}{3} & 0 \\ \frac{\sqrt{2}}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{5.8}$$

The superposition coefficients entail that observing $\mathcal{K}_{ca}$ adds more evidence to $|e_c\rangle\langle e_c|$ than to $|e_a\rangle\langle e_a|$.

Choosing When and What to Observe

Once we have defined the mapping m, one must ask three questions:

1. Which compound dependencies to consider?
2. When does such a compound dependency hold in a document?
3. When the compound dependency is detected, should we also consider the projectors for its subsets as observed events?

Regarding the first question, one may (a) use a dictionary of phrases or frequent n-grams, or (b) assume that any subset of terms that appear in short queries are candidate compound dependencies to capture. In this paper, we want to make the approach as independent as possible of any linguistic resource. So the second approach (b) is used. This will also allow us to make a fair comparison with the previous approaches using the same strategy, such as the MRF model (Metzler and Croft, 2005).

The second question regards whether such selected compound dependencies hold in a given document. In other words, one has to decide when to add the selected dependency projector into a document sequence $\mathcal{P}_d$. This can be done for example by assuming that the components terms in the dependency appear as a bigram in a document, as biterm or in a unordered window of L terms. Convergent evidence from different works (Bai et al., 2008; Lv and Zhai, 2009b; Metzler and Bruce Croft, 2007; Srikanth and Srihari, 2002; Zhao and Yun, 2009) confirms that proximity is a strong indicator of dependence. Therefore, in this work we choose to detect a dependency if its component terms appear in a window of length L.

The third question regards how to apply the mapping m and can be more easily understood by a practical example. Consider a document $\mathcal{W}_d = \{computer, architecture\}$ and a query $\mathcal{W}_q = \{computer, architecture\}$. Once the dependency

$\kappa_{ca} = \{computer, architecture\}$ has been detected in the document, i.e. the component terms appear next to each other, one can further decide:

1. to map only the dependency, i.e. $\mathcal{P}_d = \{\mathcal{K}_{ca}\}$,
2. to map both the dependency and the component terms, i.e. $\mathcal{P}_d = \{\mathcal{E}_{computer}, \mathcal{E}_{architecture}, \mathcal{K}_{ca}\}$.

These two choices are illustrated in Figure 5.3. The first choice is a highly non-classical one because it completely obfuscates the occurrence of the component terms. Nevertheless, it becomes a valid choice in our framework. Differently from classical approaches, the fact that we only consider a count for the compound *computer architecture* does not mean that we assume that the terms *computer* and *architecture* do not occur. The dependency event is not disjoint from the single term events, and its occurrence partially entails the occurrence of its component terms. However, this choice is more dangerous because it over-penalizes the component terms: we should know very precisely *when* such a strong dependency is observed and *which coefficients* to assign to it.

The second choice is implicitly done in current dependency models and is at the basis of the weight-normalization problem. From this point of view, the sequence $\mathcal{P}_d$ could be seen as composed by concepts as recently formalized by Bendersky and Croft (2012); Bendersky et al. (2011). However, there are crucial differences from that work: (1) we give a clear probabilistic status to such concepts and (2) we do not assume that concepts are atomic units of information, completely unrelated from each other. In classical dependence models, single terms and compound dependencies are scored separately and then the scores are combined together (Bendersky and Croft, 2012; Metzler and Croft, 2005; Zhai, 2007). A critical aspect of such models is that the occurrence of the phrase *computer architecture* will be counted twice - as single terms and as a compound. That is why the score on compound dependencies must be reweighed before integrating it with the independence score (Gao et al., 2004; Jones et al., 2000b; Metzler and Croft, 2005). Contrary to classical models, our model does not suffer from such a problem because the evidences brought by the compound dependency as a whole and by its component terms are integrated in the estimation phase. Even if not reported explicitly in the experiments section, conducted experiments show that including projectors for both the dependency and its subsets is much more effective for the ad-hoc task evaluated here and thus this strategy will be preferred throughout this paper. An

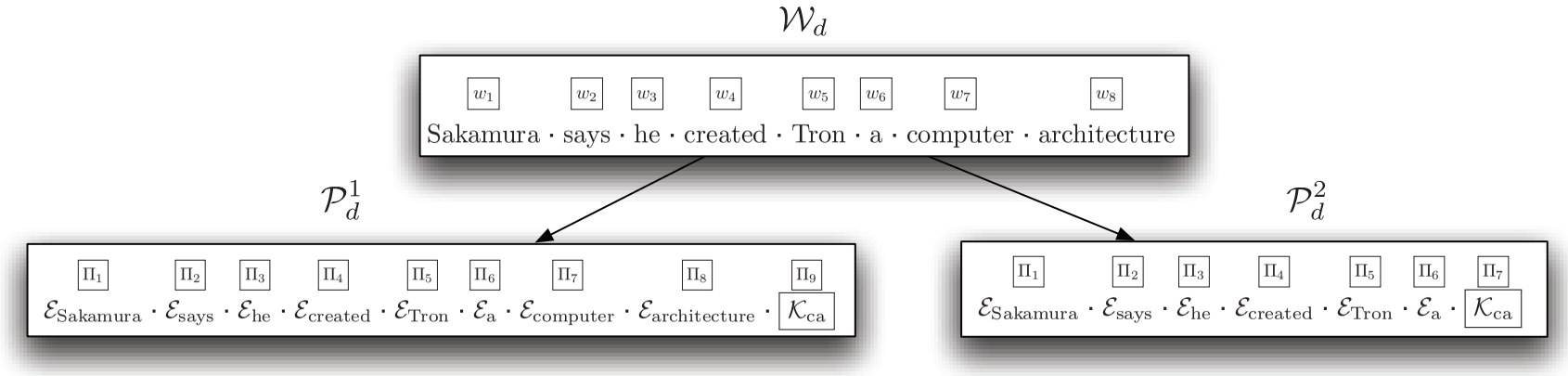


Figure 5.3 – Two possible quantum sequences $\mathcal{P}_d^i$ of an excerpt $\mathcal{W}_d$ from a TREC collection. The observation of *computer architecture* is associated to a superposition projector $\mathcal{K}_{ca} = |\kappa_{ca}\rangle\langle\kappa_{ca}|$ while $\mathcal{E}_w = |e_w\rangle\langle e_w|$ are classical projectors. For $\mathcal{P}_d^2$ we observed only the compound while in $\mathcal{P}_d^1$ we also added its subsets.

algorithm building the sequence of projectors from the document sequence will be presented in Section 5.4.3.

5.3.2 Estimation

Maximum Likelihood Estimation

Given that a document is represented by a set of observed projectors, one has to find ways to learn a quantum language model ρ to associate with a document. In QT, a number of objective functions have been proposed to estimate an unknown density matrix from a set of projectors: Linear Inversion (Nielsen and Chuang, 2010) and Hedged ML (Blume-Kohout, 2010) are notorious examples. In this work, we use the Maximum Likelihood (ML) formulation proposed by Lvovsky (2003), because (1) it can easily be seen as a quantum generalization of a classical likelihood function (2) contrary to linear inversion, ML generates a well-defined density matrix, i.e. $\rho \in \mathcal{S}^n$, and (3) proposed estimation methods remain computationally affordable in high-dimensional spaces.

Given the observed projectors $\mathcal{P}_d = \{\Pi_1, \ldots, \Pi_M\}$ for document d, we define as training criterion for the quantum language model ρ the maximization of the following product proposed in Lvovsky (2003) and corresponding in the unigram case to a proper likelihood:

$$L(\mathcal{P}_d|\rho) = \prod_{i=1}^{M} \operatorname{tr}(\rho\Pi_i). \tag{5.9}$$

The estimate $\widehat{\rho}$ can be obtained by approximately solving the following maximization problem:

$$\begin{aligned} \underset{\rho}{\text{maximize}} \quad & \log L(\mathcal{P}_d|\rho) \\ \text{subject to} \quad & \rho \in \mathcal{S}^n \end{aligned} \tag{5.10}$$

This maximization is difficult and must be approximated by using iterative methods. In Lvovsky (2003), the following iterative scheme is proposed, also called the "$R\rho R$ algorithm". One introduces the operator:

$$R(\rho) = \sum_{i=1}^{M} \frac{1}{\operatorname{tr}(\rho\Pi_i)}\Pi_i, \tag{5.11}$$

and updates an initial density matrix $\widehat{\rho}_{(0)}$ by applying repetitive iterations:

$$\widehat{\rho}_{(k+1)} = \frac{1}{Z} R(\widehat{\rho}_{(k)}) \widehat{\rho}_{(k)} R(\widehat{\rho}_{(k)}), \tag{5.12}$$

where, $Z = \mathrm{tr}\left(R(\widehat{\rho}_{(k)}) \widehat{\rho}_{(k)} R(\widehat{\rho}_{(k)})\right)$ is a normalization factor in order to ensure that $\widehat{\rho}_{(k+1)}$ respects the constraint of unitary trace Lvovsky (2003). Despite the $R\rho R$ algorithm being a quantum generalization of the well-behaving Expectation Maximization (EM) algorithm, the likelihood is not guaranteed to increase at each step because the nonlinear iteration may overshoot, similarly to a gradient descent algorithm with a too big step size. Characterizing such situations still remains an open problem (Řeháček et al., 2007). In this work, in order to ensure convergence, if the likelihood is decreased at $k+1$, we use the following damped update:

$$\widetilde{\rho}_{(k+1)} = (1-\gamma)\widehat{\rho}_{(k)} + \gamma \widehat{\rho}_{(k+1)}, \tag{5.13}$$

where $\gamma \in [0, 1)$ controls the amount of damping and is optimized by linear search in order to ensure the maximum increase of the training objective [2]. As $\mathcal{S}^n$ is convex Nielsen and Chuang (2010), $\widetilde{\rho}_{(k+1)}$ is a proper candidate density matrix. The process stops if the change in the likelihood is below a certain threshold or if a maximum number of iterations is attained.

From an IR point of view, the *metric divergence* problem (Morgan et al., 2004) tells us that the maximization of the likelihood does not mean that the evaluation metric under consideration, such as mean average precision, is also maximized. In the experiments section, we address the two following questions from a perspective closer to IR concerns:

1. Which initial matrix $\widehat{\rho}_{(0)}$ to choose?
2. When to stop the update process?

As the estimation of a quantum document model requires an iterative process, one may believe that the complexity will make the process intractable. In Section 5.4.5, we provide an analysis of the complexity of the proposed computation, which will show that the process is quite tractable.

2. Similar damped updates were successfully used in Heskes (2002) to improve convergence and stability of the loopy belief propagation algorithm.

Smoothing Density Matrices

The ML estimation presented above suffers from a generalization of the usual zero-probability problem of classical ML, i.e. the estimator assigns zero probability to unseen data Zhai (2007). This is also referred to as the zero eigenvalue problem (Blume-Kohout, 2010). Bayesian smoothing for density matrices has not yet been proposed. Bayesian inference in the quantum setting has just started to be the subject of intensive research (Warmuth and Kuzmin, 2009). In this work, we propose to smooth density matrices by linear interpolation. If $\widehat{\rho}_d$ is a document quantum language model obtained by ML, its smoothed version is obtained by interpolation with the ML collection quantum language model $\widehat{\rho}_c$:

$$\rho_d = (1 - \alpha_d)\,\widehat{\rho}_d + \alpha_d\,\widehat{\rho}_c, \tag{5.14}$$

where $\alpha_d \in [0, 1]$ controls the amount of smoothing. As the set of density matrices $\mathcal{S}^n$ is convex, the resulting ρ_d is a proper density matrix. In this work, we assume that $\alpha_d = \frac{\mu}{(\mu+M)}$, which is the well-known form of the parameter for Dirichlet smoothing (Zhai, 2007).

5.3.3 Scoring

The flexibility of the Kullback Liebler (KL) divergence approach in keeping distinct query and document representations makes it attractive for a candidate scoring function in our new framework. The direct generalization of classical KL divergence was introduced in Umegaki (1962) and is called *quantum relative entropy* or *Von-Neumann* (VN) *divergence*. Given two quantum language models ρ_q and ρ_d for the query and a document respectively, our scoring function is the negative query-to-document VN divergence:

$$\begin{array}{ccc} -\mathrm{VN}(\rho_q \| \rho_d) & = & -\mathrm{tr}\left(\rho_q(\log \rho_q - \log \rho_d)\right) \\ & \overset{rank}{=} & \mathrm{tr}\left(\rho_q \log \rho_d\right), \end{array} \tag{5.15}$$

where log applied to a matrix denotes the matrix logarithm, i.e. the classical logarithm applied to the matrix eigenvalues. Rank equivalence is obtained by noting that $\mathrm{tr}\left(\rho_q \log \rho_q\right)$ does not depend on the particular document. Denote by $\rho_q = \sum_i \lambda_{qi} |q_i\rangle\langle q_i|$, $\rho_d = \sum_i \lambda_{di} |d_i\rangle\langle d_i|$ the eigendecompositions of the density

matrices ρ_q and ρ_d respectively. By substituting into the above equation, the scoring function rewrites as:

$$-\operatorname{VN}(\rho_q||\rho_d) \overset{rank}{=} \sum_i \lambda_{qi} \sum_j \log \lambda_{dj} \langle q_i|d_j\rangle^2. \tag{5.16}$$

Compared to a classical KL divergence, the additional term $\langle q_i|d_j\rangle^2$ quantifies the difference in the eigenvectors between the two models. Following the representation introduced in Section 5.2.2, the VN divergence compares two ellipsoids not only by differences in the "shape" but also by differences in the "rotation".

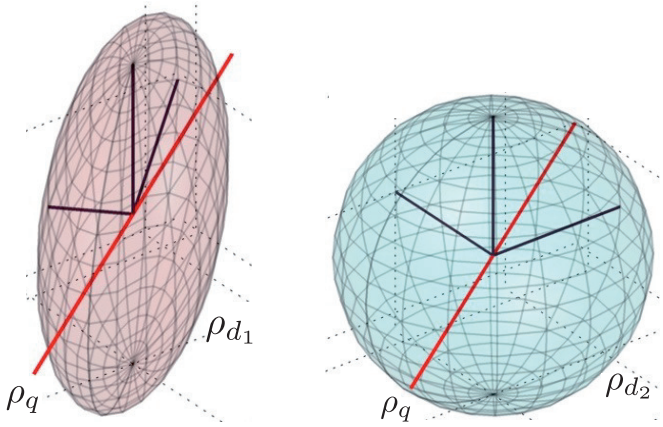
If a VSM-like interpretation is attempted, one can think about $\{|q_i\rangle\}$, $\{|d_j\rangle\}$ as semantic concepts for the query and the document respectively, whereas the vectors of eigenvalues λ_q, λ_d denote the importance of the corresponding semantic concepts in the two models. The VN divergence offers a way of matching query concepts by analyzing how much such concepts are related to documents concepts, i.e. $\forall i, j$, $\langle q_i|d_j\rangle^2$. Particularly, $\sum_j \langle q_i|d_j\rangle^2 = 1$. Thus, $\langle q_i|d_j\rangle^2$ can be interpreted as the quantum probability associated with the pure state $|q_i\rangle\langle q_i|$ for the elementary event $|d_j\rangle\langle d_j|$, i.e. $\mu_{q_i}(|d_j\rangle\langle d_j|) = \operatorname{tr}(|q_i\rangle\langle q_i|d_j\rangle\langle d_j|) = \langle q_i|d_j\rangle^2$. Hence, one could rewrite Eq. 5.16 as:

$$-\operatorname{VN}(\rho_q||\rho_d) \overset{rank}{=} \sum_i \lambda_{qi} \operatorname{E}_{\mu_{q_i}}[\log \lambda_d]. \tag{5.17}$$

Therefore, the VN divergence scores a document based on the expectation of how important concept $|q_i\rangle$ is in document d even if it does not appear in it explicitly.

5.3.4 Final Considerations

The estimation and scoring process of quantum language models retains classical unigram LMs and KL divergence as special cases. The classical unigram LM is recovered by restricting the maximization in Eq. 6.4 to diagonal density matrices and including into the sequence of projectors $\mathcal{P}_d$ only an orthonormal basis, such as the elements of $\mathcal{E}$. Classical KL divergence is recovered by noting that if ρ_q and ρ_d are diagonal density matrices, they share the same eigensystem. Hence, $|q_i\rangle = |d_i\rangle$ and $\lambda_{qi} = \theta_{qi}$, $\lambda_{di} = \theta_{di}$, where θ_q, θ_d are the parameters of classical unigram LMs for the query and the document respectively. In this setting, $\langle q_i|d_j\rangle^2 = 0$ for $i \neq j$



$-\mathrm{VN}(\rho_q \| \rho_{d_1}) \propto -.76$ $\quad$ $-\mathrm{VN}(\rho_q \| \rho_{d_2}) \propto -1.06$

o	$\mathcal{W}_\mathrm{o}$	$\mathcal{P}_\mathrm{o}$
q	$\{computer,\ architecture\}$	$\{\mathcal{E}_c, \mathcal{E}_a, \mathcal{K}_{ca}\}$
d_1	$\{computer,\ architecture,\ and,\ games\}$	$\{\mathcal{E}_c, \mathcal{E}_a, \mathcal{K}_{ca}, \mathcal{E}_g\}$
d_2	$\{computer,\ games,\ and,\ architecture\}$	$\{\mathcal{E}_c, \mathcal{E}_g, \mathcal{E}_a\}$

Figure 5.4 – A synthetic example of QLM with a vocabulary of $n = 3$ terms. The orthogonal rays are the eigenvectors of the ellipsoids. ρ_q is not smoothed thus degenerates onto a ray. ρ_{d_1} rotates towards the direction of observed query dependencies and is thus ranked higher.

and the VN divergence reduces to classical KL, i.e. $-\mathrm{VN}(\rho_q \| \rho_d) = -\mathrm{KL}(\theta_q \| \theta_d) \overset{rank}{=} \sum_i \theta_{qi} \log \theta_{di}$.

In Figure 5.4, we report a synthetic example of the application of the model. We plot the density matrices obtained by the MLE (Section 5.3.2) on the sequence of projectors reported in the table. As usual in ad-hoc tasks, we smooth only the QLMs of the documents. The model corresponding to the query is a projector, i.e. it has two zero eigenvalues, because we did not apply smoothing. If the dependencies are included in the sequence $\mathcal{P}_\mathrm{o}$, the MLE rotates the corresponding QLM towards the direction spanned by the observed projector (i.e. $\mathcal{K}_{ca}$). This entails that the model ρ_{d_1} is considered more similar to the query than the model ρ_{d_2} which corresponds to a classical language model.

Name	Content	# Docs	Topic Numbers
SJMN	Newswire	90,257	51-150
TREC7-8	Newswire	528,155	351-450
WT10g	Web	1,692,096	451-550
ClueWeb-B	Web	50,220,423	51-200

Table 5.1 – Summary of the TREC collections used to support the experimental evaluation.

5.4 Evaluation

5.4.1 Experimental Setup

All the experiments reported in this work were conducted using the open source Indri search engine (version 5.3) [3]. The test collections used are reported in Table 5.1. We choose the collections in order to vary (1) the collection size and (2) collection type. This will produce a comprehensive test set in order to verify the properties of our approach. All the collections have been stemmed with the Krovetz stemmer. Both documents and queries have been stopped using the standard INQUERY stopword list. For all the methods, the Dirichlet smoothing parameter μ is set to the default Indri value ($\mu = 2500$). The optimization of all the other free parameters for the proposed model and the baselines is done using five-fold cross validation using coordinate ascent (Metzler and Bruce Croft, 2007) with mean average precision (MAP) as the target metric. The performance is measured on the top-1000 ranked documents. In addition to MAP, for newswire collections we report the early precision metric @10 (precision at 10) and for web collections with graded relevance judgements we report the recent ERR@10, which correlates better with click metrics than other editorial metrics (Chapelle et al., 2009). The statistical significance of differences in the performance of tested methods is determined using a two-sided Fisher's randomization test (Smucker et al., 2007) with 25,000 permutations evaluated at $\alpha < 0.05$.

3. http://www.lemurproject.org

5.4.2 Methodology

Our experimental methodology goes as follows. In a first step, we compare our QLM approach to a unigram Language Modeling baseline (denoted LM) based on Dirichlet smoothing (Zhai, 2007), which is a strong bag-of-words baseline. This comparison is done by assigning uniform superposition weights to each dependency κ, i.e. $\sigma_i = 1/\sqrt{|\kappa|}$, where $|\kappa|$ is the cardinality of κ (denoted QLM-UNI). This step has two main objectives: (1) to test if quantum probability can bring better performance than a standard bag-of-words model and (2) to test if uniform superposition weights are a reasonable baseline setting.

As a second step, we test the proposed model against the strong non bag-of-words MRF model, which has shown to be highly effective especially for large scale web collections (Metzler and Croft, 2005). We test the full dependence version of the model (denoted MRF-FD) which captures dependencies between all the query terms and thus is the most natural choice for a comparison with our model. However, MRF-FD exploits both proximity (#uw) and exact matching (#1). As our model only exploits proximity as an indicator of dependence, we also propose to test the variant MRF-FD-U, which is a MRF using only the proximity feature. This could provide interesting insights on how the models score based upon the same evidence.

Finally, we propose a slightly more elaborate version of our model (denoted QLM-IDF) in which the superposition weights are no more assumed to be uniform. Instead, we assign to each σ_i the normalized *idf* weight of the corresponding term w_i. The objective is to test if a more reasonable parametrization of superposition weights can improve the retrieval effectiveness.

All the results exposed in this paper have been obtained by reranking. We rerank a pool of 20000 documents retrieved using LM in order to make a fair comparison between our method and the baselines.

5.4.3 Setting up QLM

Building the Sequence of Projectors

Very similarly to MRF-FD, given a query $\mathcal{Q} = \{q_1, \ldots, q_n\}$, we assume that the interesting dependencies to consider correspond to the power set $\mathcal{P}(\mathcal{Q})$ [4]. In order to build the set of projectors for the given document we apply Algorithm 1.

Algorithm 1 Builds the sequence $\mathcal{P}_d$ given $\mathcal{W}_d$, $\mathcal{Q}$

Require: $\mathcal{W}_d, \mathcal{Q}$
1: $\mathcal{P}_d \leftarrow \emptyset$
2: **for** $\kappa \in \mathcal{P}(\mathcal{Q})$ **do**
3: **for** $\#(\kappa, \mathcal{W}_d)$ **do**
4: $\mathcal{P}_d \leftarrow \mathcal{P}_d \oplus m(\kappa)$ *%Adds the projector to the sequence*
5: **end for**
6: **end for**
7: **return** $\mathcal{P}_d$

For each dependency κ in $\mathcal{P}(\mathcal{Q})$, the algorithm scans the document sequence $\mathcal{W}_d$. For each occurrence of κ, it adds a projector $m(\kappa)$ to the sequence $\mathcal{P}_d$. The function $\#(\kappa, \mathcal{W}_d)$ returns how many times the dependency κ is observed in $\mathcal{W}_d$. Therefore, the algorithm adds as many projectors as the number of detected compound dependencies. Note that by looping on $\mathcal{P}(\mathcal{Q})$, we are actually implementing the strategy exposed in Section 5.3.1, i.e. adding both the dependence and all of its subsets. Following Section 5.3.1, we choose to parametrize # as the unordered window operator in Indri (#uw$L$). Therefore, a given dependency κ will be detected if the component terms appear in any order in a fixed-window of length $L = l|\kappa|$. This kind of adaptive parametrization of the window length is state-of-the-art for dependence models such as MRF-FD (Bendersky and Croft, 2012; Metzler and Croft, 2005). For all the dependence models, the coordinate ascent for l spans $\{1, 2, 4, 8, 16, 32\}$, which is a robust pool covering different window lengths, including the standard value ($l = 4$) for MRF-FD.

4. In order to keep the retrieval complexity reasonable both for MRF and QLM, we limit ourselves to query term subsets with at most three terms.

MLE Convergence Analysis

Before doing any comparisons, we answer the questions related to the construction of a quantum language model, i.e. (1) how to initialize $\widehat{\rho}_{(0)}$? (2) when to stop the update process? In order to help the maximum likelihood process to converge faster, we initialize the matrix $\widehat{\rho}_{(0)}$ to the density matrix corresponding to the classical maximum likelihood language model θ^{ML} of the document or query under consideration. This is a diagonal matrix $\widehat{\rho}_{(0)} = \text{diag}(\theta^{ML})$. We also tested with the uniform density matrix, as suggested in Lvovsky (2003), but we found that the MAP was severely harmed.

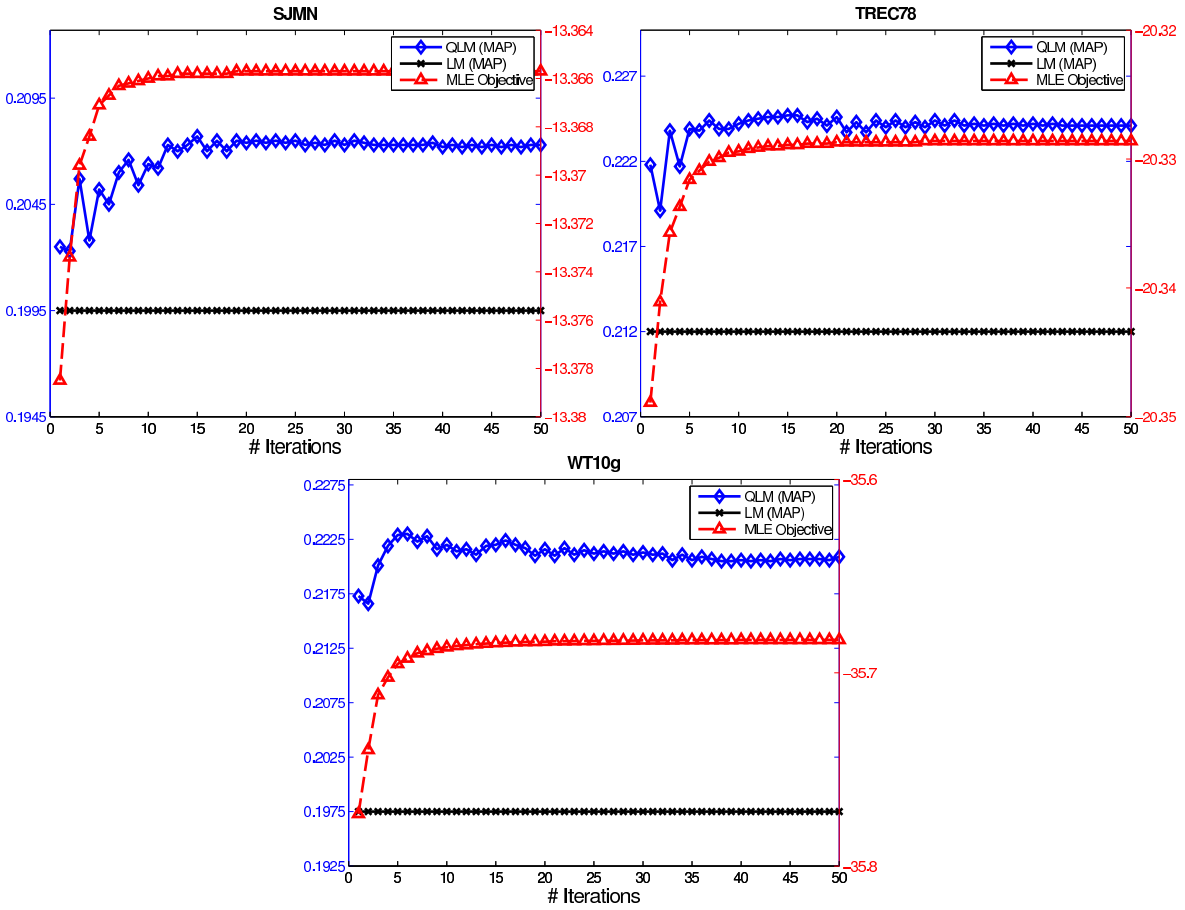


Figure 5.5 – Plots of MAP (QLM-UNI and LM) and MLE objective against the number of updates of the density matrix for SJMN, TREC7-8 and WT10g (left, right and bottom).

In order to address the second question, we analyze the variation of MAP with respect to the maximum number of iterations $n_{it} \in [1, 50]$. The damping factor γ is optimized over the set of values $\Gamma = \{0, 0.1, ..., 0.9\}$. The iterative process stops before n_{it} if the change in the likelihood is below 10^{-4}. In order to check for possible variations due to the collection type, we plot the iteration-MAP curve for two similar collections, i.e. SJMN and TREC7-8, and a web collection, WT10g. We also plot the training objective in Eq. 6.4 over the set of topics: $\frac{1}{|\mathcal{R}|} \sum_{d \in \mathcal{R}} \log \mathcal{L}_{\mathcal{P}_d}(\widehat{\rho}_d)$, where $\mathcal{R}$ is the multiset of retrieved documents. The trend is shown in Figure 5.5. Generally, at any number of iterations, the MAP stays significantly above the baseline. It seems that there is a good correlation between likelihood maximization and MAP, although one can note some overfitting at high number of iterations. Capping by $10 \leq n_{it} \leq 20$ seems a good trade-off between likelihood maximization and MAP. However, to provide a fair comparison with the baselines, we choose to include n_{it} as a free parameter to train by coordinate ascent.

5.4.4 Results

The results discussed in this section are compactly reported in Table 5.2 and 5.3.

Language Modeling Baseline

From the comparisons with the LM baseline, one can see that QLM-UNI outperforms LM significantly, with relative improvements in MAP going up to 12.1% in the case of WT10g collection and 19.2% for the ClueWeb-B collection (Table 5.3). This seems to be in line with the hypothesis formulated in Metzler and Croft (2005), for which dependence models may yield larger improvements for large collections.

The weight-normalization problem seems to be addressed automatically: our model does not need for any combination weights. Moreover, it is robust across the folds. From an analysis of the optimal values of the parameters obtained across the different folds, we found that optimal window sizes were $l \in \{1, 2\}$. This can be explained by considering that in the current version of QLM, it is possible to decide if the dependency is detected or not, but the model cannot discriminate its "importance". If one decides to increase l, more inaccurate dependencies will be detected and the performance will be deteriorated. However, even with a larger window size, statistical significance over LM is maintained. From these consider-

	SJMN		TREC7-8	
	P@10	MAP	P@10	MAP
LM	.3064	.1995	.4230	.2120
MRF-FD-U	.3138	.2071	.4350	.2228
MRF-FD	.3074	.2061	.4460	.2243
QLM-UNI	**.3181** *(+1.4/+3.5)*	.2077 *(+0.3/+0.8)*	**.4480** *(+3.0/+0.4)*	.2240 *(+0.5/-0.1)*
QLM-IDF	.3170 *(+1.0/+3.1)*	**.2093** *(+1.1/+1.6)*	.4450 *(+2.3/-0.2)*	**.2254** *(+1.2/+0.5)*

Table 5.2 – Evaluation of the performance for newswire collections shows that dependence models perform similarly. Numbers in parentheses indicate relative improvement (%) over MRF-FD-U/MRF-FD. All the results for dependence models are significant with respect to the baseline LM.

ations, we suggest $l = 2$ as a default setting for our model. Finally, the results endorse that our QLM does not need an engineered estimation of superposition weights to perform well.

Markov Random Fields Baseline

As a second test, we report the results obtained for the MRF-FD and MRF-FD-U baselines. These have proved to be very robust non bag-of-words baselines (Bendersky and Croft, 2012; Metzler and Croft, 2005). Contrary to our model, MRF does not handle dependency information in the estimation phase. One has to specify the coefficients $(\lambda_T, \lambda_O, \lambda_U)$ for the combination of dependence and independence scores. To limit per-fold overfitting, for the dependence models, we first train combination parameters ($\lambda_f \in \{0, 0.01, ..., 1\}$) then l for each fold. For MRF-FD-U, we set $\lambda_O = 0$.

Results show that for SJMN and TREC7-8, QLM-UNI, MRF-FD and MRF-FD-U are essentially equivalent (Table 5.2). However, for the two Web collections, our model significantly outperforms both MRF variants (Table 5.3). On ClueWeb-B, statistical significance is attained for the two reported measures. As conjectured in Metzler and Croft (2005), noisy web collections could be a more discriminative testbed for dependence models. Optimal l values for MRF-FD were very small for SJMN ($l \in \{1, 2\}$) in contrast to the optimal setting for ClueWeb-B ($l \in$

	WT10g		ClueWeb-B	
	ERR@10	MAP	ERR@10	MAP
LM	.1068	.1975	.0718	.1003
MRF-FD-U	.1136	.2097	.0828	.1103
MRF-FD	.1147	.2146	.0881	.1137
QLM-UNI	.1162 *(+2.2/+1.3)*	$.2215^{\alpha\beta}$ *(+5.6/+3.2)*	$\mathbf{.1015}^{\alpha\beta}$ *(+22.6/+15.2)*	$\mathbf{.1196}^{\alpha\beta}$ *(+8.4/+5.2)*
QLM-IDF	**.1176** *(+3.5/+2.6)*	$\mathbf{.2264}^{\alpha\beta}$ *(+7.9/+5.5)*	$.0997^{\alpha\beta}$ *(+20.4/+13.1)*	$.1189^{\alpha\beta}$ *(+7.8/+4.5)*

Table 5.3 – When applied to noisy web collections, the QLM variants achieve significant improvements over MRF counterparts. Numbers in parentheses indicate relative improvement (%) over MRF-FD-U/MRF-FD. All the results for dependence models are significant with respect to the baseline LM.

$\{16, 32\}$). In Metzler and Croft (2005), the authors suggest that for homogenous newswire collections a small window is enough to capture useful dependencies, while for large, noisy web collections, a larger span must be set. However, the performances obtained by our model seem to suggest that it can greatly benefit from term dependencies, on a variety of collections, even when a small window size is used. This elucidates the fact that even short range information can be extremely useful if integrated in the estimation phase. In order to get a more comprehensive view on such issues, we trained on the entire set of ClueWeb-B topics three versions of MRF-FD-U, each obtained by clamping a different value of $l \in \{1, 2, 4\}$. The best performing model obtained a MAP of 10.91. It seems that our model can exploit this short range information in a better way than MRF models.

Setting Superposition Weights

Our last test aimed at verifying if a more reasonable setting of the superposition weights could further improve retrieval performance. For a dependency $\{w_1, \ldots, w_K\}$, we set $\sigma_i = \sqrt{idf_{w_i} / \sum_i idf_{w_i}}$. This has the effect of attributing a larger count to the more "important" term in the dependency. QLM-IDF generally increases MAP. However, this is not the case for ClueWeb-B. From a query-by-query analysis, we noticed that QLM-IDF increases the performance for noisy queries by promoting the most "important" terms in unnecessary subsets. For multiword ex-

pressions such as ClueWeb-B topics *continental plates* and *rock art*, weighting by *idf* may be misleading by assigning more weight to one of the terms. In this cases, a uniform parametrization is far more effective. This demonstrates that there is still room for improvement by a clever tuning of superposition parameters, for example by leveraging feature functions (Bendersky and Croft, 2012; Bendersky et al., 2011).

5.4.5 Complexity Analysis

Complexity issues can be tackled by noting that it is not necessary to manipulate $n \times n$ matrices. We associate a dimension for each query term and an additional dimension for a "don't care" term that will store the probability mass for the other terms in the vocabulary. Therefore, a multinomial over n points is reduced to a multinomial over $|\mathcal{Q}| + 1$ points, where $|\mathcal{Q}|$ is the number of unique terms in the query and the additional dimension is simply a relabeling of the other term events. In this way, the QLM to manipulate is $k \times k$, where $k = |\mathcal{Q}| + 1$. The eigendecomposition generally requires $O(k^3)$. The iterative process requires at most $|\mathcal{P}(\mathcal{Q})| = 2^{|\mathcal{Q}|}$ matrix multiplications for the expectation step, where $2^{|\mathcal{Q}|}$ is the maximum number of *unique* projectors in $\mathcal{P}_d$ and 2 matrix multiplications for the maximization step. In the case the likelihood is decreased, $|\Gamma|$ more iterations are done giving a worst-case complexity of $O(n_{it}|\Gamma|2^k + k^3)$, i.e. if each iteration needs damping. We showed that $10 \leq n_{it} \leq 20$ is enough; we use $|\Gamma| = 10$ and k is very small for title queries, which make the process computationally tractable. In practice, we observed that the damping process is very effective and dramatically improves convergence speed. As an example, the mean number of iterations for ClueWeb-B when $n_{it} = 15$ is 7.02 which is orders of magnitude less than $n_{it}|\Gamma| = 150$. Finally, we conjecture that such process could be executed at indexing time, thus eliminating any additional on-line costs.

5.5 Conclusion

We presented a principled application of quantum probability for IR. We showed how the flexibility of vector spaces joined with the powerful tools of probabilistic

calculus can be mixed together for a flexible, yet principled account of term dependencies for IR. In our model, dependencies are neither represented as additional dimensions, nor stochastically as joint probabilities. They assume a new status as *superposition* events. The relationship of such an event to the traditional term events are encoded by the off-diagonal values in the corresponding projection matrix. Both documents and queries are associated to density matrices estimated through the maximization of a product, which in the classical case reduces to a likelihood. As our model integrates the dependencies in the estimation phase, it has no need for combination parameters. Experiments showed that it performs equivalently to the existing dependence models on newswire test collections and outperforms the latter on web data.

To our knowledge, this work provides the first experimental result showing the usefulness of this kind of probabilistic calculus for IR. The marriage between vector spaces and probability can be endlessly improved in the future. One straightforward direction is to relax the assumption that single terms represent orthogonal projectors. This could lead to a new way of integrating latent directions as estimated by purely geometric methods such as Latent Semantic Indexing (LSI) (Deerwester et al., 1990) into a probabilistic model. In this work, we did not exploit the full machinery of complex vector spaces. We do not have a practical justification for the use of the complex field for IR tasks. However, we speculate that this could bring improved representational power and thus remains an interesting direction to explore. At last, we believe that our model could be potentially applied to other fields of natural language processing only by means of a principled Bayesian calculus capable of manipulating density matrices. We hope that this work will foster future research in this direction.

5.6 QLMs in the TREC Web Track

To verify the effectiveness of the proposed retrieval approach over large-scale subsets of the Web, we took part to the 2013 TREC Web Track. The 2013 Web Track is composed of two tasks: the *ad-hoc* task tests the effectiveness of the proposed methods using standard retrieval metrics, such as ERR and NDCG; the novel *risk-sensitive* task aims at testing the robustness of the proposed methods in terms

of gains versus losses with respect to a baseline model provided by the organizers. Specifically, the risk-sensitive metric is described by the following formula:

$$URISK(Q) = \frac{1}{N}[\sum_{q \in Q^+} \Delta(q) - (\alpha + 1) \sum_{q \in Q^-} \Delta(q)], \tag{5.18}$$

where Q^+ (Q^-) is the set of queries for which the system improves (decreases) the baseline score, and α is the key risk-aversion parameter – i.e. the metric weights losses $\alpha + 1$ times as heavily as successes (Collins-Thompson and Voorhees, 2013).

5.6.1 Experimental setup

We use the english portion of the ClueWeb12 corpus (10^9 documents, Category A). We use the ClueWebB Web Track 2010, 2011 and 2012 queries to choose the parameters of our model. Both the index and the queries were stopped using the standard INQUERY stoplist and no stemming was performed. To demonstrate the efficiency of QLMs, all the retrieval experiments were performed using our modified version of Indri, with a built-in version of QLM, without recurring to reranking. For example, for the query usda food pyramid, we submit the following query expression to our modified version of Indri:

```
#q(usda food pyramid
   #uw2(usda food)
   #uw2(food pyramid)
   #uw2(usda pyramid)
   #uw3(usda food pyramid))
```

Notice that in QLM, no parameters are needed for the combination of unordered and single term scores. We further extended Indri's query language in order to run expanded queries. The modified syntax goes as follows:

```
#qweight(0.8 #q(usda food pyramid
                #uw2(usda food)
                #uw2(food pyramid)
                #uw2(usda pyramid)
                #uw3(usda food pyramid))
         0.2 #qweight( 0.8 health 0.2 nutrition ))
```

where health and nutrition are considered expansion terms with their respective probability weights.

Previously, we have shown that the QLM estimation process weakly suffers the metric divergence problem. Hence, we choose to avoid early-stopping and run the estimation algorithm until the improvement in likelihood between iterations drops below a threshold $\epsilon = 0.001$. Spam-filtering was applied on the entire ClueWeb12A corpus using the publicly available Waterloo Spam Ranking for the ClueWeb12 Dataset. We filter out the bottom 30% of the documents, as determined by the spam ranking. This threshold was found to optimize ERR@10 and NDCG@10 in our preliminary experiments with the ClueWebB queries. If compared to the standard TREC setting of filtering out the bottom 70% of the documents, our spam-filtering choice is more risk-inclined. However, we found that our model is quite robust to spam.

5.6.2 Query Expansion with QLM

Query expansion can be promptly introduced for the tasks addressed here. The idea is a straightforward generalization of query expansion in the classical LM framework: one smooths the original query model ρ_O with an expanded model ρ_L which is supposed to encode the *latent* aspects of the user information need and is simply obtained by selecting relevant terms in the top-K retrieved documents, for example using a Relevance Model (RM) (Lavrenko and Croft, 2001). The amount of smoothing is determined by a parameter λ as follows:

$$\rho_E = \lambda\, \rho_O + (1 - \lambda)\, \rho_L, \tag{5.19}$$

where ρ_E indicates the obtained expanded model. These operations are legit when manipulating density matrices because the set $\mathcal{S}_+^n$ is convex.

5.6.3 Description of the Runs

The description of the three runs is as follows:

— udemQlml1 is a "vanilla" run of QLM with the parameter settings described above. The purpose of this run was to evaluate the effectiveness of the retrieval approach on a single-pass batch retrieval setting.

Run	nDCG@20	ERR@20
TREC median	0.1738	0.098
udemQlml1	0.2286	0.1312
udemQlml1Fb	0.2074	0.1144
udemQlml1FbWiki	**0.2541**	**0.1515**

Table 5.4 – Summary of results for the TREC Web Track ad-hoc task.

— udemQlml1Fb performs query expansion using RM3 (Lv and Zhai, 2009a). We considered the top $K = 10$ retrieved documents obtained by udemQlml1 and set the smoothing parameter $\lambda = 0.8$.

— udemQlml1FbWiki performs query expansion using expansion terms from Wikipedia pages. To this end, we indexed the 2009 Wikipedia dump and performed a run of QLM. We extracted expansion terms from the top $K = 5$ retrieved documents and set the smoothing parameter $\lambda = 0.6$.

5.6.4 Ad-hoc Results

Table 5.4 compares the retrieval performance of these runs for the ad-hoc task. The expansion from the top-K retrieved documents from the Web collection fails to improve performance due to the noisy nature of the retrieved set. This result is in-line with past results trying to apply RM3 on Web collections (Lv and Zhai, 2009a). On the other hand, the expansion from Wikipedia pages has a significant positive impact on the retrieval performance for all the retrieval metrics reported. Wikipedia documents are less noisy and bear more useful feedback terms.

In Table 5.5, we report the comparison of the automatic runs submitted to the Web Track, ordered by ERR. Despite the simplicity of our run, which leverages only feedback from the entire Wikipedia pages and performs a single-pass retrieval over the whole ClueWeb index, our model performs consistently with respect to other participants in ERR, and perform similarly to the system ranked fourth. The first four systems either make use of complex reranking approaches (Technion), use learning-to-rank methods trained to maximize ERR (uogTr), use snippets of leading Web search engines (udel_fang) or parse the queries for entities exploiting external resources such as Freebase (ICTNET) (Bollacker et al., 2008). Interestingly, our method performs consistently also in NDCG, gaining three ranks and sitting in

third position.

We performed a query-by-query analysis to investigate which queries hurt our model the most. Two sources of performance loss may be identified: 1) selection of poor feedback terms and 2) selection of useless query term dependencies. An example of 1) is topic 234, "dark chocolate health benefits", for which Wikipedia expansion terms were overly generic ("brand", "found") or focused solely on the "chocolate" aspect of the query ("cocoa", "mars", "cocoavia"). For this topic, after query expansion, our model (udemQlml1Wiki) underperforms the organizers baseline. Instead, our non-expanded baseline QLM run (udemQlml1) achieves a relative improvement of 157% over the organizers baseline. The shortcomings concerning query expansion are well-known and may be addressed by explicitly penalizing overly common expansion terms (Metzler and Croft, 2007) and by explicitly diversifying expansion terms such that they cover all query aspects (Liu et al., 2014). On the contrary, we expect that the errors coming from capturing useless term dependencies be evident both in the expanded and non-expanded runs. An example is topic 216, "nicholas cage movies", in which the performance before expansion could be hurt by the useless term dependencies "nicholas movies" and "cage movies". A long line of works has already been dedicated in selecting important phrases and their weights in the user query, i.e. see Bendersky et al. (2011); Maxwell and Croft (2013). Although these strategies could also be applied to our model, our focus here was to provide experimental evidence of the effectiveness of QLMs in large-scale retrieval scenarios.

5.6.5 Risk-sensitive Results

QLM is a generalization of the LM approach for IR, which was used by the organizers to create the baseline run. Differently from existing term dependency retrieval models such as MRF, when a query phrase is not observed in a document, i.e. when the document contains only single query terms, then the QLM score corresponds exactly to the classical LM score. We expect this feature to bring increased robustness to our run when the performance is compared to the standard baseline LM. Table 5.6 show the ranking of systems for the risk-sensitive task. The systems are ordered by the performance of their best run obtained using Eq. 5.18 when $\alpha = 1$. Results show that our system gains two ranks with respect to the

Group	Run	ERR@10	NDCG@10
Technion	clustmrfaf	0.175	0.298
udel_fang	UDInfolabWEB2	0.167	0.284
uogTr	uogTrAIwLmb	0.151	0.247
ICTNET	ICTNET13RSR2	0.150	0.241
ut	ut22exact	0.149	0.224
diro_web_13	**udemQlml1FbWiki**	**0.143**[6]	**0.255**[3]
CWI	cwiwt13cps	0.121	0.211
webis	webisrandom	0.101	0.181
RMIT	RMITSC75	0.093	0.171
Organizers	baseline	0.088	0.162
UWaterlooCLAC	UWCWEB13RISK02	0.080	0.134

Table 5.5 – Ad-hoc results for automatic runs on Category A and ordered by ERR@10. We specify the rank of our run near the performance measure.

ad-hoc ranking showing robustness with respect to uogTr, ICTNET and ut runs.

Group	ERR@10	$\Delta, \alpha = 0$	$\Delta, \alpha = 1$	$\Delta, \alpha = 5$
Technion	0.175	0.087	0.076	0.033
udel_fang	0.167	0.078	0.059	-0.018
udel	0.150	0.061	0.047	-0.011
diro_web_13	**0.143**	**0.055**[6]	**0.034**[4]	**-0.051**[4]
uogTr	0.151	0.062	0.030	-0.101
ICTNET	0.149	0.060	0.028	-0.079
ut	0.144	0.056	0.025	-0.098
CWI	0.121	0.033	0.003	-0.115
Organizers	0.088	0.000	0.000	0.000
RMIT	0.093	0.005	-0.027	-0.156
webis	0.093	0.005	-0.029	-0.163
UWaterlooCLAC	0.080	-0.009	-0.040	-0.164

Table 5.6 – Ad-hoc results for automatic runs on Category A and ordered by Δ ERR@10 when $\alpha = 1$. We specify the rank of our run near the performance measure.

6 Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization

Prologue

In the previous chapter, queries and documents representations account for phrases without the need of artificially extending the representation space: documents and queries still lie in a "term space", i.e. the features of the representation are single terms. Considering terms as atomic descriptors has a major drawback: the decision on whether a relevance relation holds between a document and a query boils down to counting exact matches. The common assumption of orthogonality between terms clearly does not hold in practice because of the phenomena of word polysemy and synonymy and because of the contextual nature of meaning: for example, a query "car" would not match a document containing only the word "automobile" even if both words truly exhibit semantic commonalities and are likely to be equally correlated with relevance.

One strategy to overcome this problem is to change the representation features and embed documents, queries and terms themselves in a latent feature space. Feature learning techniques learn the feature space along with the representations of the objects of interest. The learning may be unsupervised as well as guided by the minimization of a specific task loss. Unsupervised feature learning has been known in IR since the inception of Latent Semantic Indexing (Deerwester et al., 1990). Instead, research on its supervised counterpart for IR remains relatively unexplored. The works we will present next employ the latter strategy in one way or another. In this chapter, we present an embedding method that learns a feature space given a labeled dataset of queries and relevant documents. The obtained semantic representations are used to perform query expansion, i.e. to artificially expand the user query with related terms in order to overcome the vocabulary mismatch occurring between documents and queries.

Article Details

Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization. Alessandro Sordoni, Yoshua Bengio and Jian-Yun Nie. *Proceedings of the 28th AAAI conference (AAAI '14)*, pp. 1586–1592.

Personal Contribution. The ideas, the writing and the experiments in this article are my own. Yoshua Bengio backed the theoretical considerations we put forward and provided guidance on how to perform cross-validation experiments with neural embeddings.

Context

At the time we wrote this article, very few supervised feature learning models were used for IR purposes. The most known were the Deep Structured Semantic Model (DSSM) of Huang et al. (2013) and the Supervised Semantic Indexing (SSI) (Bai et al., 2009). DSSM was used to perform retrieval of titles on a proprietary corpus. SSI was benchmarked solely on a custom Wikipedia retrieval task which may not be representative of standard TREC large-scale, open-domain retrieval tasks. This article exploits this research gap and brings additional evidence towards the usefulness of carefully estimated semantic word representations for IR purposes.

Contributions

This article presents a novel way of estimating word embeddings from a paired corpus and provides the first experimental evidence of the usefulness of the estimated representations for query expansion (QE). Our theoretical analysis shows that a particular class of word embedding models operating on paired corpora shares similarities with the relevance feedback method proposed in Rocchio (1971).

6.1 Introduction

Traditional information retrieval (IR) models consider terms as atomic units of information, disregarding the semantic commonalities and the complex syntactic relationships interweaving them in the discourse. One of the direct implications of this strong assumption is the *vocabulary mismatch*, i.e. a IR system could not retrieve documents which express the same query concepts using different linguistic expressions. For example, given a query *chevrolet trucks*, a document containing *chevy trucks* could be missed even if *chevrolet* and *chevy* are strictly related. A well-known, effective strategy to solve this issue is to perform query expansion (QE) (Carpineto and Romano, 2012), i.e. to expand the query by adding semantically related terms or compound concepts, which could be bigrams or longer phrases, i.e. *chevy* could be an important expansion term. In this setting, it is crucial to have a rich computational representation of the information need for valuable expansion terms to be mined.

The tradition of creating continuous word *embeddings* embodies the idea of folding sequences of terms into a "semantic" space capturing their topical content. Generally, a word embedding is a mathematical object associated to a word lying in a hidden high-dimensional semantic space equipped with a metric. The metric can naturally encode semantic or syntactic similarities between the corresponding terms. A typical instantiation is to choose a vector embedding for each term and estimate a similarity between terms in the latent space by taking the inner product of their corresponding embeddings (Deerwester et al., 1990). The meaning of this similarity highly depends on how the embeddings were obtained. Therefore, it is crucial to carve the semantic space for the task at hand using some task-specific training data (Bengio et al., 2006).

In this paper, we target at learning semantic representations of single terms and bigrams as a way to encode valuable semantic relationships for expanding a user query. Recently, a particularly successful way of selecting expansion terms was to use correlation and statistical translation models trained on aligned query / relevant document corpus obtained by memorizing users' clicks, i.e. *clickthrough* data. We believe that a careful structured latent space has several advantages over translation models. First, the information need has an explicit representation in the concept space, hence it is straightforward to ask questions about the most

similar terms given a query. Second, high-order term co-occurrences would be automatically captured, thus achieving better generalization. As a result of high-order co-occurrences, we automatically embed in the same space candidate terms both from relevant documents and similar queries without additional effort. Finally, using task-specific data, we learn the similarity function in such a way that query representations lie in a neighbourhood of relevant document terms, thus naturally increasing the likelihood of selecting good expansion terms. To our knowledge, the utility of semantic representations for query expansion purposes has not been investigated yet.

We propose a new model capable of learning, from clickthrough data, semantic representations for queries and arbitrary term or bigram concepts. Our model relies on the theoretical framework of the recently proposed Quantum Language Modeling (QLM) for IR (Sordoni et al., 2013). By employing such framework, our model embeds documents and queries in a larger space than single terms thus achieving higher semantic resolution without any computational fallout. This is in stark contrast to existing approaches, which use simple vectors as term and query representations. It is intuitive that text sequences should not lie in the same semantic space as single terms, as their informative content is higher. We will shed light on the theoretical implications of this enlarged representation space by analyzing our gradient updates. From an experimental standpoint, we show that this increased semantic resolution is important for query expansion purposes.

6.2 Related work

We briefly review the work which is close to this paper. We organize the related work in two subsections: query expansion approaches and semantic spaces.

6.2.1 Query expansion

Typical sources of query expansion terms are pseudo-relevant documents (Xu and Croft, 2000) or external static resources, such as clickthrough data (Cui et al., 2002; Gao et al., 2010; Gao and Nie, 2012), Wikipedia (Arguello et al., 2008) or ConceptNet (Kotov and Zhai, 2012). A classical model based on pseudo-relevant

documents was proposed by Rocchio for the SMART retrieval system (Rocchio, 1971). The new query vector is obtained by updating the original vector in the direction of the centroid of pseudo-relevant documents and far away from non-relevant ones. We will show that existing supervised embedding approaches perform similar embedding gradient updates. Our model performs a refinement of those updates.

Recently, attention turned towards static resources which allow to avoid multi-phase retrieval and noisy pseudo-relevant document sets. In particular, click-through data has shown great success as it can naturally bridge the gap from queries terms to documents terms. Recently, Gao and Nie (2012) and Gao et al. (2010) successfully performed QE by training a statistical translation model on clickthrough data and showed that it performed better than a standard correlation model (Cui et al., 2002).

6.2.2 Semantic spaces

In IR, the idea of using semantic term representations has been first put forward by the advent of LSI (Deerwester et al., 1990) and later by Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999), Non-Negative Matrix Factorization (Lee and Seung, 2000) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003). Although these models are usually referred to as *topic models*, they can be considered as implicitly learning semantic term representations from document co-occurrence statistics. Neural-Network Language Models (NLM) (Bengio et al., 2006) first advanced the idea of explicitly learning word embeddings in order to boost the performance of statistical Language Modeling tasks. A notable amount of work followed these first approaches in order to lower their computational requirements (Morin and Bengio, 2005; Mnih and Kavukcuoglu, 2013). Recently, Mikolov et al. (2013) proposed the particularly successful Skip-Gram word embedding model, combining fast learning and accurate semantic resolution. In general, very few embedding models have been used for IR purposes. The most known are the recent Deep Structured Semantic Model (DSSM) (Huang et al., 2013) and Supervised Semantic Indexing (SSI) (Bai et al., 2009). These models learn embeddings by exploiting clickthrough data and thus are related to our work. Both models try to learn an embedding structure so as to maximize the final objective function closely related

to retrieval. However, the scoring function and the representation paradigm are still inherited from the vector space model (VSM) approach (Salton et al., 1974) and thus differ from our approach: queries and documents are represented as weighted word vectors and then projected into a lower-dimensional vector space before taking their inner product. Our model can be seen as using a different scoring function and representation rationale which allow documents and queries to have a richer representation than single concepts. As our model shares many similarities with SSI, we will describe this method in more details in the next section.

6.3 Learning Concepts Embeddings

This section details our proposed approach for estimating latent concept embeddings. We recall the notions behind the SSI algorithm, shedding some light on its gradient updates rationale. This will facilitate the task in highlighting the major departures with respect to our model. In what follows, we assume that we dispose of a dataset $\mathcal{D} = \{(Q_l, D_l)\}_{l=1}^{L}$ composed of query / relevant document pairs. For all the presented models, the parameters to learn are the latent embeddings for each entry in a concept vocabulary $\mathcal{V}$, containing terms, bigrams or longer phrases, of size N. The unifying rationale of all the models is to represents concepts, documents and queries in a latent space in order to maximize a measure of similarity between Q_l and D_l.

6.3.1 Supervised Semantic Indexing

Representation

In SSI, the parameters to learn can be represented as a matrix $U \in \mathbb{R}^{K \times N}$, where K is the dimensionality of the latent embedding space and N the size of the vocabulary. Each $\kappa \in \mathcal{V}$ can be represented as a one-hot vector $x_\kappa = \{\delta_{1\kappa}, \ldots, \delta_{N\kappa}\}$, where $\delta_{ij} = 1$ iff $i = j$. In this way, the latent embedding of concept κ, $\tilde{x}_\kappa$, can be easily recovered by multiplying the parameter matrix by the one-hot representation, $\tilde{x}_\kappa = U x_\kappa$, $\tilde{x}_\kappa \in \mathbb{R}^K$. In other words, the latent embeddings $\tilde{x}$ are arranged in the columns of U, $U_{:\kappa} = \tilde{x}_\kappa$. Documents and queries are seen as unit-vectors in the vocabulary space, i.e. $q \in \mathbb{R}^N$, $\|q\|_2 = 1$, where for example q_κ will be the

frequency of occurrence κ^{th} concept in the query. The latent queries and documents are represented as linear combinations of concept embeddings which is the same rationale behind the LSI linear projection model:

$$\tilde{q} = Uq = Z_q^{-1} \sum_{\kappa \in Q} U x_\kappa = Z_q^{-1} \sum_{\kappa \in Q} \tilde{x}_\kappa, \tag{6.1}$$

where the sum is over all the concepts appearing in the query and Z_q is the normalization factor for q.

Scoring

In order to produce a score for a document given a query, SSI adopts a modification of the classical dot product used in the classical VSM. Specifically, the scoring function writes as:

$$s^{SSI}(Q, D) = q^T (U^T U + I) d = \tilde{q}^T \tilde{d} + q^T d. \tag{6.2}$$

SSI combines two scores obtained in different representation spaces: the first one is the dot product on the latent space and the second one is the dot product in the original space. This way the model learns the tradeoff between using low dimensional space and a classical term-based score.

Learning

The parameter matrix U is learned by employing a margin ranking-loss which has already been used in several *learning-to-rank* scenarios (Collobert et al., 2011):

$$L_{\mathcal{D}}^{SSI}(U) = \sum_{l=1}^{L} [1 - s^{SSI}(Q_l, D_l) + s^{SSI}(Q_l, D_c)]_+ \tag{6.3}$$

where D_c is a non-relevant document for this query and $[y]_+ = max(0, y)$ and 1 is called margin. This loss encourages the model to keep the scores of relevant documents greater than the scores of non-relevant ones at least by 1. The loss is minimized through stochastic gradient descent (SGD). Iteratively, one picks a random triplet (q_l, d_l, d_c) and update the parameters U by taking a gradient step for that triplet. In order to gather more insights on how the model behaves, we write

the derivatives with respect to each of the hidden embeddings appearing in the current update. Denote $\tilde{x}_q, \tilde{x}_d$ and $\tilde{x}_c$ the embedding of a concept appearing in the query, relevant document and non-relevant document respectively. The negative gradients for these parameters are:

$$-\frac{\partial L_{\mathcal{D}}^{SSI}}{\partial \tilde{x}_q} \approx \tilde{d}_l - \tilde{d}_c, \quad -\frac{\partial L_{\mathcal{D}}^{SSI}}{\partial \tilde{x}_d} \approx \tilde{q}_l, \quad -\frac{\partial L_{\mathcal{D}}^{SSI}}{\partial \tilde{x}_c} \approx -\tilde{q}_l,$$

where the approximation sign means up to a normalization constant, i.e. the gradients should be multiplied respectively by $Z_{q_l}^{-1}$, $Z_{d_l}^{-1}$ and Z_c^{-1}. By analyzing the gradient update step, we recognize the familiar form of Rocchio query updates (Rocchio, 1971). Each query word is moved towards the direction of relevant documents and far from non-relevant ones. As a by-product, the updated query representation will point in that direction. We will see that the updates of our model can be seen as a refinement of these updates, where the contribution of the relevant and non-relevant documents is weighted by its similarity to the query.

6.3.2 Quantum Entropy Minimization

In order to learn the latent embeddings, we stem from the computational framework proposed by the recent Quantum Language Modeling (QLM) approach for IR (Sordoni et al., 2013). This formal retrieval framework embeds concepts into rank-one projectors. Documents and queries are embedded into a special matrix called *density matrix*, a well-known mathematical object in physics. The authors show that this representation extends classical unigram language models and can be used to capture richer information than single terms from text excerpts. Given a query, documents are scored using a generalization of classical relative entropy to matrix domains called *quantum relative entropy*. Our contribution here is to show how it is possible to leverage the proposed representation and scoring function in order to learn semantic representations for each concept. From now on, we will call our model Quantum Entropy Minimization (QEM).

Representation

Stemming from the original QLM approach, we embed each concept in the vocabulary with a rank-one projector $\widetilde{\Pi}_\kappa$. Rank-one projectors are projection ma-

trices onto one-dimensional subspaces. They are parameterized as outer products of unit-norm vectors, i.e. they have only K free parameters, $\widetilde{\Pi}_\kappa = \tilde{x}_\kappa \tilde{x}_\kappa^T$, $\|\tilde{x}_\kappa\|_2 = 1$. Hence, we can still consider our latent embeddings as columns vectors of a parameter matrix $U \in \mathbb{R}^{K \times N}$, without entering matrix domains. Also, our embeddings are normalized and lie on the unit sphere.

Documents and queries are associated to a density matrix, which can be understood as a convex combination of concepts projectors. From a linear algebra perspective, a density matrix W is symmetric, positive-semidefinite and of unitary trace, $W \in \mathcal{S}_+^K = \{W : W \in \mathbb{R}^{K \times K}, W = W^T, W \succeq 0, \operatorname{tr} W = 1\}$. In QLM, the density matrix for a query (or a document) is obtained by maximizing the following convex log-likelihood form:

$$\mathcal{L}_Q(W) = \sum_{\kappa \in Q} \log \operatorname{tr} W \Pi_\kappa, \tag{6.4}$$

where the sum is over the number of concepts appearing in the query. The maximization should be restricted to the feasible set $\mathcal{S}_+^K$, i.e. the solution should be a proper density matrix. The expression $\operatorname{tr} W \Pi_\kappa$ can be considered as a similarity between the query and the concept representations. This maximization is difficult and has to be approximated by iterative methods (Sordoni et al., 2013).

In order to have a smooth analytic solution of Eq. 5, we choose to approximate the objective by a linear Taylor's expansion of $\log x$ around $x = 1$, $\log x \approx x - 1$. Hence, the linear Taylor approximation $\mathcal{L}_Q^I(W)$ of $\mathcal{L}_Q(W)$ writes as:

$$\mathcal{L}_Q^I(W) = \sum_{\kappa \in Q} \operatorname{tr} W \widetilde{\Pi}_\kappa \tag{6.5}$$

up to a constant shift. In order to see what is the effect of this approximation, note that $0 \leq \operatorname{tr} W \widetilde{\Pi}_\kappa \leq 1$. The linear approximation cuts-off the infinity of the log function around zero. Hence, the approximation is very accurate when the density matrix is "around" $\widetilde{\Pi}$, but badly underestimates the loss when $\operatorname{tr} W \widetilde{\Pi}_\kappa$ is small. As a result, the approximate objective could "forget" to represent some concepts in the documents, i.e. the objective could be high even if $\operatorname{tr} W \widetilde{\Pi}_\kappa$ is very low for some κ. Coming up with more accurate approximations is certainly an interesting way to improve the model. Nevertheless, we found that this linear approximation works well in practice.

The maximization of Eq. 6.5 is performed by enforcing the unit-trace constraint $\operatorname{tr} W = 1$ through a Langrangian multiplier λ. We have:

$$\mathcal{L}_Q^I(W) = \sum_{\kappa \in Q} \operatorname{tr} W \widetilde{\Pi}_\kappa - \lambda \left(\operatorname{tr} W - 1 \right) \tag{6.6}$$

We compute the gradient with respect to W and we set it to zero obtaining $\lambda W = \sum_{\kappa \in Q} \widetilde{\Pi}_\kappa$. By taking the trace on both sides and exploiting the fact that for unit rank projectors $\operatorname{tr} \widetilde{\Pi}_\kappa = 1$, we find that the multiplier $\lambda = N_Q$, the number of concepts in the query. Therefore, the latent representation $\widetilde{W}_Q$ for the query Q can be written as:

$$\widetilde{W}_Q = {N_Q}^{-1} \sum_{\kappa \in Q} \widetilde{\Pi}_\kappa = {N_Q}^{-1} \sum_{\kappa \in Q} \tilde{x}_\kappa \tilde{x}_\kappa^T, \tag{6.7}$$

As the combination of symmetric positive-definite matrices is still positive-definite - see for example (Nielsen and Chuang, 2010) - the solution above is a valid maximizer of $\mathcal{L}_Q^I(W)$, i.e. $\widetilde{W}_Q$ lies in the feasible set $\mathcal{S}^K$.

Considering the solution presented in Eq. 6.7, we see that our model represents documents and queries as mixtures of rank-one projectors. Contrary to existing embeddings models such as SSI, documents and queries lie in a larger space than the concepts themselves. This is intuitively appealing for it seems reductive to consider them as carrying the same information as single concepts. In our model, this idea is embodied by the notion of *rank*: concepts from the vocabulary are embedded in rank-one matrices; as documents and queries are mixtures of rank-one matrices, they can have higher rank and tend to degenerate to rank-one matrices if and only if the projectors for their component terms get closer to each other, i.e. they all encode the same semantic information.

Scoring

Given a document density matrix W_D and a query density matrix W_Q, both estimated through Eq. 6.7, QLM defines the retrieval score for a document with respect to a query with a generalization of the classical relative entropy called quantum relative entropy:

$$s(Q, D) = \operatorname{tr} W_Q \log W_D, \tag{6.8}$$

where log denotes the matrix logarithm, i.e. the classical logarithm applied to the matrix eigenvalues. In order to formulate a differentiable form of the scoring function, we expand the matrix logarithm in Eq. 6.8 by its Taylor's series around I_K, the identity matrix in $\mathbb{R}^{K\times K}$. This is a common choice for matrix logarithm (Nielsen and Chuang, 2010). Truncating to the linear expansion term we obtain:

$$\log W \approx \log^{\mathrm{I}} W = W - I_K. \tag{6.9}$$

Hence, the first-order approximation of the matrix logarithm is just the matrix itself, up to a constant shift. By substituting the expression above in our scoring function we obtain our linear approximation:

$$s^{QEM}(Q, D) = \operatorname{tr} W_Q(W_D - I_K) \overset{rank}{=} \operatorname{tr} W_Q W_D, \tag{6.10}$$

where the rank equivalence is obtained by noting that the constant shift does not depend on a particular document thus cannot influence the relative rank of two documents with respect to a given query. This scoring function is the generalization of dot product for symmetric matrices. However, in the case of density matrices, $s^{QEM}(Q, D)$ is bounded and ranges in $[0, 1]$ (Nielsen and Chuang, 2010).

Learning

Similarly to SSI, we adopt margin-ranking loss in order to train our model. In our case however, instead of fixing the margin to 1, we consider it as an hyperparameter:

$$L_{\mathcal{D}}^{QEM}(U) = \sum_{l=1}^{L} [m - s^{QEM}(Q_l, D_l) + s^{QEM}(Q_l, D_c)]_+. \tag{6.11}$$

As our scoring function is bounded from above exactly by 1, parameterizing the margin is necessary. If the margin was fixed to 1, the model would always suffer a loss. We also choose to minimize our objective function by SGD. By exploiting the analytic approximate solution for the density matrices in Eq. 6.7, we can rewrite

our scoring function as:

$$
\begin{aligned}
s^{QEM}(Q, D) &= Z \sum_{\kappa \in Q} \sum_{\eta \in D} \operatorname{tr} \tilde{x}_\kappa \tilde{x}_\kappa^T \tilde{x}_\eta \tilde{x}_\eta^T \quad \text{(Linearity of trace)} \\
&= Z \sum_{\kappa \in Q} \sum_{\eta \in D} \operatorname{tr} \tilde{x}_\kappa^T \tilde{x}_\eta \tilde{x}_\eta^T \tilde{x}_\kappa \quad \text{(Circular Property)} \\
&= Z \sum_{\kappa \in Q} \sum_{\eta \in D} (\tilde{x}_\kappa^T \tilde{x}_\eta)^2,
\end{aligned}
$$

where the first inequality is given by the linearity of the trace, the second one by the circular property of the trace and $Z = N_Q^{-1} N_D^{-1}$. Working out the gradients is straightforward. Denote $\tilde{x}_q, \tilde{x}_d$ and $\tilde{x}_c$ the embedding of a concept appearing in the query, in the relevant document and in the non relevant document respectively. Our updates are:

$$
-\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_q} \approx \tilde{x}_q^T (\widetilde{W}_{D_l} - \widetilde{W}_{D_c}), \; -\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_d} \approx \tilde{x}_d^T \widetilde{W}_Q, \; -\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_c} \approx -\tilde{x}_c^T \widetilde{W}_Q,
$$

where the approximation sign means up to a normalization constant, i.e. the gradients should be multiplied respectively by $2N_Q^{-1}$, $2N_{D_l}^{-1}$ and $2N_{D_c}^{-1}$. The updates look very similar to the SSI updates except for a dot product, which appears in the update. In order to gain more insight on what's happening, let's develop the update for $\tilde{x}_q$ by substituting the density matrices with their explicit form in Eq. 6.7:

$$
-\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_q} \approx N_{D_l}^{-1} \sum_{\kappa \in D_l} (\tilde{x}_\kappa^T \tilde{x}_q) \tilde{x}_\kappa - N_{D_c}^{-1} \sum_{\eta \in D_c} (\tilde{x}_\eta^T \tilde{x}_q) \tilde{x}_\eta. \tag{6.12}
$$

Differently from SSI, the update direction for a query concept is not a static linear combination of relevant and non-relevant document embeddings: our model does not require $\tilde{x}_q$ to be near each of the concepts of the relevant document $\tilde{x}_\kappa$ and far away each of the concepts of the non-relevant document $\tilde{x}_\eta$. Instead, $\tilde{x}_q$ is moved towards the region of its nearest document concepts $\tilde{x}_\kappa$ and farther away from its nearest non-relevant document concepts $\tilde{x}_\eta$. Similarly to a translation model, this has the effect of *selecting* which document concepts the query concept should be aligned to: in general the selection will be driven by co-occurrence patterns. Interestingly, we also obtain a refinement of the Rocchio expansion method. The update direction for query expansion is obtained by weighting relevant and non-relevant documents by their similarity to the query: we require the query to be

Anchor Log	# Anchors	# κ	# Uni	# Big
WIKI	13,570,292	442,738	167,615	275,123

Table 6.1 – Number of anchors, concepts, unigram and bigram concepts in the anchor log used in the experiments.

near to the most similar relevant documents and far away from the most similar non-relevant documents, which is intuitive and can help to filter out noise in the relevance labels.

6.4 Experimental study

6.4.1 Experimental setup

All our experiments were conducted using the open source Indri search engine. As query expansion with external resources have shown to be effective for difficult web queries, we test the effectiveness of our approach on the ClueWeb09B collection, a noisy web collection containing 50,220,423 documents. We choose to use the three set of topics of the TREC Web Track from 2010 to 2012 (topics 51-200). In addition to MAP, precision at top-ranks is an important feature for query expansion models. Hence, we also report NDCG@10 and the recent ERR@10, which correlates better with click metrics than other editorial metrics (Chapelle et al., 2009). The statistical significance of differences in the performance of tested methods is determined using a randomization test (Smucker et al., 2007) evaluated at $\alpha < 0.05$.

Baselines

We first propose to compare all our baselines to a standard language modelling (LM) approach for IR, which does not exploit query expansion techniques. In order to provide a strong baseline performing traditional query expansion, we compare our model with the successful concept translation model (CTM), which allows to find translations from/to terms or longer phrases (Gao and Nie, 2012). We also propose to compare our model to SSI as it shares the same learning rationale and was conceived for similar datasets.

Model	$p(\kappa\|\theta_E)$
CTM	$\sum_{\eta\in Q} p(\kappa\|\eta)p(\eta\|\theta_Q) \approx \sum_{\eta\in Q} p(\kappa\|\eta)$
SSI	$\exp\ \tilde{x}_\kappa^T \tilde{q} \approx \exp \sum_{\eta\in Q} \tilde{x}_\kappa^T \tilde{x}_\eta$
QEM	$\exp \operatorname{tr} \widetilde{W}_Q \widetilde{\Pi}_\kappa \approx \exp \sum_{\eta\in Q} (\tilde{x}_\kappa^T \tilde{x}_\eta)^2$

Table 6.2 – Explicit parameterizations of the probability of an expansion concept given the query for each of the models.

Anchor log

The studies asserting the efficiency of clickthrough data for QE nearly all make use of proprietary query logs (Gao and Nie, 2012; Gao et al., 2010). In Dang and Croft (2010), the authors show that an anchor log made of anchor text / title pairs can bring similar performance to a real query log for query reformulation purposes. For this paper, we built the anchor log from the high-quality Wikipedia collection [1]. Anchor texts on Wikipedia have already been successfully used for expansion purposes in Arguello et al. (2008) for blog recommendation task. In order to embed both terms and compound concepts, we included all terms and bigrams occurring more than 6 times in the corpus. Table 6.1 reports some statistics about our paired corpus.

Query expansion

In order to evaluate the effectiveness of the proposed approach and the baselines, we perform QE using the powerful KL-divergence framework (Zhai, 2008). KL has been used in numerous QE studies as a way of integrating expansion terms mined from a variety of external resources (Kotov and Zhai, 2012). Given a query language model θ_Q and a document model θ_D, the documents in the collection are scored according to the relative entropy:

$$s^{KL}(Q, D) = \sum_{\kappa\in\mathcal{V}} p(\kappa|\theta_Q) \log p(\kappa|\theta_D) \tag{6.13}$$

where κ is an entry of the vocabulary. The process of QE is obtained by smoothing the query language model with a concept model θ_E obtained by external resources:

$$p(\kappa|\tilde{\theta}_Q) = \lambda\, p(\kappa|\theta_Q) + (1-\lambda)\, p(\kappa|\theta_E), \tag{6.14}$$

1. http://www.wikipedia.org

which has the effect of assigning non-zero probability of an expansion concept. The training of λ is discussed in more details in the next section. In order to test the quality of the mined expansion terms, it is necessary to parameterize the probability $p(\kappa|\theta_E)$ for each of the tested models. These are reported in Table 6.2. In CTM, the model θ_E is considered as a mixture of translation probabilities corresponding to query concepts where the translation probabilities $p(\kappa|\eta)$ are estimated on the anchor log and $p(\eta|\theta_Q) = N_Q^{-1}$ is the uniform query distribution. For all the latent models, we parameterize the probability of a term given a query by employing a softmax formulation, i.e. (Mikolov et al., 2013). The energy is the similarity between a concept and a query which conjugates differently in the different models. In SSI, this similarity is the inner product between the query and the concept latent representations, i.e. $\tilde{x}_\kappa^T \tilde{q}$. In QEM, we follow the formulation in Eq. 6 and naturally consider the similarity of a concept given a query as $\operatorname{tr} \widetilde{W}_Q \widetilde{\Pi}_\kappa$. Differently from SSI and similarly to CTM, in our approach the contributions of query terms are always positive, which reminds the basic rationale of successful approaches such as NMF or LDA.

Hyperparameter Selection

A novelty of this work is that we choose to train all the hyperparameters of the models in order to optimize expansion performance measured with MAP. In this paper, we use a random search recently proposed in Bergstra and Bengio (2012). Our procedure is depicted in Fig. 6.1. Given our anchor log $\mathcal{D}$, we sample hyper parameters Φ from a uniform distribution over a fine-grained set of possible values Ω_Φ. Clamping Φ, we train the model parameters (embeddings or translation probabilities) on the anchor log. We expand the original queries by selecting the top-10 concepts according to the parameterization discussed previously. Finally, we tune by grid-search the smoothing parameter λ. We repeat the process $n = 50$ times in order to have good chances to find minima of the hyperparameter space. We report the results obtained by performing 5-fold cross-validation. For all the models we cross-validate λ. For all the embeddings model, we fix the number of latent dimensions to $K = 100$, the number of epochs to 3. For SSI, we cross-validate the gradient step, while for QEM we include also the margin m.

(a) Training Phase

$\mathcal{Q} \leftarrow$ Train queries

For $t = 1 \ldots n$

1. $\Phi^t \sim \text{Random}(\Omega_\Phi)$
2. $\mathcal{M}^t \leftarrow \text{Train}(\mathcal{D}, \Phi^t)$
3. $\mathcal{Q}_E \leftarrow \text{Expand}(\mathcal{Q}, \mathcal{M}^t)$
4. $\lambda^t \leftarrow \text{Grid}(\mathcal{Q}_E, \lambda)$
5. $\text{MAP}_{\Phi^t} \leftarrow \text{Search}(\mathcal{Q}_E, \lambda^t)$
6. If $\text{MAP}_{\Phi^t} \geq \text{MAP}_{\Phi^*}$

   5.1 $\Phi^* = \Phi^t, \lambda^* = \lambda^t$

Return Φ^*, λ^*

(a) Testing Phase

$\mathcal{Q} \leftarrow$ Test queries

1. $\mathcal{M}^* \leftarrow \text{Train}(\mathcal{D}, \Phi^*)$
2. $\mathcal{Q}_E \leftarrow \text{Expand}(\mathcal{Q}, \mathcal{M}^*)$
3. $\text{MAP}_{\Phi^*} \leftarrow \text{Search}(\mathcal{Q}_E, \lambda^*)$
4. Return MAP_{Φ^*}

Figure 6.1 – Algorithms for training (a) and testing (b) the hyper parameters Φ of the expansion models directly on MAP.

Queries	Model	nDCG@10	ERR@10	MAP
WT10	LM	.0850	.0443	.1069
	CTM	.0954	.0494	.1128
	SSI	.0877	.0437	.1123
	QEM	**.1091**$_s$	**.0583**$_{cs}$	**.1137**
WT11	LM	.1341	.0613	.0894
	CTM	.1278	.0611	.0936
	SSI	.1331	.0624	.0882
	QEM	**.1514**$_{cs}$	**.0727**$_{cs}$	**.1002**$_s$
WT12	LM	.0738	.1087	.1047
	CTM	.0837	.1144	.1095
	SSI	**.1063**	.1475	.1200
	QEM	.1040$_c$	**.1488**$_c$	**.1210**$_c$

Table 6.3 – Evaluation of the performance for the four methods tested. Best results are highlighted in boldface. Numbers in parentheses indicate relative improvement (%) over SSI and CTM. s, c means statistical significance over SSI and CTM.

6.4.2 Results

Table 6.3 resumes all our experimental results. First of all, we note that all the expansion methods increase significantly on the term-matching retrieval baseline LM. Our implementation of CTM trained on the high-quality Wikipedia anchor logs has overall positive effects on the three reported measures and on the three collections of topics tested. CTM increases considerably the precision at top-ranks, achieving relative improvements up to 13.4% on nDCG@10 and 11.51% on ERR@10 for WT-10 and WT-12. For WT-11, CTM suffers non-significant losses with respect to LM on precision-oriented measures while still achieving 4.69% relative improvement on MAP. Analyzing the average query length on three collections of topics tested, we found for WT-10, WT-11 and WT-12 respectively 1.979, 3.396 and 2.122. WT-11 queries are thus longer on average and reflect long-tail queries which are particularly difficult to expand because of the complex syntactic relationships between terms in the query formulation. We then compared latent semantic models with CTM. Experimental results confirm that learned semantic spaces can be useful in encoding useful relationships for query expansion. Even when fixing a relatively low latent dimensionality, i.e. K = 100, SSI performs as well as CTM on WT-10 while outperforming the latter on WT-12 on all measures. QEM outperforms both SSI and CTM yielding consistent improvements for all the topics tested. It is interesting to note that SSI is not effective on WT11 and actually degrades performance with respect to the baseline LM nearly for all the measures reported. By representing queries as linear combination of concepts embeddings, SSI seems to fail in capturing semantic content of relatively long queries such as those found in WT-11. The fact that QEM increases significantly all measures on those difficult topics brings evidence towards the usefulness of the enriched query representation space, capable of adequate modelling of longer text sequences. It is also striking how QEM can bring relative improvements both on SSI and CTM for precision at top-ranks by at least 14% in WT-10 and 13% for difficult WT-11 topics. This is especially important in web search where top-ranks are most valuable for users. It seems that QEM can select compact and focused expansion concepts in order to increase the quality of top-ranked documents. On WT-12, the situation is more mitigated but still QEM can bring improvements over CTM and SSI. Even if not reported here, we conducted preliminary experiments by varying the number of dimensions and by choosing a more appropriate ranking loss such as

proposed in Weston et al. (2011) and found that the performance of QEM can be further increased by a significant amount with respect to classical CTM and SSI. Therefore, the automatic setting of appropriate dimensions will be an interesting research in the future.

6.5 Conclusion

Overall, we believe that the potential of latent semantic model for encoding useful semantic relationship is real and should be fostered by enriching query and document representations. To this end, we proposed a new method called Quantum Entropy Minimization (QEM) which is, to our knowledge, the first model to allocate text sequences in a larger space than their component terms. This is automatically encoded in the notion of rank. Higher-rank objects encode broader semantic information while unit-rank objects bring only localized semantic content. Experimental results show that our model is useful in order to boost precision at top-ranks with respect to a state-of-the-art expansion model and a recently proposed semantic model. Particularly striking was the ability of our model to find useful expansion terms for longer queries: we believe this is a direct consequence of the higher semantic resolution allocated by our model. There are many interesting directions for future research. One could find more reasonable approximations both to the scoring function and the representation capable of bringing further improvements. Finally, we argue that incorporating existing advanced gradient descent procedures, refined loss functions can certainly further increase the retrieval performance, well beyond traditional query expansion methods.

Related Articles

Although not explicitly discussed in this thesis, we published two related papers on supervised feature learning to encode semantic relationships and overcome exact matching problems:

Modelling Latent Topic Interactions using Quantum Interference for IR. Alessandro Sordoni, Jing He, Jian-Yun Nie, *Proceedings of CIKM (CIKM '13)*.

Summary. In this article, we use an unsupervised feature learning technique called Latent Dirichlet Allocation (LDA) to estimate a latent topic space (Blei et al., 2003) and we apply the mathematical framework of quantum interference to model the interactions between topics. Instead of performing query expansion, we evaluate our model in a document expansion setting.

Compact Aspect Embedding for Diversified Query Expansions. Xiaohua Liu, Arbi Bouchoucha, Alessandro Sordoni, Jian-Yun Nie, *Proceedings of the 28th AAAI conference (AAAI '14)*.

Summary. In this paper, we propose a novel method for query expansion, called compact aspect embedding, which exploits trace norm regularization to learn a low rank embedding space for each query, with each eigenvector of the learnt vector space representing an aspect of the query, and the absolute value of its corresponding eigenvalue representing the association strength of that aspect to the query. Meanwhile, each expansion term is mapped into the vector space as well.

7 A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion

Prologue

Article Details

A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob G. Simonsen, Jian-Yun Nie. *Proceedings of CIKM (CIKM '15)*, pp. 553–562.

Personal Contribution The idea for the new hierarchical architecture is my own. I wrote the code for the model with initial support from Caglar Gulcehre and ran all the experiments. Hossein Vahabi provided guidance on setting up the experimental framework to benchmark the effectiveness of the proposed method. The idea of performing a robust prediction task was my own. Christina Lioma and Jakob G. Simonsen participated to the initial brainstorming while I was interning in Copenhagen. All the authors contributed to the correction of the article.

Context

In order to support users during their search tasks, current search engines provide query suggestions, i.e. alternative textual formulations of the user information need which are likely to lead to more relevant results. In this article, we investigate the use of recurrent neural networks (RNN) for the problem of contextual query suggestion: given a sequence of past queries issued by the user, our objective is to predict the query the user will issue next. RNNs are notorious deep learning architectures that have demonstrated to be extremely effective in a variety of NLP tasks such as Language Modeling (LM) (Mikolov et al., 2010; Pascanu et al., 2013) and Machine Translation (MT) (Cho et al., 2014; Sutskever et al., 2014). The overwhelming success of RNNs stems from their ability to model long-term de-

pendencies appearing in the input sequences. In our setting, capturing long-term dependencies is important as the user may have issued a large number of previous queries. Moreover, RNNs come with other desirable properties such as robustness to long-tail queries and query generation capabilities. Until today, the application of RNNs techniques to IR remains unexplored.

Contributions

The core contribution of the article is to present a first application of RNNs to query suggestion. Our novel hierarchical recurrent encoder-decoder (HRED) architecture makes possible to condition the suggestion on a – theoretically unlimited – number of previously submitted queries. Additionally, our model can suggest for rare, or long-tail, queries. The produced suggestions are synthetic and are sampled one word at a time, using computationally cheap decoding techniques. This is in contrast to current synthetic suggestion models relying upon machine learning pipelines and hand-engineered feature sets. We believe that, in addition to query suggestion, our architecture is general enough to be used in a variety of other applications.

Recent Developments

Although the article has been published very recently, some works expanded upon its core ideas and architecture. Serban et al. (2016) extend the hierarchical architecture to deal with a response generation in conversational tasks. Similarly, Yao et al. (2015) extend the architecture by introducing a neural attention component.

7.1 Introduction

Modern search engines heavily rely on query suggestions to support users during their search task. Query suggestions can be in the form of auto-completions or query reformulations. Auto-completion suggestions help users to complete their queries while they are typing in the search box. In this paper, we focus on query

reformulation suggestions, that are produced after one or more queries have already been submitted to the search engine.

Search query logs are an important resource to mine user reformulation behaviour. The query log is partitioned into query sessions, i.e. sequences of queries issued by a unique user and submitted within a short time interval. A query session contains the sequence of query reformulations issued by the user while attempting to complete the search mission. Therefore, query co-occurrence in the same session is a strong signal of query relatedness and can be straightforwardly used to produce suggestions.

Methods solely relying on query co-occurrence are prone to data sparsity and lack coverage for rare and *long-tail* queries, i.e. unseen in the training data. A suggestion system should be able to translate infrequent queries to more common and effective formulations based on similar queries that have been seen in the training data. Amongst the interesting models that have been proposed, some capture higher order collocations (Boldi et al., 2008), consider additional resources (Jain et al., 2011; Vahabi et al., 2013), move towards a word-level representation (Bonchi et al., 2012; Broccolo et al., 2012) or describe queries using a rich feature space and apply learning to rank techniques to select meaningful candidates (Ozertem et al., 2012; Santos et al., 2013).

An additional desirable property of a suggestion system is *context-awareness*. Pairwise suggestion systems operate by considering only the most recent query. However, previous submitted queries provide useful context to narrow down ambiguity in the current query and to produce more focused suggestions (Jiang et al., 2014). Equally important is the order in which past queries are submitted, as it denotes generalization or specification reformulation patterns (Huang and Efthimiadis, 2009). A major hurdle for current context-aware models is dealing with the dramatic growth of diverse contexts, since it induces sparsity, and classical count-based models become unreliable (Cao et al., 2008; He et al., 2009).

Finally, relatively unexplored for suggestion systems is the ability to produce *synthetic* suggestions. Typically, we assume that useful suggestions are already present in the training data. The assumption weakens for rare queries or complex information needs, for which it is possible that the best suggestion has not been previously seen (Jain et al., 2011; Szpektor et al., 2011). In these cases, synthetic suggestions can be leveraged to increase coverage and can be used as candidates in

complex learning to rank models (Ozertem et al., 2012).

We present a generative probabilistic model capable of producing synthetic, context-aware suggestions not only for popular queries, but also for long tail queries. Given a sequence of queries as prefix, it predicts the most likely sequence of words that follow the prefix. Variable context lengths can be accounted for without strict built-in limits. Query suggestions can be mined by sampling likely continuations given one or more queries as context. Prediction is efficient and can be performed using standard natural language processing word-level decoding techniques (Koehn, 2009). The model is robust to long-tail effects as the prefix is considered as a sequence of words that share statistical weight and not as a sequence of atomic queries.

As an example, given a user query session composed of two queries *cleveland gallery* $\rightarrow$ *lake erie art* issued sequentially, our model predicts sequentially the words *cleveland*, *indian*, *art* and $\circ$, where $\circ$ is a special end-of-query symbol that we artificially add to our vocabulary. As the end-of-query token has been reached, the suggestion given by our model is *cleveland indian art*. The suggestion is contextual as the concept of *cleveland* is justified by the first query thus the model does not merely rely on the most recent query only. Additionally, the produced suggestion is synthetic as it does not need to exist in the training set.

To endow our model with such capabilities, we rely on recent advances in generative natural language applications with neural networks (Bengio, 2013; Cho et al., 2014; Mitra, 2015). We contribute with a new hierarchical neural network architecture, called hierarchical recurrent encoder-decoder (HRED), that allows to embed a complex distribution over sequences of sentences within a compact parameter space. Differently from count-based models, we avoid data sparsity by assigning single words, queries and sequences of queries to embeddings, i.e. dense vectors bearing syntactic and semantic characteristics (Figure 7.1) (Bengio et al., 2003). Our model is compact in memory and can be trained end-to-end on query sessions. We envision future applications to various tasks, such as query auto-completion, query next-word prediction and general language modeling.

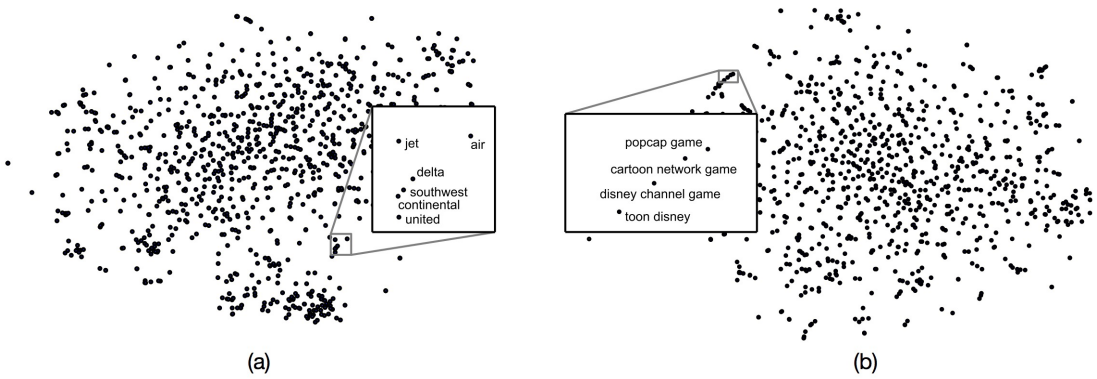


Figure 7.1 – Projection of the (a) word and (b) query embeddings learnt by our neural network architecture. Topically similar terms and queries are close in the embedding space.

7.2 Key Idea

Suggestion models need to capture the underlying similarities between queries. Vector representations of words and phrases, also known as *embeddings*, have been successfully used to encode syntactic or semantic characteristics thereof (Bengio, 2013; Bengio et al., 2003; Mikolov et al., 2013; Shen et al., 2014). We focus on how to capture query similarity and query term similarity by means of such embeddings. In Figure 7.1 (a) and (b), we plot a two-dimensional projection of the word and query embeddings learnt by our model. The vectors of topically similar terms or queries are close to each other in the vector space.

Vector representations for phrases can be obtained by averaging word vectors (Mikolov et al., 2013). However, the order of terms in queries is usually important (Sordoni et al., 2013). To obtain an order-sensitive representation of a query, we use a particular neural network architecture called Recurrent Neural Network (RNN) (Bengio, 2013; Mikolov et al., 2010). For each word in the query, the RNN takes as input its embedding and updates an internal vector, called *recurrent* state, that can be viewed as an order-sensitive summary of all the information seen up to that word. The first recurrent state is usually set to the zero vector. After the last word has been processed, the recurrent state can be considered as a compact order-sensitive encoding of the query (Figure 7.2 (a)).

A RNN can also be trained to decode a sentence out of a given query encoding. Precisely, it parameterizes a conditional probability distribution on the space of possible queries given the input encoding. The process is illustrated in Fig-

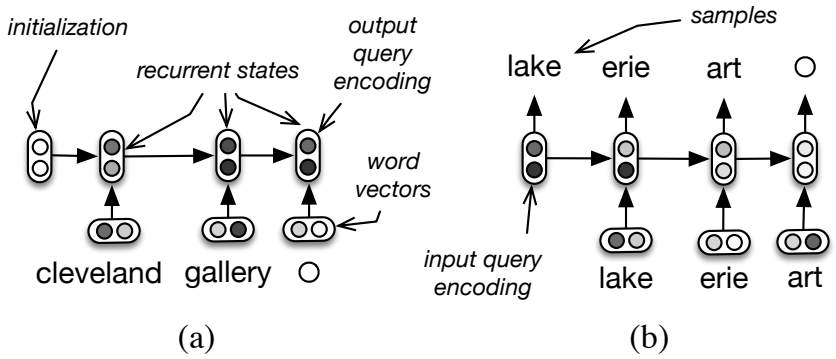


Figure 7.2 – (a) An encoder RNN processing the query *cleveland gallery* followed by a special end-of-query symbol ∘. Each solid arrow represents a non-linear transformation. (b) A decoder RNN generating the next query in the session, *lake erie art*, from a query encoding as input.

ure 7.2 (b). The input encoding may be used as initialization of the recurrence. Then, each of the recurrent states is used to estimate the probability of the next word in the sequence. When a word is sampled, the recurrent state is updated to take into account the generated word. The process continues until the end-of-query symbol ∘ is produced.

The previous two use cases of RNNs can be pipelined into a single recurrent encoder-decoder, as proposed in (Cho et al., 2014; Sutskever et al., 2014) for Machine Translation purposes. The architecture can be used to parameterize a mapping between sequences of words. This idea can be promptly casted in our framework by predicting the next query in a session given the previous one. With respect to our example, the query encoding estimated by the RNN in Figure 7.2 (a) can be used as input to the RNN in Figure 7.2 (b): the model learns a mapping between the consecutive queries *cleveland gallery* and *lake erie art*. At test time, the user query is encoded and then decoded into likely continuations that may be used as suggestions.

Although powerful, such mapping is pairwise, and as a result, most of the query context is lost. To condition the prediction of the next query on the previous queries in the session, we deploy an additional, session-level RNN on top of the query-level RNN encoder, thus forming a *hierarchy* of RNNs (Figure 7.3). The query-level RNN is responsible to encode a query. The session-level RNN takes as input the query encoding and updates its own recurrent state. At a given position in the session, the session-level recurrent state is a learnt summary of the past queries,

keeping the information that is relevant to predict the next one. At this point, the decoder RNN takes as input the session-level recurrent state, thus making the next query prediction contextual.

The contribution of our hierarchical recurrent encoder-decoder is two-fold. The query-level encoder RNN maps similar queries to vectors close in the embedding space (Figure 7.1 (b)). The mapping generalizes to queries that have not been seen during training, as long as their words appear in the model vocabulary. This allows the model to map rare queries to more useful and general formulations, well beyond past co-occurred queries. The session-level RNN models the sequence of the previous queries, contextualizing the prediction of the next query. Similar contexts are mapped close to each other in the vector space. This property allows to avoid sparsity, and differently from count-based models (Cao et al., 2008; He et al., 2009), to account for contexts of arbitrary length.

7.3 Mathematical Framework

We start by presenting the technical details of the RNN architecture, which our model extends. We consider a query session as a sequence of M queries $S = \{Q_1, \ldots, Q_M\}$ submitted by a user in chronological order, i.e. $Q_m <_t Q_{m+1}$ where $<_t$ is the total order generated by the submission time, and within a time frame, usually 30 minutes. A query Q_m is a sequence of words $Q_m = \{w_{m,1}, \ldots, w_{m,N_m}\}$, where N_m is the length of query m. V is the size of the vocabulary.

7.3.1 Recurrent Neural Network

For each query word w_n, a RNN computes a dense vector called *recurrent* state, denoted h_n, that combines w_n with the information that has already been processed, i.e. the recurrent state h_{n-1}. Formally:

$$h_n = f(h_{n-1}, w_n),\ h_0 = 0 \tag{7.1}$$

where $h_n \in \mathbb{R}^{d_h}$, d_h is the number of dimensions of the recurrent state, f is non-linear transformation and the recurrence is seeded with the 0 vector. The recurrent

state h_n acts as a compact *summary* of the words seen up to position n.

Usually, f consists of a non-linear function, i.e. the logistic sigmoid or hyperbolic tangent, applied element-wise to a time-independent affine transformation (Mikolov et al., 2010). The complexity of the function f has an impact on how accurately the RNN can represent sentence information for the task at hand. To reduce the fundamental difficulty in learning long-term dependencies (Bengio et al., 1994), i.e. to store information for longer sequences, more complex functions have been proposed such as the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and the Gated Recurrent Unit (GRU) (Cho et al., 2014).

Once Eq. 7.1 has been run through the entire query, the recurrent states $h_1, \ldots, h_N$ can be used in various ways. In an encoder RNN, the last state h_N may be viewed as an order-sensitive compact summary of the input query. In a decoder RNN, the recurrent states are used to predict the next word in a sequence (Cho et al., 2014; Mikolov et al., 2010). Specifically, the word at position n is predicted using h_{n-1}. The probability of seeing word v at position n is:

$$P(w_n = v|w_{1:n-1}) = \frac{\exp o_v^\top h_{n-1}}{\sum_k \exp o_k^\top h_{n-1}}, \tag{7.2}$$

where $o_i \in \mathbb{R}^{d_e}$ is a real-valued vector of dimensions d_e associated to word i, i.e. a word embedding, and the denominator is a normalization factor. A representation of the embeddings learnt by our model is given in Figure 7.1 (a). The semantics of Eq. 7.2 dictates that the probability of seeing word v at position n increases if its corresponding embedding vector o_v is "near" the context encoded in the vector h_{n-1}. The parameters of the RNN are learned by maximizing the likelihood of the sequence, computed using Eq. 2.

Gated Recurrent Unit

We choose to use the Gated Recurrent Unit (GRU) as our non-linear transformation f. GRUs have demonstrated to achieve better performance than simpler parameterizations at an affordable computational cost (Cho et al., 2014). This function reduces the difficulties in learning our model by easing the propagation of the gradients. We let w_n denote the one-hot representation of $w_n = v$, i.e. a vector of the size of the vocabulary with a 1 corresponding to the index of the query word

v. The specific parameterization of f is given by:

$$\begin{aligned} r_n &= \sigma(I_r w_n + H_r h_{n-1}), && \textit{(reset gate)} \\ u_n &= \sigma(I_u w_n + H_u h_{n-1}), && \textit{(update gate)} \\ \bar{h}_n &= \tanh(I w_n + H(r_n \cdot h_{n-1})), && \textit{(candidate update)} \\ h_n &= (1 - u_n) \cdot h_{n-1} + u_n \cdot \bar{h}_n, && \textit{(final update)} \end{aligned} \tag{7.3}$$

where σ is the logistic sigmoid, $\sigma(x) \in [0, 1]$, $\cdot$ represents the element-wise scalar product between vectors, $I, I_u, I_r \in \mathbb{R}^{d_h \times V}$ and H, H_r, H_u are in $\mathbb{R}^{d_h \times d_h}$. The I matrices encode the word w_n while the H matrices specialize in retaining or forgetting the information in h_{n-1}. In the following, this function will be noted $GRU(h_{n-1}, w_n)$.

The gates r_n and u_n are computed in parallel. If, given the current word, it is preferable to forget information about the past, i.e. to reset parts of h_n, the elements of r_n will be pushed towards 0. The update gate u_n plays the opposite role, i.e. it judges whether the current word contains relevant information that should be stored in h_n. In the final update, if the elements of u_n are close to 0, the network discards the update $\bar{h}_n$ and keeps the last recurrent state h_{n-1}. The gating behaviour provides robustness to noise in the input sequence: this is particularly important for IR as it allows, for example, to exclude from the summary non-discriminative terms appearing in the query.

7.3.2 Architecture

Our hierarchical recurrent encoder-decoder (HRED) is pictured in Figure 7.3. Given a query in the session, the model encodes the information seen up to that position and tries to predict the following query. The process is iterated throughout all the queries in the session. In the forward pass, the model computes the query-level encodings, the session-level recurrent states and the log-likelihood of each query in the session given the previous ones. In the backward pass, the gradients are computed and the parameters are updated.

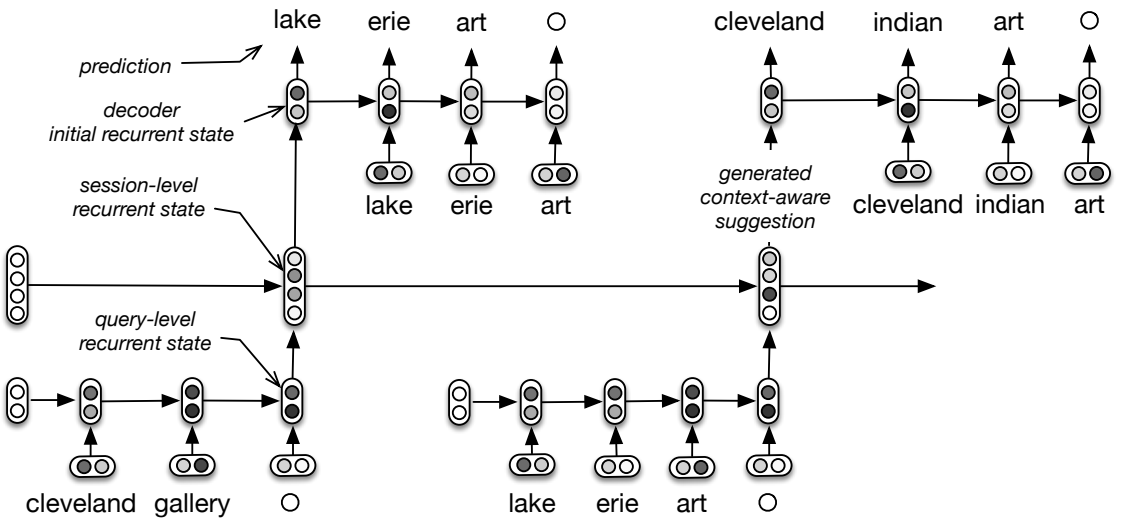


Figure 7.3 – The hierarchical recurrent encoder-decoder (HRED). The user types *cleveland gallery* → *lake erie art*. During training, the model encodes *cleveland gallery*, updates the session-level recurrent state and maximize the probability of seeing the following query *lake erie art*. The process is repeated for all queries in the session. During testing, a contextual suggestion is generated by encoding the previous queries, by updating the session-level recurrent states accordingly and by sampling a new query from the last obtained session-level recurrent state. In the example, the generated contextual suggestion is *cleveland indian art*.

Query-Level Encoding

For each query $Q_m = \{w_{m,1}, \ldots, w_{m,N_m}\}$ in the training session S, the query-level RNN reads the words of the query sequentially and updates its hidden state according to:

$$h_{m,n} = GRU_{enc}(h_{m,n-1}, w_{m,n}), \quad n = 1, \ldots, N_m, \tag{7.4}$$

where GRU_{enc} is the query-level encoder GRU function in Eq. 7.3, $h_{m,n} \in \mathbb{R}^{d_h}$ and $h_{m,0} = 0$, the null vector. The recurrent state h_{m,N_m} is a vector storing order-sensitive information about all the words in the query. To keep the notation uncluttered, we denote $q_m \equiv h_{m,N_m}$ the vector for query m. In summary, the query-level RNN encoder maps a query to a fixed-length vector. Its parameters are shared across the queries. Therefore, the obtained query representation q_m is a general, acontextual representation of query m. The computation of the $q_1, \ldots, q_M$ can be performed in parallel, thus lowering computational costs. A projection of the generated query vectors is provided in Figure 7.1 (b).

Session-Level Encoding

The session-level RNN takes as input the sequence of query representations $q_1, \ldots, q_M$ and computes the sequence of session-level recurrent states. For the session-level RNN, we also use the GRU function:

$$s_m = GRU_{ses}(s_{m-1}, q_m), \quad m = 1, \ldots, M, \tag{7.5}$$

where $s_m \in \mathbb{R}^{d_s}$ is the session-level recurrent state, d_s is its dimensionality and $s_0 = 0$. The number of session-level recurrent states s_m is M, the number of queries in the session.

The session-level recurrent state s_m summarizes the queries that have been processed up to position m. Each s_m bears a particularly powerful characteristic: it is sensitive to the order of previous queries and, as such, it can potentially encode order-dependent reformulation patterns such as generalization or specification of the previous queries (Huang and Efthimiadis, 2009). Additionally, it inherits from the query vectors q_m the sensitivity to the order of words in the queries.

Next-Query Decoding

The RNN decoder is responsible to predict the next query Q_m given the previous queries $Q_{1:m-1}$, i.e. to estimate the probability:

$$P(Q_m|Q_{1:m-1}) = \prod_{n=1}^{N_m} P(w_n|w_{1:n-1}, Q_{1:m-1}). \tag{7.6}$$

The desired conditioning on previous queries is obtained by initializing the recurrence of the RNN decoder with a non-linear transformation of s_{m-1}:

$$d_{m,0} = \tanh(D_0 s_{m-1} + b_0), \tag{7.7}$$

where $d_{m,0} \in \mathbb{R}^{d_h}$ is the decoder initial recurrent state (depicted in Figure 7.3), $D_0 \in \mathbb{R}^{d_h \times d_s}$ projects the context summary into the decoder space and $b_0 \in \mathbb{R}^{d_h}$. This way, the information about previous queries is transferred to the decoder RNN. The recurrence takes the usual form:

$$d_{m,n} = GRU_{dec}(d_{m,n-1}, w_{m,n}), \quad n = 1, \ldots, N_m, \tag{7.8}$$

where GRU_{dec} is the decoder GRU, $d_{m,n} \in \mathbb{R}^{d_h}$ (Cho et al., 2014). In a RNN decoder, each recurrent state $d_{m,n-1}$ is used to compute the probability of the next word $w_{m,n}$. The probability of word $w_{m,n}$ given previous words and queries is:

$$\begin{aligned} P(w_{m,n} = v \,|\, w_{m,1:n-1},\, Q_{1:m-1}) = \\ = \frac{\exp o_v^\top \omega(d_{m,n-1}\,, w_{m,n-1})}{\sum_k \exp o_k^\top \omega(d_{m,n-1}\,, w_{m,n-1})}, \end{aligned} \tag{7.9}$$

where $o_v \in \mathbb{R}^{d_e}$ is the output embedding of word v and ω is a function of both the recurrent state at position n and the last input word:

$$\omega(d_{m,n-1}, w_{m,n-1}) = H_o\, d_{m,n-1} + E_o\, w_{m,n-1} + b_o, \tag{7.10}$$

where $H_o \in \mathbb{R}^{d_e \times d_h}$, $E_o \in \mathbb{R}^{d_e \times V}$ and $b_o \in \mathbb{R}^{d_e}$. To predict the first word of Q_m, we set $w_{m,0} = 0$, the 0 vector. Instead of using the recurrent state directly as in Eq. 7.2, we add another layer of linear transformation ω. The E_o parameter accentuates the responsibility of the previous word to predict the next one. This formulation has shown to be beneficial for language modelling tasks (Pascanu et al., 2013; Cho et al., 2014; Mikolov et al., 2010). If o_v is "near" the vector $\omega(d_{m,n-1}, w_{m,n-1})$ the word v has high probability under the model.

7.3.3 Learning

The model parameters comprise the parameters of the three GRU functions, GRU_{enc}, GRU_{dec}, GRU_{ses}, the output parameters H_o, E_o, b_o and the V output vectors o_i. These are learned by maximizing the log-likelihood of a session S, defined by the probabilities estimated with Eq. 7.6 and Eq 7.9:

$$\begin{aligned} \mathcal{L}(S) &= \sum_{m=1}^{M} \log P(Q_m | Q_{1:m-1}) \\ &= \sum_{m=1}^{M} \sum_{n=1}^{N_m} \log P(w_{m,n} | w_{m,1:n-1}, Q_{1:m-1}). \end{aligned} \tag{7.11}$$

The gradients of the objective function are computed using the back-propagation through time (BPTT) algorithm (Rumelhart et al., 1986).

Context	Synthetic Suggestions
ace series drive	ace hardware ace hard drive hp officejet drive ace hardware series
cleveland gallery → lake erie art	cleveland indian art lake erie art gallery lake erie picture gallery sandusky ohio art gallery

Table 7.1 – HRED suggestions given the context.

7.3.4 Generation and Rescoring

Generation In our framework, the query suggestion task corresponds to an inference problem. A user submits the sequence of queries $S = \{Q_1, \ldots, Q_M\}$. A query suggestion is a query Q^* such that:

$$Q^* = \arg\max_{Q \in \mathcal{Q}} P(Q|Q_{1:M}), \tag{7.12}$$

where $\mathcal{Q}$ is the space of possible queries, i.e. the space of sentences ending by the end-of-query symbol. The solution to the problem can be approximated using standard word-level decoding techniques such as beam-search (Cho et al., 2014; Koehn, 2009). We iteratively consider a set of k best prefixes up to length n as candidates and we extend each of them by sampling the most probable k words given the distribution in Eq. 7.9. We obtain k^2 queries of length $n+1$ and keep only the k best of them. The process ends when we obtain k well-formed queries containing the special end-of-query token $\circ$.

Example Consider a user who submits the queries *cleveland gallery* → *lake erie artist*. The suggestion system proceeds as follows. We apply Eq. 7.4 to each query obtaining the query vectors $q_{\text{cleveland gallery}}$ and $q_{\text{lake erie art}}$. Then, we compute the session-level recurrent states by applying Eq. 7.5 to the query vectors. At this point, we obtain two session-level recurrent states, $s_{\text{cleveland gallery}}$ and $s_{\text{lake erie art}}$. To generate context-aware suggestions, we start by mapping the last session-level recurrent state, $s_{\text{lake erie art}}$, into the initial decoder input d_0 using Eq. 7.7. We are ready to start the sampling of the suggestion. Let assume that the beam-search

size is 1. The probability of the first word w_1 in the suggestion is computed using Eq. 7.9 by using d_0 and $w_0 = 0$, the null vector. The word with the highest probability, i.e. *cleveland*, is added to the beam. The next decoder recurrent state d_1 is computed by means of Eq. 7.8 using d_0 and $w_1 = cleveland$. Using d_1, we are able to pick $w_2 = indian$ as the second most likely word. The process repeats and the model selects *art* and $\circ$. As soon as the end-of-query symbol is sampled, the context-aware suggestion *cleveland indian art* is presented to the user. In Table 7.1 we give an idea of the generated suggestions for 2 contexts in our test set.

Rescoring Our model can evaluate the likelihood of a given suggestion conditioned on the history of previous queries through Eq. 7.6. This makes our model integrable into more complex suggestion systems. In the next section, we choose to evaluate our model by adding the likelihood scores of candidate suggestions as additional features into a learning-to-rank system.

7.4 Experiments

We test how well our query suggestion model can predict the next query in the session given the history of previous queries. This evaluation scenario aims at measuring the ability of a model to propose the target next query, which is assumed to be one desired by the user. We evaluate this with a learning-to-rank approach (explained in Section 7.4.3), similar to the one used in Mitra (2015); Shokouhi (2013) for query auto-completion and in Ozertem et al. (2012); Santos et al. (2013) for query suggestion. We first generate candidates using a co-occurrence based suggestion model. Then, we train a baseline ranker comprising a set of contextual features depending on the history of previous queries as well as pairwise features which depend only on the most recent query. The likelihood scores given by our model are used as additional features in the supervised ranker. At the end, we have three systems: (1) the original co-occurrence based ranking, denoted ADJ; (2) the supervised context-aware ranker, which we refer to as Baseline Ranker; and (3) a supervised ranker with our HRED feature. We evaluate the performance of the model and the baselines using mean reciprocal rank (MRR). This is common for tasks whose ground truth is only one instance (Jiang et al., 2014; Mitra, 2015).

Batches Seen	Training	Decoding (50)	Memory
135,350	44h 01m	$\sim$ 1s	301 Mb

Table 7.2 – Full statistics about training time, memory impact and decoding time with a beam size of 50.

7.4.1 Dataset

We conduct our experiments on the well-known search log from AOL, which is the only available search log that is large enough to train our model and the baselines. The queries in this dataset were sampled between 1 March, 2006 and 31 May, 2006. In total there are 16,946,938 queries submitted by 657,426 unique users. We remove all non-alphanumeric characters from the queries, apply a spelling corrector and lowercasing. After filtering, we sort the queries by timestamp and we use those submitted before 1 May, 2006 as our *background* data to estimate the proposed model and the baselines. The next two weeks of data are used as a *training* set for tuning the ranking models. The remaining two weeks are split into the *validation* and the *test* set. We define the end of a session by a 30 minute window of idle time (Jansen et al., 2007). After filtering, there are 1,708,224 sessions in the background set, 435,705 in the training set, 166,836 in the validation and 230,359 in the test set.

7.4.2 Model Training

The most frequent $90K$ words in the background set form our vocabulary V. This is a common setting for RNN applied to language and allows to speed-up the repeated summations over V in Eq. 7.9 (Cho et al., 2014; Sutskever et al., 2014). Parameter optimization is done using mini-batch RMSPROP (Graves, 2013). We normalize the gradients if their norm exceeds a threshold $c = 1$ (Pascanu et al., 2013). The training stops if the likelihood of the validation set does not improve for 5 consecutive iterations. We train our model using the Theano library (Bastien et al., 2012; Bergstra et al., 2010). The dimensionality of the query-level RNN is set to $d_h = 1000$. To memorize complex information about previous queries, we ensure a high-capacity session-level RNN by setting $d_s = 1500$. The output word embeddings o_i are 300 dimensional vectors, i.e. $d_e = 300$. Our model is

compact and can easily fit in memory (Table 7.2). The complexity of the decoding is largely dominated by the computation of the output probabilities, giving $O(nkVd_e)$, where n is the generated query length and k the beam size. In the future, the computational cost may be greatly reduced by employing locality sensitive hashing (LSH) based techniques (Shrivastava and Li, 2014).

7.4.3 Learning to Rank

Given a session $S = \{Q_1, \ldots, Q_M\}$, we aim to predict the *target* query Q_M given the *context* $Q_1, \ldots, Q_{M-1}$. Q_{M-1} is called the *anchor* query and will play a crucial role in the selection of the candidates to rerank. To probe different capabilities of our model, we predict the next query in three scenarios: (a) when the anchor query exists in the background data (Section 7.4.4); (b) when the context is perturbed with overly common queries (Section 7.4.5); (c) when the anchor is not present in the background data (Section 7.4.6).

For each session, we select a list of 20 possible candidates to rerank. The exact method used to produce the candidates will be discussed in the next sections. Once the candidates are extracted, we label the true target as relevant and all the others as non-relevant. We choose to use one of the state-of-the-art ranking algorithms LambdaMART as our supervised ranker, which is the winner in the Yahoo! Learning to Rank Challenge in 2010 (Wu et al., 2010). We tune the LambdaMART model with 500 trees and the parameters are learnt using standard separate training and validation set. We describe the set of pairwise and contextual features (17 in total) used to train a supervised baseline prediction model, denoted Baseline Ranker. The baseline ranker is a competitive system comprising features that are comparable with the ones described in the literature for query auto-completion (Jiang et al., 2014; Mitra, 2015) and next-query prediction (He et al., 2009).

Pairwise and Suggestion Features For each candidate suggestion, we count how many times it follows the anchor query in the background data and add this count as a feature. Additionally, we use the frequency of the anchor query in the background data. Following Jiang et al. (2014); Ozertem et al. (2012) we also add the Levenshtein distance between the anchor and the suggestion. Suggestion features include: the suggestion length (characters and words) and its frequency in the background set.

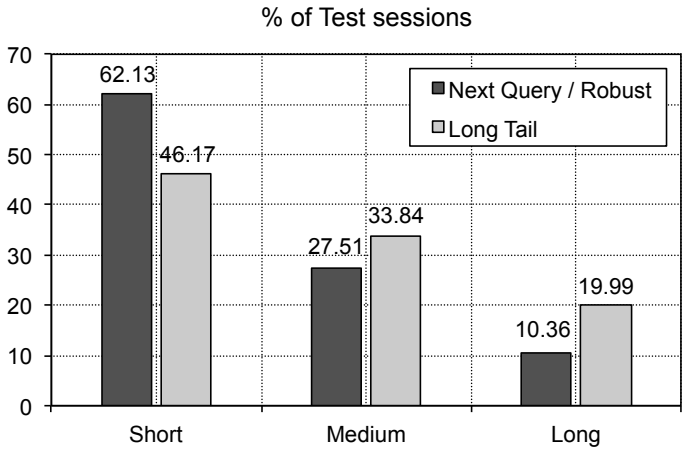


Figure 7.4 – Proportion (%) of short (2 queries), medium (3 or 4 queries) and long (at least 5 queries) sessions in our test scenarios.

Contextual Features Similarly to Mitra (2015); Shokouhi (2013), we add 10 features corresponding to the character n-gram similarity between the suggestion and the 10 most recent queries in the context. We add the average Levenshtein distance between the suggestion and each query in the context (Jiang et al., 2014). We use the scores estimated using the context-aware Query Variable Markov Model (QVMM) (He et al., 2009) as an additional feature. QVMM models the context with a variable memory Markov model able to automatically back-off shorter query n-grams if the exact context is not found in the background data.

HRED Score The proposed hierarchical recurrent encoder-decoder contributes one additional feature corresponding to the log-likelihood of the suggestion given the context, as detailed in Section 7.3.4.

7.4.4 Test Scenario 1: Next-Query Prediction

For each session in the training, validation and test set, we extract 20 queries that most likely follow the anchor query in the background data, i.e. with the highest ADJ score. The session is included if and only if at least 20 queries have been extracted and the target query appears in the candidate list. In that case, the target query is the positive candidate and the 19 other candidates are the negative examples. Note that a similar setting has been used in Jiang et al. (2014); Mitra (2015) for query auto-completion. We have 18,882 sessions in the training, 6,988

Method	MRR	Δ%
ADJ	0.5334	-
Baseline Ranker	0.5563	*+4.3%*
+ HRED	**0.5749**	*+7.8%/+3.3%*

Table 7.3 – Next-query prediction results. All improvements are significant by the t-test ($p < 0.01$).

sessions in the validation and 9,348 sessions in the test set. The distribution of the session length is reported in Figure 7.4. The scores obtained by the ADJ counts are used as an additional non-supervised baseline.

Main Result Table 7.3 shows the MRR performance for our model and the baselines. Baseline Ranker achieves a relative improvement of 4.3% with respect to the ADJ model. We find that the HRED feature brings additional gains achieving 7.8% relative improvement over ADJ. The differences in performance with respect to ADJ and the Baseline Ranker are significant using a t-test with $p < 0.01$. In this general next-query prediction setting, HRED boosts the rank of the first relevant result.

Impact of Session Length We expect the session length to have an impact on the performance of context-aware models. In Figure 7.5, we report separate results for short (2 queries), medium (3 or 4 queries) and long sessions (at least 5 queries). HRED brings statistically significant improvements across all the session lengths. For short sessions, the improvement is marginal but consistent even though only a short context is available in this case. The semantic mapping learnt by the model appears to be useful, even in the pairwise case. ADJ is affected by the lack of context-awareness and suffers a dramatic loss of performance with increasing session length. In the medium range, context-aware models account for previous queries and achieve the highest performance. The trend is not maintained for long sessions, seemingly the hardest for the Baseline Ranker. Long sessions can be the result of complex search tasks involving a topically broad information need or changes of search topics. Beyond the intrinsic difficulty in predicting the target query in these cases, exact context matches may be too coarse to infer the user need. Count-based methods such as QVMM meet their limitations due to data

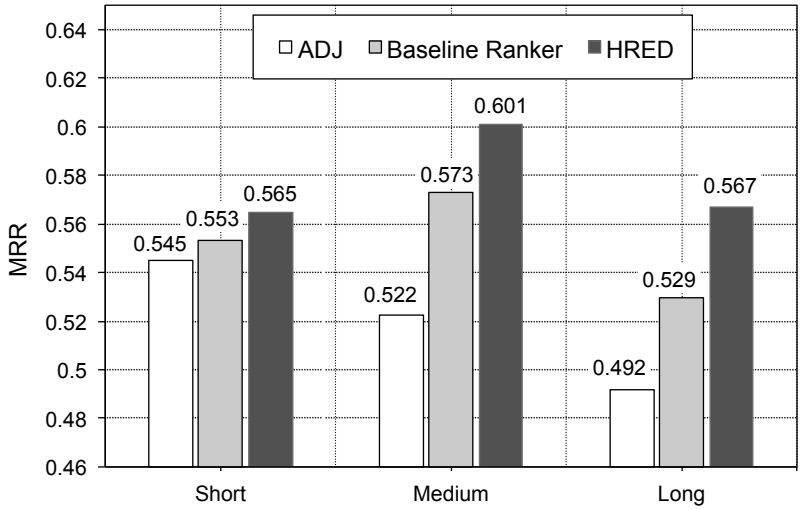


Figure 7.5 – Next-query performance in short, medium and long sessions. All differences in MRR are statistically significant by the t-test ($p < 0.01$).

sparsity. In this difficult range, HRED achieves its highest relative improvement with respect to both ADJ (+15%) and the Baseline Ranker (+7%), thus showing robustness across different session lengths.

Impact of Context Length We test whether the performance obtained by HRED on long sessions can be obtained using a shorter context. For each long session in our test set, we artificially truncate the context to make the prediction depend on the anchor query, Q_{M-1}, only (1 query), on Q_{M-2} and Q_{M-1} (2 queries), on 3 queries and on the entire context. When one query is considered, our model behaves similarly to a pairwise recurrent encoder-decoder model trained on consecutive queries. Figure 7.6 shows that when only one query is considered, the performance of HRED is similar to the Baseline Ranker (0.529) which uses the whole context. HRED appears to perform best when the whole context is considered. Additional gains can be obtained by considering more than 3 queries, which highlights the ability of our model to consider long contexts.

7.4.5 Test Scenario 2: Robust Prediction

Query sessions contain a lot of common and navigational queries such as *google* or *facebook* which do not correspond to a specific search topic. A context-aware

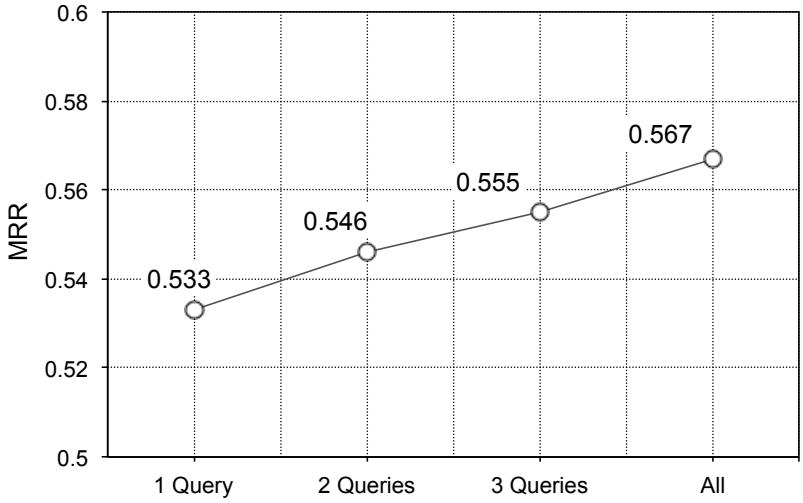


Figure 7.6 – Variation of HRED performance with respect to the number of previous queries considered. The evolution is computed on long sessions.

suggestion system should be *robust* to noisy queries and learn to discard them from the relevant history that should be retained. We propose to probe this capability by formulating an original robust prediction task as follows. We label the 100 most frequent queries in the background set as *noisy*[1]. For each entry in the training, validation and test set of the previous next-query prediction task, we corrupt its context by *inserting* a noisy query at a random position. The candidates and the target rest unchanged. The probability of sampling a noisy query is proportional to its frequency in the background set. For example, given the context *airlines* → *united airlines* and the true target *delta airlines*, the noisy sample *google* is inserted at a random position, forcing the models to predict the target given the corrupted context *airlines* → *united airlines* → *google*.

Main Result Table 7.4 shows that corruption considerably affects the performance of ADJ. Cases in which the corruption occurred at the position of the anchor query severely harm pairwise models. The Baseline Ranker achieves significant gains over ADJ by leveraging context matches. Its performance is inferior to the baseline ADJ performance in the next-query setting reported in Table 7.3 (0.5334). HRED appears to be particularly effective in this difficult setting achieving a relative improvement of 17.8% over ADJ and 9.9% over the Baseline Ranker, both

1. A similar categorization has been proposed in Raman et al. (2014).

Method	MRR	Δ%
ADJ	0.4507	-
Baseline Ranker	0.4831	*+7,2%*
+ HRED	**0.5309**	*+17,8%/+9.9%*

Table 7.4 – Robust prediction results. The improvements are significant by the t-test ($p < 0.01$).

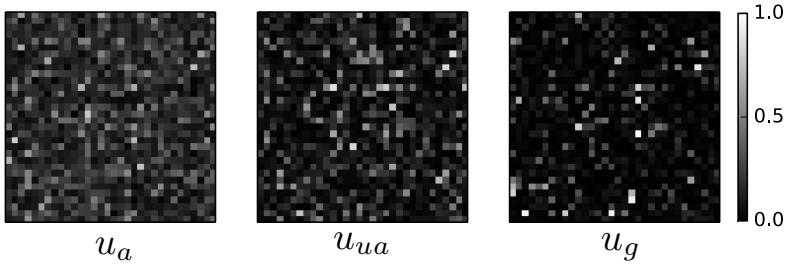


Figure 7.7 – Magnitude of the elements in the session-level update gates. The darker the image, the more the model discards the current query. The vector corresponding to *google*, u_g, is darker, i.e. the network mainly keeps its previous recurrent state.

statistically significant. Comparative to the next-query task, the improvements over ADJ and the Baseline Ranker are 2.5 and 3 times higher respectively. Our model appears to be more robust than the baselines in these extreme cases and can better reduce the impact of the noisy query.

Impact of the Hierarchical Structure As noisy queries bring little information to predict future queries in the session, HRED may automatically learn to be robust to the noise at training time. The hierarchical structure allows to decide, for each query, if it is profitable to account for its contribution to predict future queries. This capability is sustained by the session-level GRU, which can ignore the noisy queries by "turning-off" the update gate u_n when they appear (see Section 7.3.1). Given the corrupted context *airlines* → *united airlines* → *google*, the session-level GRU computes three update gate vectors: u_a, u_{ua}, u_g, each corresponding to a position in the context. In Figure 7.7, we plot the magnitude of the elements in these vectors. As the model needs to memorize the initial information, u_a shows a significant number of non-zero (bright) entries. At this point, general topical information has already been stored in the first recurrent state. Hence, u_{ua} shows a larger number of zero (dark) entries. When *google* is processed, the

Method	MRR	Δ%
ADJ	0.3830	-
Baseline Ranker	0.6788	*+77.2%*
+ HRED	**0.7112**	*+85.3%/+5.6%*

Table 7.5 – Long-tail prediction results. All improvements are significant by the t-test ($p < 0.01$).

network tends to keep past information in memory by further zeroing entries in the update gate. This sheds an interesting perspective: this mechanism may be used to address other search log related tasks such as session-boundary detection.

7.4.6 Test Scenario 3: Long-Tail Prediction

To analyze the performance of the models in the long-tail, we build our training, validation and test set by retaining the sessions for which the anchor query has not been seen in the background set, i.e. it is a long-tail query. In this case, we cannot leverage the ADJ score to select candidates to rerank. For each session, we iteratively shorten the anchor query by dropping terms until we have a query that appears in the background data. If a match is found, we proceed as described in the next-query prediction setting, that is, we guarantee that the target appears in the top 20 queries that have the highest ADJ scores given the anchor prefix. The statistics of the obtained dataset are reported in Figure 7.4. As expected, the distribution of lengths changes substantially with respect to the previous settings. Long-tail queries are likely to appear in medium and long sessions, in which the user strives to find an adequate textual query.

Main Result Table 7.5 shows that, due to the anchor prefix matching, ADJ suffer a significant loss of performance. The performances of the models generally confirm our previous findings. HRED improves significantly by 5.6% over the Baseline Ranker and proves to be useful even for long-tail queries. Supervised models appear to achieve higher absolute scores in the long-tail setting than in the general next-query setting reported in Table 7.3. After analysis of the long-tail testing set, we found that only 8% of the session contexts contain at least one noisy query. In the general next-query prediction case, this number grows to 37%. Noisy queries

generally harm performance of the models by increasing the ambiguity in the next query prediction task. This fact may explain why the Baseline ranker and HRED perform better on long-tail queries than in the general case. It is interesting to see how the improvement of HRED with respect to the Baseline Ranker is larger for long-tail queries than in the general setup (5.6% to 3.3%). Although not explicitly reported, we analyzed the performance with respect to the session length in the long-tail setting. Similarly to the general next-query prediction setting, we found that the Baseline Ranker suffers significant losses for long sessions while our model appears robust to different session lengths.

7.4.7 User Study

The previous re-ranking setting doesn't allow to test the generative capabilities of our suggestion system. We perform an additional user study and ask human evaluators to assess the quality of synthetic suggestions. To avoid sampling bias towards overly common queries, we choose to generate suggestions for the 50 topics of the TREC Web Track 2011 Clarke et al. (2011). The assessment was conducted by a group of 5 assessors. To palliate the lack of context information for TREC queries, we proceed as follows: for each TREC topic Q_M, we extract from the test set the sessions ending exactly with Q_M and we take their context $Q_1, \ldots, Q_{M-1}$. After contextualization, 19 TREC queries have one or more queries as context and the remaining are singletons. For HRED, we build synthetic queries following the generative procedure described in Section 7.3.4. In addition to QVMM and ADJ, we compare our model with two other baselines: CACB (Cao et al., 2008), which is similar to QVMM but builds clusters of queries to avoid sparsity, and SS (Search Shortcuts) (Broccolo et al., 2012), which builds an index of the query sessions and extracts the last query of the most similar sessions to the source context. Note that we do not compare the output of the previous supervised rankers as this would not test the generative capability of our model. Each assessor was provided with a random query from the test bed, its context, if any, and a list of recommended queries (the top-5 for each of the methods) selected by the different methods. Recommendations were randomly shuffled, so that the assessors could not distinguish which method produced them. Each assessor was asked to judge each recommended query using the following scale: useful, somewhat useful, and not useful. The user study

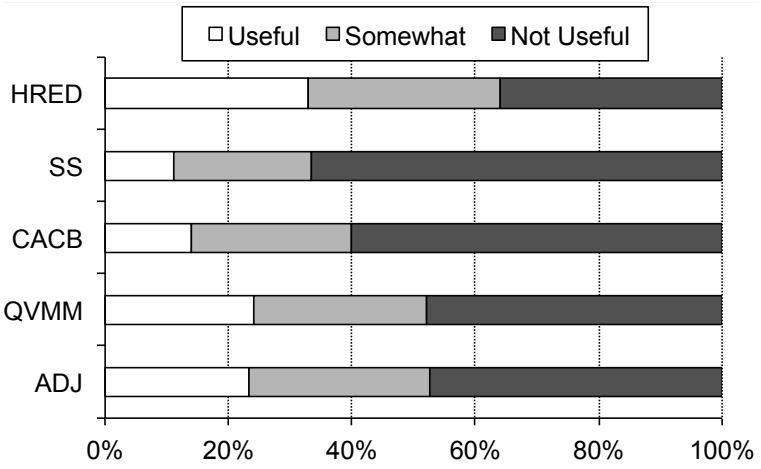


Figure 7.8 – User study results, which compare the effectiveness of HRED with the baseline techniques.

finished when each assessor had assessed all recommendations for all 50 queries in the test bed. Figure 7.8 reports the results of the user study averaged over all raters. Overall, for HRED, 64% of the recommendations were judged useful or somewhat useful. The quality of the queries recommended by HRED is higher than our baselines both in the somewhat and in the useful category.

7.5 Related Works

Query Suggestion A notorious context-aware method was proposed in He et al. (2009). The authors use a Variable Memory Markov model (QVMM) and build a suffix tree to model the user query sequence. We used this model as a context-aware baseline feature in our supervised ranker. The method by Cao et al. (2008) is similar but they build a suffix tree on clusters of queries and model the transitions between clusters. We didn't notice any improvements by adding this model as a feature in our case. For both models, the number of parameters increases with the depth of the tree inducing sparsity. Instead, our model can consider arbitrary length contexts with a fixed number of parameters. Jiang et al. (2014) and Shokouhi (2013) propose context-aware approaches for query auto-completion. We adopted a similar framework for query suggestion and use our model as a feature to rank the next-query. Santos et al. (2013) and Ozertem et al. (2012) also use learning to

rank approach for query suggestion. In those cases, the rankers are trained using pairwise features and do not consider previous queries. Interestingly, the authors model explicitly the usefulness of a suggestion by using click data and the result list.

Query suggestion algorithms use clustering methods to find similar queries so that they can be used as suggestions for one another (Baeza-Yates et al., 2004; Wen et al., 2001). We demonstrated that our model exhibits similar clustering properties due to the embeddings learnt by the neural network.

Other works build a Query Flow Graph (QFG) to capture high-order query co-occurrence (Boldi et al., 2008; Sadikov et al., 2010). Operating at the query-level, these methods suffer from the long-tail problem. Bonchi et al. (2012) propose a solution to these problems by introducing the Term-QFG (TQG), where single query terms are also included into the graph. However, suggestion requires repeated complex random walks with restart. Similarly, our model can handle rare queries as long as their words appear in the model vocabulary. Vahabi et al. (2013) find suggestions to long-tail queries by comparing their search results. Although effective, the approach requires to have 100 results per query. A related approach is the Search Shortcut (SS) (Broccolo et al., 2012) which avoids the long-tail problem by means of a retrieval algorithm.

Few synthetic suggestion models have been proposed in the literature. Szpektor et al. (2011) use a template generation method by leveraging WordNet. Jain et al. (2011) combine different resources and use a machine learning approach to prune redundant suggestions. These methods achieve automatic addition, removal and substitution of related terms into the queries. By maximizing the likelihood of the session data, our model learns to perform similar modifications.

Neural Networks for NLP Neural networks have found several applications in a variety of tasks, ranging from Information Retrieval (IR) (Huang et al., 2013; Shen et al., 2014), Language Modeling (LM) (Mikolov et al., 2010; Pascanu et al., 2013) and Machine Translation (MT) (Cho et al., 2014; Sutskever et al., 2014). Cho et al. (2014) and Sutskever et al. (2014) use a Recurrent Neural Network (RNN) for end-to-end MT. Our model bears similarities to these approaches but we contribute with the hierarchical structure. The idea of encoding hierarchical multi-scale representations is also explored in Hihi and Bengio (1995). In IR, neural

networks embeddings were used in Li et al. (2014). The authors used deep feed-forward neural networks to use previous queries by the same user to boost document ranking. In Huang et al. (2013); Shen et al. (2014), the authors propose to use clickthrough data to learn a ranking model for ad-hoc IR. Recently, Grbovic et al. (2015) used query embeddings to include session-based information for sponsor search. Our model shares similarities with the interesting recent work by Mitra (2015). The authors use the pairwise neural model described in Shen et al. (2014) to measure similarity between queries. Context-awareness is achieved at ranking time, by measuring the similarity between the candidates and each query in the context. Our work has several key differences. First, we deploy a novel RNN architecture. Second, our model is generative. Third, we model the session context at training time. To our knowledge, this is the first work applying RNNs to an IR task.

7.6 Conclusion

In this paper, we formulated a novel hierarchical neural network architecture and used it to produce query suggestions. Our model is context-aware and it can handle rare queries. It can be trained end-to-end on query sessions by simple optimization procedures. Our experiments show that the scores provided by our model help improving MRR for next-query ranking. Additionally, it is generative by definition. We showed with a user study that the synthetic generated queries are better than the compared methods.

In future works, we aim to extend our model to explicitly capture the usefulness of a suggestion by exploiting user clicks (Ozertem et al., 2012). Then, we plan to further study the synthetic generation by means of a large-scale automatic evaluation. Currently, the synthetic suggestions tend to be *horizontal*, i.e. the model prefers to add or remove terms from the context queries and rarely proposes orthogonal but related reformulations (Jain et al., 2011; Vahabi et al., 2013). Future efforts may be dedicated to diversify the generated suggestions to account for this effect. Finally, the interactions of the user with previous suggestions can also be leveraged to better capture the behaviour of the user and to make better suggestions accordingly. We are the most excited about possible future applications

beyond query suggestion: auto-completion, next-word prediction and other NLP tasks such as Language Modelling may be fit as possible candidates.

8 General Conclusion

This thesis presented four articles dealing with the estimation of documents and queries representations. The first two articles can be categorized as feature engineering approaches, which transduce a priori knowledge about the domain into the features of the representation. The remaining two articles originate from the widespread interest in deep learning research that took place during the time they were written. Therefore, they naturally belong to the category of representation learning approaches, also known as feature learning, which let the learning model discover the most important features for the task at hand.

Initially, we investigated the possibilities opened by applying the density matrix formalism to IR, which mixes probability theory with linear algebra. For this reason, it appealed to us as an appropriate framework to glean insights on both probabilistic and geometric retrieval models. After shedding light on the assumptions behind well-known retrieval models, we proposed the Quantum Language Model (QLM), a novel retrieval model capturing the presence of words and phrases into an holistic representation. The success of the QLM is attributable in part to the fact that phrases are not considered as additional indexing units but related to their component terms.

Our study of past IR literature taught us that each retrieval model may easily be extended by employing well-known techniques: query expansion or relevance feedback may be used to tackle the semantic gap between queries and documents while phrase selection algorithms may help in identifying which phrases are most discriminative for relevance. Our view is that their application to a novel retrieval model only brings incremental improvements to the state-of-the-art and is likely to generate limited impact on future research. Instead, we tracked the rising interest towards deep learning research in hopes of dealing with old problems in radically new ways.

Following this line of thought, we proposed a word and phrase embedding model for IR, which takes inspiration from the mathematics of QLMs but carves query

and document representations in order to bring relevance estimation to a more semantic level. Both documents and queries are special matrices within a vector-space whose dimensions are semantic descriptors rather than single atomic terms. Finally, we moved further into the deep learning of representations by considering the task of query suggestions. In this case, in-context query representations are obtained by leveraging a new deep learning architecture.

Representation learning methods are extremely flexible and have the potential to bring dramatic improvements to the state-of-the-art in IR. However, one lesson that can be gleaned from our articles is that, in order to obtain good results, it is necessary to mix the learned semantic representations with standard retrieval techniques. This shows that the current models based on deep learning are still unable to capture some important characteristics of traditional models. An interesting future research topic is to understand which characteristics are missing in deep learning models and how to incorporate them into these models. We identify two possible improvements that can help in bringing robust performance. First of all, it is necessary to augment the capacity of the representation learning models. Compressing entire documents and queries into a single vector representation may not be sufficient to reflect the complexity of their information content. Recent developments critically augmented the capacity of deep learning architectures by employing neural attention techniques and dynamic memory modules. It would be interesting to apply these more complex representational models to IR. Second, currently, it is crucial to get access to large amounts of high-quality labeled data. The scarcity of publicly available relevance data makes it difficult to train reliable high-capacity deep learning models in an academic setting. Approximations to the true relevance labels are possible but often weakly correlated to the true underlying distribution. In academia, it is necessary to investigate new ways to learn robust models in this low-data regime.

Overall, we believe that information retrieval may greatly benefit from the study and application of representation learning techniques. In our two last papers, the models based on learned representations showed some capability to capture important hidden characteristics, which would be difficult to define manually. As IR systems become more complex and take into account more signals, it becomes very difficult to manually design a good representation for documents and queries. We see the representation learning techniques as an effective way to cope with such a

complex situation. These may allow, for example, to move retrieval models closer to artificial intelligence agents capable of simulating the human relevance assessment process. Conversely, information retrieval researchers may benefit the entire representation learning field by addressing weaknesses of existing learning techniques when applied to IR challenges. Representation learning is a fast-developing field and has already set unrivaled state-of-the-art in a variety of domains including computer vision and machine translation. Overall, we hope this thesis will help in closing the gap between representation learning and information retrieval and will foster future research in this direction.

A List of articles published during the thesis

I. Serban, **A. Sordoni**, Y. Bengio, A. Courville and J. Pineau. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Networks. In Proc of AAAI, 2016.

A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J.G. Simonsen and J.-Y. Nie. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion, In Proc of CIKM, 2015. *Presented in Chapter 7.*

M. Galley, C. Brockett, **A. Sordoni**, Y. Ji, M. Auli, C. Quirk, M. Mitchell, J. Gao, B. Dolan. deltaBLEU: A Discriminative Metric for Generation Tasks with Intrinsically Diverse Targets, In Proc of ACL, 2015.

A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, B. Dolan. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In Proc of NAACL-HLT, 2015

A. Sordoni, Y. Bengio and J.-Y. Nie. Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization. In Proc of AAAI, 2014. *Presented in Chapter 6.*

X. Liu, A. Bouchoucha, **A. Sordoni** and J.-Y. Nie. Compact Aspect Embedding For Diversified Query Expansion. In Proc of AAAI, 2014

A. Sordoni, J.Y.-Nie. Looking at Vector Space and Language Models for IR using Density Matrices. In Proc of QI, 2014. *Presented in Chapter 4.*

A. Sordoni, W. Yuan and J.-Y. Nie. Université de Montréal at TREC 2013: Experiments with Quantum Language Models in the Web Track. In Proc of TREC, 2013. *Presented in Chapter 5.*

A. Sordoni, J.Y.-Nie and Y. Bengio. Modeling Term Dependencies with Quantum Language Models for IR. In Proc of SIGIR, pages 653-662, 2013. *Presented in Chapter 5.*

A. Sordoni, J. He, J.Y.-Nie. Modeling Latent Topic Interactions using Quantum Interference for IR. In Proc of CIKM, pages 1197-1200, 2013

G. M. di Nunzio, **A. Sordoni**. Picturing Bayesian Classifiers. In Data Mining Applications with R, Elsevier, 2013

G. M. di Nunzio, **A. Sordoni**. How Well Do We Know Bernoulli? In Proc of IIR, pages 38-44, 2012

G. M. di Nunzio, **A. Sordoni**. A Visual Tool for Bayesian Data Analysis: the Impact of Smoothing on Naive-Bayes Text Classifiers. In Proc of SIGIR pages 1002, 2012

Bibliography

Accardi, L. (1984). The probabilistic roots of the quantum mechanical paradoxes. In *The wave-particle dualism*, pp. 297–330. Springer.

Aerts, D. and S. Sozzo (2011). Quantum structure in cognition: Why and how concepts are entangled. In *Quantum interaction*, pp. 116–127. Springer.

Arguello, J., J. L. Elsas, J. Callan, and J. G. Carbonell (2008). Document representation and query expansion models for blog recommendation. In E. Adar, M. Hurst, T. Finin, N. S. Glance, N. Nicolov, and B. L. Tseng (Eds.), *ICWSM*. The AAAI Press.

Atreya, A. and C. Elkan (2011, March). Latent semantic indexing (LSI) fails for TREC collections. *SIGKDD Explor. Newsl. 12*(2), 5–10.

Baeza-Yates, R., C. Hurtado, and M. Mendoza (2004). Query recommendation using query logs in search engines. In *In Proc. of Int. Conf. on Current Trends in Database Tech.*, pp. 588–596.

Bai, B., J. Weston, R. Collobert, and D. Grangier (2009). Supervised semantic indexing. In M. Boughanem, C. Berrut, J. Moth, and C. Soulé-Dupuy (Eds.), *ECIR*, Volume 5478 of *Lecture Notes in Computer Science*, pp. 761–765. Springer.

Bai, J., Y. Chang, H. Cui, Z. Zheng, G. Sun, and X. Li (2008). Investigation of partial query proximity in web search. In *Proceedings of the 17th international conference on World Wide Web*, pp. 1183–1184.

Balkir, E., M. Sadrzadeh, and B. Coecke (2016). *Topics in Theoretical Computer Science: The First IFIP WG 1.8 International Conference, TTCS 2015, Tehran, Iran, August 26-28, 2015, Revised Selected Papers*, Chapter Distributional Sentence Entailment Using Density Matrices, pp. 1–22. Cham: Springer International Publishing.

Bastien, F., P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

Bendersky, M. and W. B. Croft (2012). Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 941–950.

Bendersky, M., D. Metzler, and W. B. Croft (2010). Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, New York, NY, USA, pp. 31–40. ACM.

Bendersky, M., D. Metzler, and W. B. Croft (2011). Parameterized concept weighting in verbose queries. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 605–614.

Bengio, Y. (2013). *Statistical Language and Speech Processing: First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013. Proceedings*, Chapter Deep Learning of Representations: Looking Forward, pp. 1–37. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bengio, Y., R. Ducharme, and P. Vincent (2003). A neural probabilistic language model. *Journal of Machine Learning Research 3*, 1137–1155.

Bengio, Y., H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain (2006). Neural probabilistic language models. In *Innovations in Machine Learning*, pp. 137–186. Springer.

Bengio, Y., P. Simard, and P. Frasconi (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 157–166.

Bergstra, J. and Y. Bengio (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13*, 281–305.

Bergstra, J., O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio (2010). Theano: a CPU and GPU math expression compiler. In *In Proc. of SciPy*.

Birkhoff, G. and J. V. Neumann (1936, October). The logic of quantum mechanics. *The Annals of Mathematics 37*(4), 823.

Blei, D. M. (2012, April). Probabilistic topic models. *Commun. ACM 55*(4), 77–84.

Blei, D. M., A. Y. Ng, M. I. Jordan, and J. Lafferty (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research 3*, 2003.

Blume-Kohout, R. (2010, January). Hedged maximum likelihood estimation. *arXiv:1001.2029*. Phys. Rev. Lett. 105, 200504 (2010).

Bodoff, D. and S. Robertson (2004). A new unified probabilistic model. *Journal of the American Society for Information Science and Technology 55*(6), 471–487.

Boldi, P., F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna (2008). The query-flow graph: Model and applications. In *In Proc. of CIKM*, pp. 609–618.

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, New York, NY, USA, pp. 1247–1250. ACM.

Bonchi, F., R. Perego, F. Silvestri, H. Vahabi, and R. Venturini (2012). Efficient query recommendations in the long tail via center-piece subgraphs. In *In Proc. of SIGIR*, pp. 345–354.

Bravo-Marquez, F., G. L'Huillier, S. A. Ríos, and J. D. Velásquez (2010). Hypergeometric language model and Zipf-like scoring function for web document similarity retrieval. In *Proceedings of the 17th international conference on String processing and information retrieval*, SPIRE'10, Berlin, Heidelberg, pp. 303–308. Springer-Verlag.

Broccolo, D., L. Marcon, F. M. Nardini, R. Perego, and F. Silvestri (2012). Generating suggestions for queries in the long tail with an inverted index. *Inf. Process. Manage. 48*(2), 326–339.

Bruza, P., K. Kitto, D. Nelson, and C. McEvoy (2009). Is there something quantum-like about the human mental lexicon? *Journal of mathematical psychology 53*(5), 363–377.

Bruza, P. D. and R. Cole (2005). Quantum logic of semantic space: An exploratory investigation of context effects in practical reasoning. *We Will Show Them: Essays in Honour of Dov Gabbay*, 339–361.

Buccio, E. D., M. Lalmas, and M. Melucci (2010). From entities to geometry: Towards exploiting multiple sources to support relevance prediction. In *IIR'10*, pp. 35–39.

Busemeyer, J. R., E. M. Pothos, R. Franco, and J. S. Trueblood (2011). A quantum theoretical explanation for probability judgment errors. *Psychological Review 118*(2), 193.

Cao, H., D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li (2008). Context-aware query suggestion by mining click-through and session data. In *In Proc. of SIGKDD*, pp. 875–883.

Carpineto, C. and G. Romano (2012, January). A survey of automatic query expansion in information retrieval. *ACM Comput. Surv. 44*(1), 1:1–1:50.

Chapelle, O., D. Metlzer, Y. Zhang, and P. Grinspan (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 621–630. ACM.

Chelba, C. and F. Jelinek (2000, October). Structured language modeling. *Computer Speech & Language 14*(4), 283–332.

Cho, K., B. Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *In Proc. of EMNLP*.

Clarke, C. L., N. Craswell, I. Soboroff, and E. M. Voorhees (2011). Overview of the trec 2011 web track. *Proceedings of the 2011 Text Retrieval Conference (TREC 2011)*.

Collins-Thompson, K. and E. M. Voorhees (2013). TREC 2013 web track overview.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011, November). Natural language processing (almost) from scratch. *J. Mach. Learn. Res. 999888*, 2493–2537.

Cooper, W. S. (1995, January). Some inconsistencies and misidentified modeling assumptions in probabilistic information retrieval. *ACM Trans. Inf. Syst. 13*(1), 100–111.

Craswell, N., O. Zoeter, M. Taylor, and B. Ramsey (2008). An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 87–94. ACM.

Croft, W. B. (2002). Combining approaches to information retrieval. In W. B. Croft (Ed.), *Advances in Information Retrieval*, Volume 7, pp. 1–36. Boston: Kluwer Academic Publishers.

Croft, W. B. and D. J. Harper (1979). Using probabilistic models of document retrieval without relevance information. *Journal of documentation 35*(4), 285–295.

Cui, H., J.-R. Wen, J.-Y. Nie, and W.-Y. Ma (2002). Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, New York, NY, USA, pp. 325–332. ACM.

Cummins, R. and C. O'Riordan (2009). Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 251–258.

Cummins, R., C. O'Riordan, and M. Lalmas (2010). An analysis of learned proximity functions. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pp. 168–171.

Dang, V. and W. B. Croft (2010). Query reformulation using anchor text. In B. D. D. 0001, T. Suel, N. Craswell, and B. L. 0001 (Eds.), *WSDM*, pp. 41–50. ACM.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, 391–407.

Ding, C., T. Li, and W. Peng (2008). On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis 52*(8), 3913–3927.

Eickhoff, C., A. P. de Vries, and T. Hofmann (2015). Modelling term dependence with copulas. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 783–786. ACM.

Fagan, J. L. (1987). Experiments in automatic phrase indexing for document retrieval: A comparison of syntactic and non-syntactic methods. Technical report, Ithaca, NY, USA.

Gao, J., X. He, and J.-Y. Nie (2010). Clickthrough-based translation models for web search: from word models to phrase models. In J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K. Collins-Thompson, and A. An (Eds.), *CIKM*, pp. 1139–1148. ACM.

Gao, J. and J.-Y. Nie (2012). Towards concept-based translation models using search logs for query expansion. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, New York, NY, USA, pp. 1:1–1:10. ACM.

Gao, J., J. Y. Nie, G. Wu, and G. Cao (2004). Dependence language model for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 170–177.

Gleason, A. (1957). Measures on the closed subspaces of a Hilbert space. *Journal of Mathematics and Mechanics 6*, 885–893.

Goodfellow, I., Y. Bengio, and A. Courville (2016). Deep learning. Book in preparation for MIT Press.

Graves, A. (2013). Generating sequences with recurrent neural networks. *CoRR abs/1308.0850*.

Grbovic, M., N. Djuric, V. Radosavljevic, F. Silvestri, and N. Bhamidipati (2015). Context- and content-aware embeddings for query rewriting in sponsored search. In *Proc. of SIGIR*.

Hazimeh, H. and C. Zhai (2015). Axiomatic analysis of smoothing methods in language models for pseudo-relevance feedback. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, New York, NY, USA, pp. 141–150. ACM.

He, Q., D. Jiang, Z. Liao, S. C. H. Hoi, K. Chang, E. P. Lim, and H. Li (2009). Web query recommendation via sequential query prediction. In *In Proc. of ICDE*, pp. 1443–1454.

Heskes, T. (2002). Stable fixed points of loopy belief propagation are local minima of the bethe free energy. In *Advances in neural information processing systems*, pp. 343–350.

Hihi, S. E. and Y. Bengio (1995). Hierarchical recurrent neural networks for long-term dependencies. In *NIPS*, pp. 493–499.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural computation 9*(8), 1735–1780.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 289–296. Morgan Kaufmann Publishers Inc.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning 42*(1), 177–196.

Hou, Y., X. Zhao, D. Song, and W. Li (2013). Mining pure high-order word associations via information geometry for information retrieval. *ACM Transactions on Information Systems (TOIS) 31*(3), 12.

Hoyer, P. O. (2004, December). Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res. 5*, 1457–1469.

Huang, J. and E. N. Efthimiadis (2009). Analyzing and evaluating query reformulation strategies in web search logs. In *In Proc. of CIKM*, pp. 77–86.

Huang, P.-S., X. He, J. Gao, L. Deng, A. Acero, and L. Heck (2013). Learning deep structured semantic models for web search using clickthrough data. In Q. He, A. Iyengar, W. Nejdl, J. Pei, and R. Rastogi (Eds.), *CIKM*, pp. 2333–2338. ACM.

Huston, S. and W. B. Croft (2014). A comparison of retrieval models using term dependencies. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 111–120. ACM.

Jain, A., U. Ozertem, and E. Velipasaoglu (2011). Synthesizing high utility suggestions for rare web search queries. In *In Proc. of SIGIR*, pp. 805–814.

Jansen, B. J., A. Spink, C. Blakely, and S. Koshman (2007, April). Defining a session on web search engines: Research articles. *J. Am. Soc. Inf. Sci. Technol. 58*(6), 862–871.

Järvelin, K. and J. Kekäläinen (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS) 20*(4), 422–446.

Jiang, J., Y. Ke, P. Chien, and P. Cheng (2014). Learning user reformulation behavior for query auto-completion. In *In Proc. of SIGIR*, pp. 445–454.

Jones, K. S., S. Walker, and S. E. Robertson (1998). *A Probabilistic Model of Information Retrieval: Development and Status*, Volume Department of Information Science, City University, London 74.

Jones, K. S., S. Walker, and S. E. Robertson (2000a, November). A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage. 36*(6), 779–808.

Jones, K. S., S. Walker, and S. E. Robertson (2000b). A probabilistic model of information retrieval: development and comparative experiments. In *Information Processing and Management*, pp. 779–840.

Kaszkiel, M. and J. Zobel (2001). Effective ranking with arbitrary passages. *Journal of the American Society for Information Science and Technology 52*(4), 344–364.

Kaszkiel, M., J. Zobel, and R. Sacks-Davis (1999, October). Efficient passage ranking for document databases. *ACM Trans. Inf. Syst. 17*(4), 406–439.

Khrennikov, A. (2007). Bell's inequality: Physics meets probability. *Arxiv preprint ArXiv:0709.3909*.

Kitto, K., B. Ramm, P. Bruza, and L. Sitbon (2010). Testing for the non-separability of bi-ambiguous compounds. In *Proceedings of the AAAI Fall Symposium on Quantum Informatics for Cognitive, Social, and Semantic Processes (QI 2010), edited by.*

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Kontostathis, A. and W. M. Pottenger (2006, January). A framework for understanding latent semantic indexing (LSI) performance. *Information Processing & Management 42*(1), 56–73.

Koolen, W. M., W. Kotłowski, and M. K. Warmuth (2011). Learning eigenvectors for free. *In NIPS*, 945–953.

Kotov, A. and C. Zhai (2012). Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries. In E. Adar, J. Teevan, E. Agichtein, and Y. Maarek (Eds.), *WSDM*, pp. 403–412. ACM.

Lafferty, J. and C. Zhai (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, New York, NY, USA, pp. 111–119. ACM.

Lavrenko, V. (2004). *A generative theory of relevance*. Ph. D. thesis, University of Massachusetts Amherst.

Lavrenko, V. and W. B. Croft (2001). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 120–127.

Lee, C., G. G. Lee, and M. G. Jang (2006). Dependency structure applied to language modeling for information retrieval. *ETRI journal 28*(3), 337–346.

Lee, D. D. and H. S. Seung (2000). Algorithms for non-negative matrix factorization. In *NIPS*, pp. 556–562.

Levy, O. and Y. Goldberg (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 2177–2185.

Li, H. (2011, April). Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies 4*(1), 1–113.

Li, Q., J. Li, P. Zhang, and D. Song (2015). Modeling multi-query retrieval tasks using density matrix transformation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 871–874. ACM.

Li, X., C. Guo, W. Chu, Y. Wang, and J. Shavlik (2014). Deep learning powered in-session contextual ranking using clickthrough data. In *In Proc. of NIPS*.

Lioma, C., J. G. Simonsen, B. Larsen, and N. D. Hansen (2015). Non-compositional term dependence for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, New York, NY, USA, pp. 595–604. ACM.

Liu, X., A. Bouchoucha, A. Sordoni, and J.-Y. Nie (2014). Compact aspect embedding for diversified query expansionsquery expansions. In *Proc. of AAAI*.

Losada, D. E. and L. Azzopardi (2008, June). Assessing multivariate Bernoulli models for information retrieval. *ACM Trans. Inf. Syst. 26*(3), 17:1–17:46.

Lu, X. (2015). Improving search using proximity-based statistics. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, New York, NY, USA, pp. 1065–1065. ACM.

Lu, X., A. Moffat, and J. S. Culpepper (2014). How effective are proximity scores in term dependency models? In *Proceedings of the 2014 Australasian Document Computing Symposium*, pp. 89. ACM.

Lu, Y., Q. Mei, and C. Zhai (2010, August). Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval 14*(2), 178–203.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development 1*(4), 309–317.

Lv, Y. and C. Zhai (2009a). A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1895–1898. ACM.

Lv, Y. and C. Zhai (2009b). Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, New York, NY, USA, pp. 299–306. ACM.

Lvovsky, A. I. (2003, November). Iterative maximum-likelihood reconstruction in quantum homodyne tomography. *arXiv:quant-ph/0311097*. Journal of Optics B: Quantum and Semiclassical Optics 6 (2004) S556–S559.

Maisonnasse, L., E. Gaussier, and J. P. Chevallet (2007). Revisiting the dependence language model for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 695–696.

Maxwell, K. T. and W. B. Croft (2013). Compact query term selection using topically related text. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 583–592. ACM.

Melucci, M. (2008, June). A basis for information retrieval in context. *ACM Trans. Inf. Syst. 26*, 14:1–14:41.

Melucci, M. (2010). An investigation of quantum interference in information retrieval. In H. Cunningham, A. Hanbury, and S. Rüger (Eds.), *Advances in Multidisciplinary Retrieval*, Volume 6107, pp. 136–151. Berlin, Heidelberg: Springer Berlin Heidelberg.

Melucci, M. (2013). Deriving a quantum information retrieval basis. *The Computer Journal 56*(11), 1279–1291.

Melucci, M. (2015). *Introduction to Information Retrieval and Quantum Mechanics*. Springer.

Melucci, M. and K. Rijsbergen (2011). Quantum mechanics and information retrieval. In M. Melucci, R. Baeza-Yates, and W. B. Croft (Eds.), *Advanced Topics in Information Retrieval*, Volume 33 of *The Information Retrieval Series*, pp. 125–155. Springer Berlin Heidelberg.

Metzler, D. (2011). *A feature-centric view of information retrieval*, Volume 27 of *The Kluwer International Series on Information Retrieval*. Springer Science & Business Media.

Metzler, D. and W. Bruce Croft (2007). Linear feature-based models for information retrieval. *Inf. Retr. 10*(3), 257–274.

Metzler, D. and W. B. Croft (2004). Combining the language model and inference network approaches to retrieval. *Information processing & management 40*(5), 735–750.

Metzler, D. and W. B. Croft (2005). A Markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, New York, NY, USA, pp. 472–479. ACM.

Metzler, D. and W. B. Croft (2007). Latent concept expansion using Markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 311–318. ACM.

Metzler, D., V. Lavrenko, and W. B. Croft (2004). Formal multiple-Bernoulli models for language modeling. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 540–541.

Mikolov, T., M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur (2010). Recurrent neural network based language model. In *In Proc. of ACISCA*, pp. 1045–1048.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Mitra, B. (2015). Exploring session context using distributed representations of queries and reformulations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12. ACM.

Mitra, M., C. Buckley, A. Singhal, and C. Cardie (1997). An analysis of statistical and syntactic phrases. In L. Devroye and C. Chrisment (Eds.), *RIAO*, pp. 200–217.

Mnih, A. and K. Kavukcuoglu (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*, pp. 2265–2273.

Morgan, W., W. Greiff, and J. Henderson (2004). Direct maximization of average precision by hill-climbing, with a comparison to a maximum entropy approach. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04, Stroudsburg, PA, USA, pp. 93–96. Association for Computational Linguistics.

Morin, F. and Y. Bengio (2005). Hierarchical probabilistic neural network language model. In *AISTATS'05*, pp. 246–252.

Nallapati, R. and J. Allan (2002). Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the eleventh international conference on Information and knowledge management*, New York, NY, USA, pp. 383–390. ACM.

Nielsen, M. A. and I. L. Chuang (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.

Ozertem, U., O. Chapelle, P. Donmez, and E. Velipasaoglu (2012). Learning to suggest: A machine learning framework for ranking query suggestions. In *In Proc. of SIGIR*, pp. 25–34.

Park, J. H., W. B. Croft, and D. A. Smith (2011). A quasi-synchronous dependence model for information retrieval. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, New York, NY, USA, pp. 17–26. ACM.

Pascanu, R., C. Gulcehre, K. Cho, and Y. Bengio (2013). How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Pascanu, R., T. Mikolov, and Y. Bengio (2013). On the difficulty of training recurrent neural networks. *In Proc. of ICML*.

Pass, G., A. Chowdhury, and C. Torgeson (2006). A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems*, InfoScale '06, New York, NY, USA. ACM.

Pennington, J., R. Socher, and C. Manning (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543. Association for Computational Linguistics.

Piwowarski, B., M.-R. Amini, and M. Lalmas (2012). On using a quantum physics formalism for multidocument summarization. *Journal of the American Society for Information Science and Technology 63*(5), 865–888.

Piwowarski, B., I. Frommholz, M. Lalmas, and K. van Rijsbergen (2010). What can quantum theory bring to information retrieval. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, New York, NY, USA, pp. 59–68. ACM.

Ponte, J. M. and W. B. Croft (1998). A language modeling approach to information retrieval. In *Proc. of SIGIR*, pp. 275–281.

Pothos, E. M. and J. R. Busemeyer (2009, June). A quantum probability explanation for violations of 'rational' decision theory. *Proceedings of the Royal Society B: Biological Sciences 276*(1665), 2171–2178.

Raman, K., P. Bennett, and K. Collins-Thompson (2014, October). Understanding intrinsic diversity in web search: Improving whole-session relevance. *ACM Trans. Inf. Syst. 32*(4), 20:1–20:45.

Řeháček, J., Z. Hradil, E. Knill, and A. Lvovsky (2007). Diluted maximum-likelihood algorithm for quantum tomography. *Physical Review A 75*(4), 042108.

Rijsbergen, C. J. V. (1979). *Information Retrieval* (2nd ed.). Newton, MA, USA: Butterworth-Heinemann.

Robertson, S. (2003). The unified model revisited. In *SIGIR 2003 Workshop on Mathematical/Formal Models in Information Retrieval*.

Robertson, S. (2010). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval 3*(4), 333–389.

Robertson, S. and K. Jones (1976). Relevance weighting of search terms. *Journal of the American Society for Information science 27*(3), 129–146.

Robertson, S., M. Maron, and W. Cooper (1983). The unified probabilistic model for ir. *Research and development in Information Retrieval*, 108–117.

Robertson, S. E., C. J. van Rijsbergen, and M. F. Porter (1980). Probabilistic models of indexing and searching. In *Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pp. 35–56. Butterworth & Co.

Rocchio, J. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System*, pp. 313–323.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature* (323), 533–536.

Sadikov, E., J. Madhavan, L. Wang, and A. Halevy (2010). Clustering query refinements by user intent. In *In Proc. of WWW*, pp. 841–850.

Salakhutdinov, R. and G. Hinton (2009). Semantic hashing. *International Journal of Approximate Reasoning 50*(7), 969–978.

Salton, G. (1986, July). Another look at automatic text-retrieval systems. *Commun. ACM 29*(7), 648–656.

Salton, G. and C. Buckley (1988, August). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage. 24*(5), 513–523.

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

Salton, G., C. S. Yang, and C. T. Yu (1974, July). A theory of term importance in automatic text analysis. Technical report, Cornell University.

Santos, R. L., C. Macdonald, and I. Ounis (2013). Learning to rank query suggestions for adhoc and diversity search. *Information Retrieval 16*(4), 429–451.

Serban, I. V., A. Sordoni, Y. Bengio, A. Courville, and J. Pineau (2016). Hierarchical neural network generative models for movie dialogues. *The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*.

Severyn, A. and A. Moschitti (2015). Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 373–382. ACM.

Shen, Y., X. He, J. Gao, L. Deng, and G. Mesnil (2014). A latent semantic model with convolutional-pooling structure for information retrieval. In *In Proc. of CIKM*, pp. 101–110.

Shi, L. and J.-Y. Nie (2009). Integrating phrase inseparability in phrase-based model. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, New York, NY, USA, pp. 708–709. ACM.

Shi, L. and J.-Y. Nie (2010). Using various term dependencies according to their utilities. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, New York, NY, USA, pp. 1493–1496. ACM.

Shokouhi, M. (2013). Learning to personalize query auto-completion. In *In Proc. of SIGIR*, pp. 103–112.

Shrivastava, A. and P. Li (2014). Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *In Proc. of NIPS*, pp. 2321–2329.

Smucker, M. D., J. Allan, and B. Carterette (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 623–632. ACM.

Song, F. and W. B. Croft (1999). A general language model for information retrieval. In *In Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 279–280.

Song, R., M. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu (2008). Viewing term proximity from a different perspective. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. White (Eds.), *Advances in Information Retrieval*, Volume 4956 of *Lecture Notes in Computer Science*, pp. 346–357. Springer Berlin / Heidelberg.

Song, Y.-I., J.-T. Lee, and H.-C. Rim (2009). Word or phrase?: learning which unit to stress for information retrieval. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, Stroudsburg, PA, USA, pp. 1048–1056. Association for Computational Linguistics.

Sordoni, A., Y. Bengio, and J.-Y. Nie (2014). Learning concept embeddings for query expansion by quantum entropy minimization. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1586–1592.

Sordoni, A., J. He, and J.-Y. Nie (2013). Modeling latent topic interactions using quantum interference for information retrieval. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pp. 1197–1200. ACM.

Sordoni, A. and J.-Y. Nie (2013). Looking at vector space and language models for ir using density matrices. In *Quantum Interaction*, pp. 147–159. Springer.

Sordoni, A., J.-Y. Nie, and Y. Bengio (2013). Modeling term dependencies with quantum language models for IR. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, New York, NY, USA, pp. 653–662. ACM.

Sordoni, A., W. Yuan, and J.-Y. Nie (2013). Experiments with quantum language models in the web track. In *Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013*.

Srikanth, M. and R. Srihari (2002). Biterm language models for document retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, New York, NY, USA, pp. 425–426. ACM.

Srikanth, M. and R. Srihari (2003). Exploiting syntactic structure of queries in a language modeling approach to IR. In *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 476–483.

Strohman, T., D. Metzler, H. Turtle, and W. B. Croft (2005). Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, Volume 2. Citeseer.

Sutskever, I., O. Vinyals, and Q. V. V. Le (2014). Sequence to sequence learning with neural networks. In *In Proc. of NIPS*, pp. 3104–3112.

Svore, K. M., P. H. Kanani, and N. Khan (2010). How good is a span of terms?: exploiting proximity to improve web retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, New York, NY, USA, pp. 154–161. ACM.

Symonds, M., P. D. Bruza, L. Sitbon, and I. Turner (2012). Tensor query expansion: a cognitively motivated relevance model. In *Proceeding of the Sixteenth Australasian Document Computing Symposium*.

Szpektor, I., A. Gionis, and Y. Maarek (2011). Improving recommendation for long-tail queries via templates. In *In Proc. of WWW*, pp. 47–56.

Tao, T. and C. Zhai (2007). An exploration of proximity measures in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, New York, NY, USA, pp. 295–302. ACM.

Tipping, M. E. and C. M. Bishop (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 61*(3), 611–622.

Trueblood, J. S. and J. R. Busemeyer (2011, November). A quantum probability account of order effects in inference. *Cognitive Science 35*(8), 1518–1552.

Tsuda, K., G. Ratsch, and M. K. Warmuth (2006). Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research 6*(1), 995.

Turney, P. D., P. Pantel, et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research 37*(1), 141–188.

Turtle, H. and W. B. Croft (1989). Inference networks for document retrieval. In *Proc. of SIGIR*, pp. 1–24.

Turtle, H. and W. B. Croft (1991, July). Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst. 9*(3), 187–222.

Umegaki, H. (1962). Conditional expectation in an operator algebra, iv (entropy and information). In *Kodai Mathematical Seminar Reports*, Volume 14, pp. 59–85.

Vahabi, H., M. Ackerman, D. Loker, R. Baeza-Yates, and A. Lopez-Ortiz (2013). Orthogonal query recommendation. In *In Proc. of RECSYS*, RecSys '13, pp. 33–40. ACM.

van Rijsbergen, C. (2004). *The Geometry of Information Retrieval*. Cambridge University Press.

van Rijsbergen, C. J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of documentation 33*(2), 106–119.

Wang, Q., J. Xu, H. Li, and N. Craswell (2013, January). Regularized latent semantic indexing: A new approach to large-scale topic modeling. *ACM Trans. Inf. Syst. 31*(1), 5:1–5:44.

Warmuth, M. K. and D. Kuzmin (2009, July). Bayesian generalized probability calculus for density matrices. *Machine Learning 78*(1-2), 63–101.

Wei, X. and W. B. Croft (2006). LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, New York, NY, USA, pp. 178–185. ACM.

Wen, J., J. Nie, and H. Zhang (2001). Clustering user queries of a search engine. In *In Proc. of WWW*, pp. 162–168. ACM.

Weston, J., S. Bengio, and N. Usunier (2011). Wsabie: Scaling up to large vocabulary image annotation. In T. Walsh (Ed.), *IJCAI*, pp. 2764–2770. IJCAI/AAAI.

Widdows, D. and S. Peters (2003). Word vectors and quantum logic: Experiments with negation and disjunction. *Mathematics of language 8*, 141–154.

Wong, S. K. M. and Y. Y. Yao (1995). On modeling information retrieval with probabilistic inference. *ACM Trans. Inf. Syst. 13*(1), 38–68.

Wong, S. K. M., W. Ziarko, and P. C. N. Wong (1985). Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '85, New York, NY, USA, pp. 18–25. ACM.

Wu, Q., C. J. Burges, K. M. Svore, and J. Gao (2010). Adapting boosting for information retrieval measures. *Inf. Retr. 13*(3), 254–270.

Xie, M., Y. Hou, P. Zhang, J. Li, W. Li, and D. Song (2015). Modeling quantum entanglements in quantum language models. *Proc. of IJCAI*, 1362–1368.

Xu, J. and W. B. Croft (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS) 18*, 79–112. ACM ID: 333138.

Yao, K., G. Zweig, and B. Peng (2015). Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565*.

Zhai, C. (2007). Statistical language models for information retrieval a critical review. *Foundations and Trends® in Information Retrieval 2*(3), 137–213.

Zhai, C. (2008, January). Statistical language models for information retrieval. *Synthesis Lectures on Human Language Technologies 1*(1), 1–141.

Zhao, J. and Y. Yun (2009). A proximity language model for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, New York, NY, USA, pp. 291–298. ACM.

Zhao, X., P. Zhang, D. Song, and Y. Hou (2011). A novel re-ranking approach inspired by quantum measurement. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, Berlin, Heidelberg, pp. 721–724. Springer-Verlag.

Zheng, G. and J. Callan (2015). Learning to reweight terms with distributed representations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 575–584. ACM.

Zobel, J. and A. Moffat (1998, April). Exploring the similarity space. *SIGIR Forum 32*(1), 18–34.

Zuccon, G., L. Azzopardi, C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger, and K. van Rijsbergen (2010). Advances in information retrieval. Volume 5993 of *Lecture Notes in Computer Science*, pp. 357–369. Springer Berlin / Heidelberg.

Zuccon, G., B. Piwowarski, and L. Azzopardi (2011, January). On the use of complex numbers in quantum models for information retrieval. In G. Amati and F. Crestani (Eds.), *Advances in Information Retrieval Theory*, Number 6931 in Lecture Notes in Computer Science, pp. 346–350. Springer Berlin Heidelberg.