Exploring a Multidimensional Representation of Documents and Queries[*]

Benjamin Piwowarski
benjamin@bpiwowar.net

Ingo Frommholz
ingo@dcs.gla.ac.uk

Mounia Lalmas
University of Glasgow, UK
mounia@acm.org

Keith van Rijsbergen
University of Glasgow, UK
keith@dcs.gla.ac.uk

Abstract

In Information Retrieval (IR), whether implicitly or explicitly, queries and documents are often represented as vectors. However, it may be more beneficial to consider documents and/or queries as multidimensional objects. Our belief is this would allow building "truly" interactive IR systems, i.e., where interaction is fully incorporated in the IR framework.

The probabilistic formalism of quantum physics represents events and densities as multidimensional objects. This paper presents our first step towards building an interactive IR framework upon this formalism, by stating how the first interaction of the retrieval process, when the user types a query, can be formalised. Our framework depends on a number of parameters affecting the final document ranking. In this paper we experimentally investigate the effect of these parameters, showing that the proposed representation of documents and queries as multidimensional objects can compete with standard approaches, with the additional prospect to be applied to interactive retrieval.

1 Introduction

Most information retrieval (IR) models, including probabilistic and vector ones, use the same underlying one-dimensional representation of documents and queries, i.e., as vectors defined in a vector space, typically a term space. However, this representation has some limits when dealing with more complex IR aspects like interaction, diversity and novelty[1]. Indeed, recent research showed that these complex aspects of the retrieval process benefit from more sophisticated representations of documents and queries [15, 3], in particular those providing for more powerful geometric manipulations of IR components.

The representation of documents and queries in IR should evolve so the user interaction can be incorporated in a natural and principled way in the IR process [13]. Our claim is that representing documents and queries as *multidimensional* objects (e.g. subspaces in a vector space) allows for not only a novel

[*] This an extended version of a paper published in RIAO 2010 [8].

[1] In our research, we are particularly interested in these aspects of the IR process.

but also a more powerful way to tackle this challenge. This representation is particularly interesting from a theoretical point of view because it is possible to use a principled interpretation of the probabilities associated with such multidimensional objects, which comes from quantum physics [13] – the so-called "quantum probabilities" framework. This representation is also interesting from an intuitive point of view because it relies on a geometric representation of documents and queries in a vector space, which has proved successful in IR [2]. This representation reveals also a strong connection between orthogonality (in the vector space) and non-relevance, which has been successfully used to represent term negation in queries [16].

In [10], a framework for interactive IR that relies on such a multidimensional representation of documents and queries was proposed. In this framework, the user's information need (IN) is represented by a set of weighted vectors that evolve with the user's interaction. A probability of relevance of a document (for that IN) is computed with respect to this set. Although the components of our framework were described, they remained abstract. In particular, no explicit document and query representations were proposed. The next step is to operationalise the framework, which is the focus of this paper. We show how document and query representations are computed to then allow estimating the probability of relevance of the document to a given IN.

With respect to related work, multidimensional representations, respectively, of queries were used in [15] to model negative user feedback, and of documents were investigated in [3] in an ad hoc setting. Our work encompasses those since it provides a principled and probabilistic way to work with multidimensional objects. Finally, two lines of research explored, respectively, a subspace representation of documents [7] and of a user's IN [7]. In our work, we go further and show that both documents and INs can be represented as multidimensional objects, and propose a principled methodology to construct these representations.

The outline of this paper is as follows. We first briefly introduce our framework and describe how the probability of relevance is computed within the quantum probability framework (Section 2). Next we show how we construct the query and document representations, and introduce several parameters for these representations (Sections 3 and 4). Finally, we present experimental results, which validate our document and query representations, some of the investigated parameters, and give insights on how our framework can be further developed (Section 5).

2 A Quantum-inspired View for IR

Our IR framework is built upon [10], which is based on quantum probabilities and where we assume that there exists a vector space of *pure*[2] *information needs* (INs), where each vector corresponds to an IN that completely characterises a possible user's IN – by analogy with quantum physics where a vector completly characterises a physical system. Knowing a user's pure IN would determine which documents the IR system should return to that user. From a geometric perspective, a pure IN is answered by a document with a probability

[2] The concept of "pure" IN is new and central to our framework. In this paper, we use "pure IN" to distinguish it from "IN", where the latter refers to information need in its usual sense in IR, e.g., see [6].

that depends on the length of the projection of the pure IN vector onto the document subspace. Because of the uncertainty attached to the IR search process, we suppose that the information being searched by a user can be represented by a set of such pure INs, one for each possible *pure* IN that composes a user's IN.

To compute a probability of relevance of a document to a user's IN, we make use of the generalisation of probabilities developed in quantum physics, which is strongly connected to the geometry of the space used to represent events and densities. A probabilistic event is represented as a subspace (denoted S) in a Hilbert space[3]. Let us assume that S is the event "the document is relevant". A probability can first be defined for a pure IN, represented as a *unit* vector φ, by computing the length of the projection of the vector φ onto the subspace S, that is by computing the value $\left\|\widehat{S}\varphi\right\|^2$ where $\widehat{S}$ is the projector onto the subspace S. This value is the probability that the document is relevant with respect to the pure IN[4].

When a user starts interacting with an IR system by, for instance, typing a query[5], we first compute (see Section 3) an initial set of weighted pure IN vectors, where each weight is the probability that the pure IN corresponds to the actual user's IN. This captures the uncertainty typical to IR where firstly, the representation is only an approximation of the user's IN, and, secondly, the query may be ambiguous. The goal of an IR system is to reduce this undeterminism through interaction.

More formally, we assume that each pure IN vector φ_i is associated with a probability p_i (the weight). We define the probability of the event S by using the usual total probability theorem (across all possible pure INs)[6]:

$$\Pr(S) = \sum_i p_i \Pr(S|\varphi_i) = \sum_i p_i \varphi_i^\top \widehat{S} \varphi_i = \operatorname{tr}\left(\rho\widehat{S}\right) \tag{1}$$

where tr is the trace operator [13, p. 83] and $\rho = \sum_i p_i \varphi_i \varphi_i^\top$ is called a *density operator*[7] and corresponds to a (probabilistic) *mixture* of the pure INs φ_i. In general, any operator ρ characterised by the fact that it is both positive-semi-definite[8] and of trace 1 defines a probability distribution over the subspaces, i.e. it is possible to interpret $\Pr(S) = \operatorname{tr}\left(\rho\widehat{S}\right)$ as a probability [13].

For each document d, we compute a projector $\widehat{S}_d$ (Section 3) and, for a query q, the IN density ρ is approximated by ρ_q (Section 4). Using the projector $\widehat{S}_d$ and the density ρ_q, the probability that a document d is relevant to the query q is then given by $\operatorname{tr}\left(\rho_q\widehat{S}_d\right)$.

In our work, we assume that the vector space of pure INs is the term space, where each dimension corresponds to a term. A pure IN is hence described by a series of weighted terms. A (simplified) example is shown in Figure 1, where

[3] Hilbert spaces (roughly, vector spaces with complex scalars) are a central mathematical concept in quantum physics.

[4] We have $\left\|\widehat{S}\varphi\right\|^2 \in [0,1]$ since $\|\varphi\| = 1$.

[5] Queries are what (usually) users provide to an IR system, as means to express their INs [6].

[6] As in quantum physics, we assume different φ_i correspond to different systems and are thus mutually exclusive.

[7] We will omit the term "operator" in the remaining of the paper.

[8] This means $v^\top \rho v \geq 0$ for any vector v.

the pure IN "pop music" (one unit vector) is represented by the terms "music", "chart" and "hit" of the term space. We show now how document and query representations are computed in this term space.

3 Creating the Document Subspace

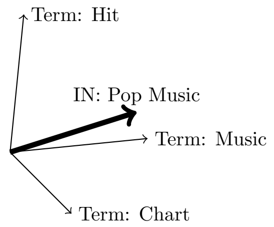


Figure 1: A pure IN in a term space

It is reasonable to assume that a typical document answers various (pure) INs, since it is likely to contain answers (be relevant) to several queries. Moreover, [11] have shown in the context of XML retrieval, that answers to topics (statements of INs) usually correspond to document fragments and not full documents. Building on this, we assume that for each document there is a mapping between its (possibly overlapping and non-contiguous) fragments and a set of pure INs.

A document is thus associated with a set $\mathcal{U}_d$ of vectors in the IN space. We hypothesise that a document is *fully* relevant to a pure IN if the latter can be written as a linear combination of the vectors of $\mathcal{U}_d$, that is, if it is contained in the subspace S_d defined as the span of the vectors in $\mathcal{U}_d$. The document will be *partially* relevant to a pure IN with a probability that depends on the length of the projection of the pure IN vector onto the subspace S_d. The subspace S_d can be interpreted as a geometric representation of the event "the document is relevant". This construction process was validated in a document filtering task [9]. In this paper, we investigate the effect of several parameters (written in bold below) on this process.

Document Fragments. We now assume that document fragments are disjoint, and are obtained through a "natural" segmentation of the document. Various choices are possible, and our first strategy is to use a single fragment, the document itself. This corresponds to the vector space approach where a single vector represents a document. The second strategy is to use paragraphs as fragments as they seem to be of an appropriate size to correspond to a pure IN. We also selected a third type of fragment, the sentence, as it is one of the smallest coherent units in a document.

Weighting Schemes. We now need a vector representation for each fragment. Three weighting schemes are used, namely, tf-idf, tf and binary (term presence/absence). The latter two are chosen since they allow substantial reduction in computational complexity. In addition, binary vectors are close or equal to tf vectors for small fragments, for example, sentences.

$\mathcal{U}_d$ is formally defined as the set of vectors associated with a document d, obtained through one of the above segmentation and weighting scheme, i.e., we have one vector for each fragment. As discussed before, we need to compute the subspace S_d spanned by the vectors of $\mathcal{U}_d$. For this, we use an eigenvalue decomposition where $\sum_{\varphi\in\mathcal{U}_d}\varphi\varphi^\top$ is expressed as $\sum_{i=1}^{D}\lambda_i v_i v_i^\top$ where D is the number of eigenvectors with non null eigenvalues (D is also the dimension of the associated subspace), $\lambda_i > 0$ are the eigenvalues (we suppose without loss of generality that they are of decreasing magnitude, i.e. $\lambda_i \geq \lambda_{i+1}$) and the vectors v_i form an orthonormal basis of the subspace S_d [12].

Dimension selection. As the vectors constructed from the terms occurring in the document fragments are only an approximation of the underlying pure IN vectors, the vectors from $\mathcal{U}_d$ will contain terms that should not be associated with the document. We are thus interested in the eigenvectors associated with the K highest eigenvalues since low eigenvalues are likely to be associated with noise [5]. We are interested in measuring the effect of different dimensions to represent a document. Hence, we chose a simple strategy, where we keep the eigenvectors whose eigenvalue is higher than the average of the eigenvalues, which we compared to two extreme strategies, namely, the case where we select the eigenvector with the highest eigenvalue (one dimension, $K = 1$) and the case where we keep all the eigenvectors (full dimension, $K = D$).

Finally, the projector $\widehat{S}_d$ associated with the K dimensional subspace of document d is expressed as $\sum_{i=1}^{K} v_i v_i^\top$.

4 Creating the Query Density

We now focus on the primary contribution of the paper, namely, the construction of the IN density ρ_q for a given query q.

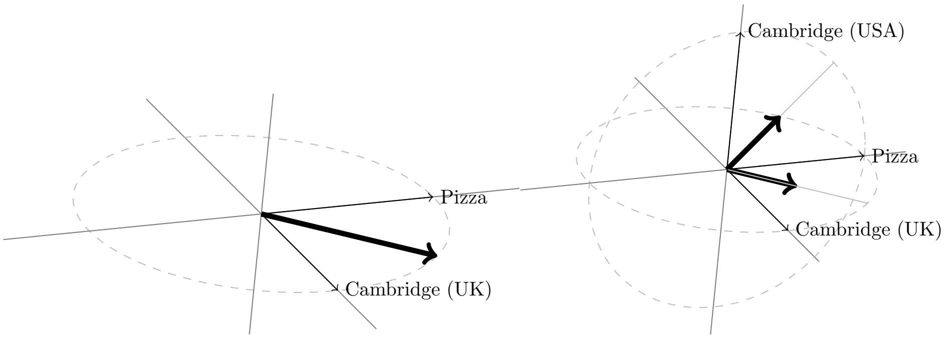
As a query in its simplest form consists of a set of terms, we are first interested in building the query representation for a query composed of a single term, t. We described how a document is represented as a set of pure IN vectors corresponding to different fragments of the document. We extend this idea, and suppose that a query term t can be represented as the set $\mathcal{U}_t$ of pure IN vectors that correspond to document fragments containing the term t. That is, we use the immediate surroundings of the term occurrences in the documents of the collection being searched to build that term representation. This is similar to pseudo-relevance feedback using passages from retrieved documents containing the query terms [1]. The difference is that we use all the passages to build the query representation as we want to consider all possible pure INs associated with the term t.

As we have *a priori* no way to distinguish between the different vectors in $\mathcal{U}_t$, we assume that each vector is equally likely to be a pure IN composing the user's actual IN. Hence, a document is relevant to the user's IN if it is relevant to any of the vectors of $\mathcal{U}_t$, where the vectors are drawn with a uniform probability. The corresponding density is then written as:

$$\rho_t = \frac{1}{N_t} \sum_{\varphi \in \mathcal{U}_t} \varphi \varphi^\top \tag{2}$$

where N_t is the number of vectors associated with term t (the cardinality of $\mathcal{U}_t$). This definition of ρ_t has all the required properties of a density (see Section 2). In practice, this representation of a single-term query t means that, the more vectors φ from $\mathcal{U}_t$ lie in the document subspace, the higher the relevance of the document to the query. This query representation hence favours documents containing different "aspects" of the IN, each of them as represented by one of the pure INs in $\mathcal{U}_t$ associated with a query term t.

We discuss next the representation of a query composed of several terms. There are three main parameters (written in bold below).



(a) A superposition of two INs

(b) A mixture of two INs

Figure 2: Combining INs

Weighting scheme. As for documents, three weighting schemes, namely, tf-idf, tf and binary, are used to build the vectors forming $\mathcal{U}_t$.

Query construction (mixture). The above query representation (Equation 2) can be generalised to a query composed of several terms. We assume that a relevant document should equally answer all pure INs associated with each query term. To compute the probability of relevance of a document d, we first select a term from the query (with a probability w_t, see the next paragraph), and then one of the vectors in $\mathcal{U}_t$. With this vector, we compute the probability of document d to be relevant to this pure IN. We repeat the process and average over all the possible combinations. This defines the probability of relevance of document d given the query. Formally, this corresponds to a density defined as a *mixture* of all the pure IN vectors associated with the query terms. This density is built from the individual query term densities ρ_t (Equation 2):

$$\rho_q^{(m)} = \sum_{t \in q} \sum_{\varphi \in \mathcal{U}_t} \frac{w_t}{N_t} \varphi \varphi^\top = \sum_{t \in q} w_t \rho_t \tag{3}$$

Query term weight. The weights w_t are used to quantify the importance of each term t of the query. We experimented with two settings, one where all the w_t were equal, and the other where they were set to the corresponding term idf values. In both approaches, we normalise the weights so their sum equals 1.

We present a second query construction process, inspired from IR and quantum theory. In vectorial IR, a query is represented by a vector that corresponds to a linear combination of the vectors associated with the query terms. In quantum theory, a normalised linear combination corresponds to the principle of superposition, where the description of a system state can be *superposed* to describe a new system state.

In our case, the system state corresponds to the user's pure IN, and we use the superposition principle to build new pure INs from existing ones, as illustrated with the example shown in Figure 2. Let φ_p, $\varphi_{c/uk}$ and $\varphi_{c/usa}$ be three vectors in a three-dimensional IN space that, respectively, represent the INs "I want a pizza", "I want it to be delivered in Cambridge (UK)" and "I want it to be delivered in Cambridge (USA)". The pure IN vector "Pizza delivered in

Cambridge (UK)" would be represented by a (normalised) linear combination (or superposition) of φ_p and $\varphi_{c/uk}$, as depicted in Figure 2(a). We can similarly build the IN for Cambridge (USA). To represent the ambiguous query "pizza delivery in Cambridge" where we do not know whether Cambridge is in the USA or the UK, and assuming there is no other source of ambiguity, we would use a mixture of the two possible superposed INs, as depicted by the two vectors of the mixture in Figure 2(b), which brings us to another variant of query construction, the mixture of superpositions.

Query construction (mixture of superpositions). To compute the probability of relevance, for each term t of the query, we randomly select a vector from the set $\mathcal{U}_t$. We then superpose (i.e., compute a linear combination) the selected vectors (one for each term), where the weight in the linear combination is $\sqrt{w_t}$ (see below for why we use a square root). From this vector, we compute the probability of the document to be relevant to this IN made from the superposition of IN vectors (one per query term). With respect to our example, the set $\mathcal{U}_{pizza}$ would be just one vector ("*I want a pizza to be delivered*"), and $\mathcal{U}_{Cambridge}$ would contain two vectors (one for UK, one for USA).

As with the simple mixture approach, the above process can be repeated for all the possible selections of vectors and the corresponding query density is:

$$\rho_q^{(ms)} = \frac{1}{Z_q} \sum_{\varphi_1 \in \mathcal{U}_{t_1}} \cdots \sum_{\varphi_n \in \mathcal{U}_{t_n}} \left(\sum_{i=1}^{n} \sqrt{\frac{w_{t_i}}{N_{t_i}}} \varphi_i \right) \left(\sum_{i=1}^{n} \sqrt{\frac{w_{t_i}}{N_{t_i}}} \varphi_i \right)^\top \tag{4}$$

where Z_q is a normalisation coefficient, and t_i ($i = 1 \ldots n$) are the n query terms. We use N_t to ensure that each term contribution is equally important, and square roots because both N_t and w_t appear two times in the above formula. In theory the vector $\sum_i \sqrt{\frac{w_{t_i}}{N_{t_i}}} \varphi_i$ should be normalised but to obtain a computable formula we did not do so [9].

Note that for one-term queries, the two described query constructions (mixture and mixture of superpositions) give the same result. Another important point from a computational perspective is that in both cases, the query can be estimated from single term densities (not demonstrated for Equation 4). We hence pre-compute the densities ρ_t for each term t, and use them at query time to compute $\rho_q^{(m)}$ and $\rho_q^{(ms)}$.

Dimension selection. As for the representation of documents, both densities are expressed, through eigenvalue decomposition, as a sum $\sum_{i=1}^{D} \lambda_i v_i v_i^\top$ where the (λ_i, v_i) are eigenpairs ordered by decreasing eigenvalues. Our final density used for computing the probability of relevance is then $\rho_q = \sum_{i=1}^{K} \lambda_i v_i v_i^\top$ where K is the selected dimension (where $K \leq D$). We use the same three strategies to set K that were used for the document representations (see end of Section 3).

5 Experiments and Analysis

In previous work [9], we validated the subspace document representation on a filtering task. In this paper, we explore both the document and the query

[9] The effect will be to give higher importance to superpositions of vectors φ_i who are similar, i.e., whose cosine is closer to 1.

Parameters	**Means**
(1) Document fragment	sentence (0.14) >> paragraph (0.12) >> document (0.11)
(2) Weighting scheme (document fragment)	tf (0.13) >> tf-idf (0.12), binary (0.12)
(3) Weighting scheme (query)	tf-idf (0.13) >> tf (0.12), tf-idf > binary (0.12)
(4) Dimension selection (document)	all (0.14) >> highest (0.11), mean (0.14) >> highest (0.11)
(5) Dimension selection (query)	all (0.13), mean (0.13), highest (0.12)
(6) Term weight in query	idf (0.13) >> uniform (0.12)
(7) Query construction	mixture (0.13), mixture of superpositions (0.13)

Table 1: Means of medians of average precision for each topic. The ">" (resp. ">>") sign is used to denote statistical significance at 0.05 (resp. 0.01).

representations in an ad hoc retrieval task. In particular, we look at the effects of the parameters discussed in Sections 3 and 4. These are listed on the left column of Table 1. As the parameters are mostly independent from each other, we experimented with 756 settings; those not making sense were ignored[10].

We used the INEX 2008 collection in our experiments because its documents have markup (in XML format) delineating text units. The collection consists of 659,388 Wikipedia documents in XML format, using tags such as article, section and paragraph to model a document logical structure [4]. INEX 2008 has 70 assessed topics, and for each topic, relevant passages in (pooled) documents were highlighted by human assessors. A document containing a relevant passage is assumed relevant, which is in accordance with Trec guidelines.

We preprocessed the documents by extracting the fragments, i.e., the whole document, the paragraphs (as determined by the XML markup) and the sentences[11]. We then stemmed and stopped (using the SMART list of stop-words) the text fragments. For each term t, we computed an approximation of the term density ρ_t (Equation 2) based on a sample of 10,000 documents (maximum) containing the term t and using a thin eigenvalue decomposition with maximum rank set to 10 [12, pp. 171-181]. This value, chosen through experimentation, represents a good trade-off between complexity and efficiency. For each query q, we computed the query density ρ_q using the densities ρ_t of its composing terms t, using either the simple mixture (Equation 3) or the mixture of superpositions (Equation 4). Then, we first retrieved a set of 1,500 documents using BM25[12] [14]. For each retrieved document d and each parameter setting, we computed the projector $\widehat{S}_d$ and computed a probability of relevance as $\operatorname{tr}\left(\rho_q \widehat{S}_d\right)$. We used this value to re-rank the documents.

Table 1 shows our results. For each parameter (left column), we show in the right column the means of the medians of average precision computed for

[10] When using a whole document as fragment, the document subspace is one-dimensional and in this case there is no point to investigate the dimension selection parameter.

[11] We use `http://www.andy-roberts.net/software/jTokeniser/index.html` for this.

[12] With the standard parameter values.

Mixture of superpositions		**Mixture**	
Δ_{AP}	**Topic**	Δ_{AP}	**Topic**
0.22	social networks mining	0.32	”records management” metadata
0.19	virtual museums	0.16	Tata Motors Company in India
0.10	genetically modified food safety	0.15	Nikola Tesla inventions patents
0.09	wikipedia vandalism	0.08	vodka producing countries
0.06	flower meaning	0.08	mahler symphony song

Table 2: Top five performing topics using, respectively, mixture of superpositions (Equation 4) and mixture (Equation 3) as query representation.

the different settings of that parameter. For example in row (1), when the fragment is “sentence”, this value is 0.14. To compare two settings, say “sentence” vs. “paragraph”, we performed a paired t-test where each pair of samples corresponds to the same topic and same parameter values (weighting scheme, dimension selection, query term weight, query construction) but for the document fragment setting. For this example, the result shows that using sentence fragments outperformed paragraph fragments at a 0.01 significance level. We discuss each result next.

For the document fragment parameter (1), the best performing setting was with “sentence” followed by “paragraph” and “document”. Each time the difference was found to be significant at a 0.01 level. This indicates that the right level of segmentation (to construct the pure IN vectors) is at sentence level.

Overall, the weighting scheme for document fragments and queries had some effect on retrieval effectiveness. For building the query term density (3), the tf-idf scheme led to significantly better results, whereas for document fragments (2), the tf scheme performed better. The results are somehow in contradiction with vectorial IR findings, but might stem from the fact that to build the query term representation we sample much more vectors than for the document one; hence in the former case it is important to weight terms according to their importance (idf). When looking more in details into the results, we also found out that the weighting scheme was highly dependent on the other parameters, and should hence be chosen depending on them.

The setting of the subspace dimension has a different effect on documents and queries. For documents (4), performance was improved using the full dimension or the mean of the eigenvalues (to determine the dimension of the subspace representation). This shows that using more than one dimension to represent a document is beneficial. However, for queries (5) we observe only a slight improvement when using multiple dimensions (none of which were significant).

For the query construction methodology, we first see that weighting the query terms by their idf values outperformed using a uniform scheme (6). When looking at a mixture vs. mixture of superpositions (7), no significant overall performance difference exists. However, we observe different behaviours depending on the topic. Table 2 shows the best performing topics for, respectively, the mixture of superpositions and the mixture. The topics better handled by the mixture of superpositions are topics for which the terms form a “concept”, for example “social networks mining” where the three terms together have a specific meaning. For the mixture, topics for which each term reflects a different aspect of the topic, e.g. “”records management” metadata”, where “metadata” and “records management" are the two different concepts, had a better performance. This indicates that selecting the query density computation according to the topic may prove beneficial.

The above example suggests that it may be beneficial to treat parts of the query differently by combining both construction methods into one query. For example, the terms "records" and "management" form a single aspect and should thus be superposed. Afterwards, the superposed terms should be mixed with "metadata", which describes another aspect, to answer the query ""records management" metadata". In general, to determine which terms form a single concept, we can rely on explicit markers like quotes in this example, or on an automatic algorithm based e.g. on co-occurrences.

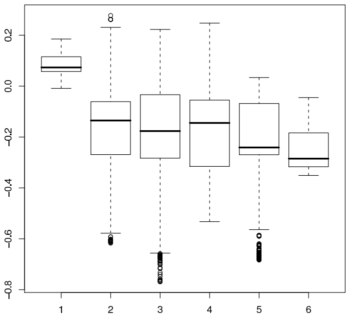


Figure 3: Boxplot of the effect of query length (number of terms) on average precision. The x-axis is the query length (number of terms) and the y-axis is the difference in average precision between BM25 and our method in different settings.

We also compared our results to a state-of-the-art retrieval IR system, namely BM25 [14]. We found that the performances of our framework were consistently lower in average (using standard IR evaluation metrics). A brief analysis (not reported here) comparing the results of the best performing configurations with BM25 for the topics in Table 2 reveals that we could get closer to BM25 performances by (again) choosing the right query construction methodology (mixture vs mixture of superpositions).

Finally, we investigated the effect of query length (number of terms) and the number of relevant documents (of a query) on retrieval effectiveness. No correlation was found between the difference in performance between BM25 and our framework, and the number of relevant documents. There was however a strong dependency on the query length. As illustrated in Figure 3, when the query length is one (there is no difference between the two query density construction methods), our approach outperforms consistently BM25; when the number of terms in the query increases, retrieval performance drops. This further confirms that the appropriate calculation of the query density – in particular for multi-term queries – needs to be investigated.

6 Conclusion and Future Work

In this paper, we presented a methodology to build multidimensional representations of documents and queries. These representations are inspired from the geometric/probabilistic framework of quantum physics. The latter allows us to compute probabilities of relevance based on a more complex representations of documents than a simple bag of words, namely, a multidimensional one based on document fragments. We believe that such a multidimensional representation is key to a successful framework for exploiting user's interaction [13].

We performed experiments to explore various parameters influencing the effectiveness of our representations. We showed that using more than one dimension to represent documents improves performance, confirming previous results. Considering a document as a fragment, as done in most classical models, is not sufficient to distinguish between the different information needs a document covers. Indeed, while most of the classical models only take the mere occurrence of a term into account, we showed in our experiments that the vicinity of terms (the fact that they appear in the same fragment) plays an important role.

We also explored two different and principled ways to construct the query representation. We have shown that queries whose terms define a concept and those whose terms are more independent are better handled by two different methods, respectively, the mixture of superpositions and the (simple) mixture. This suggests that we can gain further improvements if both strategies are applied together in an adaptive manner. This is part of our future work.

As our representation of queries and documents aims at tackling interactive IR, this works validates our framework for the most common first interaction step between a user and an IR system – a user typing a query. Exploiting further interaction steps (for example viewing or saving a document), is also part of our future work.

Acknowledgements This research was supported by an Engineering and Physical Sciences Research Council grant (Grant Number EP/F015984/2). Mounia Lalmas is currently funded by Microsoft Research/Royal Academy of Engineering.

References

[1] J. Allan. Relevance feedback with too much data. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *18th ACM SIGIR conference*, Seattle, Washington, United States, 1995. ACM.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, New York, USA, 1999.

[3] L. Che, J. Zen, and N. Tokud. A "stereo" document representation for textual information retrieval. *JASIST*, 5, 2006.

[4] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.

[5] M. Efron. Eigenvalue-based model selection during latent semantic indexing. *JASIST*, 56(9), 2005.

[6] P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context (The Information Retrieval Series)*. Springer-Verlag, Secaucus, NJ, USA, 2005.

[7] M. Melucci. A basis for information retrieval in context. *ACM TOIS*, 26(3), 2008.

[8] B. Piwowarski, I. Frommholz, M. Lalmas, and K. van Rijsbergen. Exploring a multidimensional representation of documents and queries. In *RIAO proceedings*, 2010.

[9] B. Piwowarski, I. Frommholz, Y. Moshfeghi, M. Lalmas, and K. van Rijsbergen. Filtering documents with subspaces. In *Proceedings of the 32nd ECIR Conference*, 2010. Poster.

[10] B. Piwowarski and M. Lalmas. A Quantum-based Model for Interactive Information Retrieval (extended version). *ArXiv e-prints*, (0906.4026), 2009.

[11] B. Piwowarski, A. Trotman, and M. Lalmas. Sound and complete relevance assessments for XML retrieval. *ACM TOIS*, 27(1), 2009.

[12] G. W. Stewart. *Eigensystems*, volume 2 of *Matrix algorithms*. SIAM, 2001.

[13] C. J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, New York, NY, USA, 2004.

[14] S. Walker and S. E. Robertson. Okapi/keenbow at TREC-8. In E. M. Voorhees and D. K. Harman, editors, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland, USA, 1999.

[15] X. Wang, H. Fang, and C. Zhai. A study of methods for negative relevance feedback. In S.-H. Myaeng, D. W. Oard, F. Sebastiani, T.-S. Chua, and M.-K. Leong, editors, *Proceedings of the 31st Annual International ACM SIGIR*, New York, NY, USA, 2008. ACM.

[16] D. Widdows. Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In *Proceedings of the 41st ACL conference*, Morristown, NJ, USA, 2003. Association for Computational Linguistics.