The Quantum Information Access Framework

(part of the EPSRC "renaissance" project)

B. Piwowarski (University of Glasgow)

18th of February, 2011

University of Edinburgh, Scotland, UK

Outline

- Quick quantum probabilities tutorial
- Topical space: densities and events
- Applications
  - Ad-hoc IR
  - Summarisation
- Computational issues & kernels
- Current and future works

The Quantum IR framework

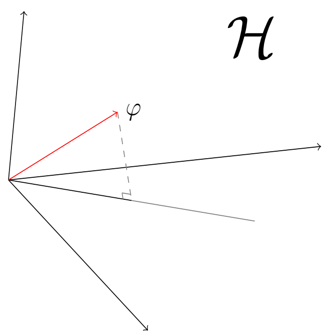
QUANTUM PROBABILITIES IN FOUR SLIDES

Systems and states

Quantum theory describes the behavior of matter at atomic and subatomic scales

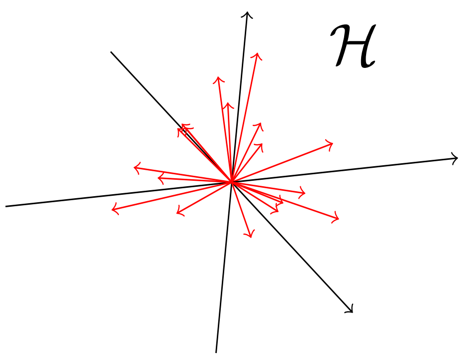
Physical system in a known state = **unit** vector in a state (Hilbert) space

Known state



one state vector

Unknown state

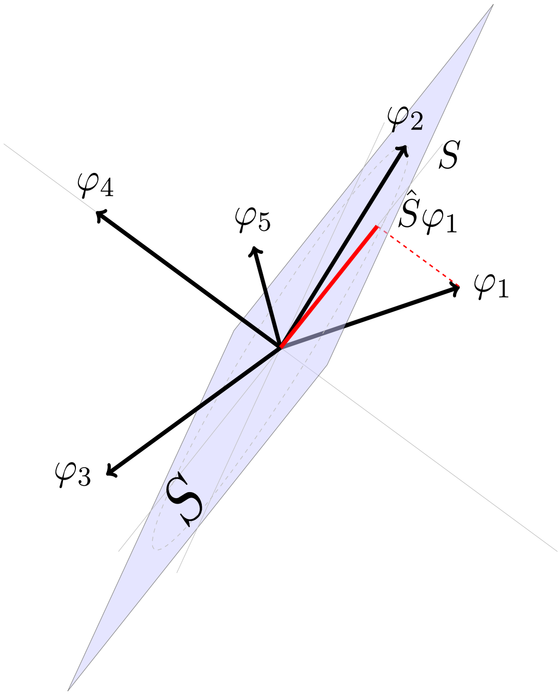


$Pr(\varphi|\mathcal{V})$

Ensemble of state vectors (sum to 1)

Computing probabilities

A state vector (and by extension an ensemble of state vectors) defines a probability distribution over events (subspaces *S*)



Pure state

$$Q(S|\varphi) = ||\hat{S}\varphi||^2$$

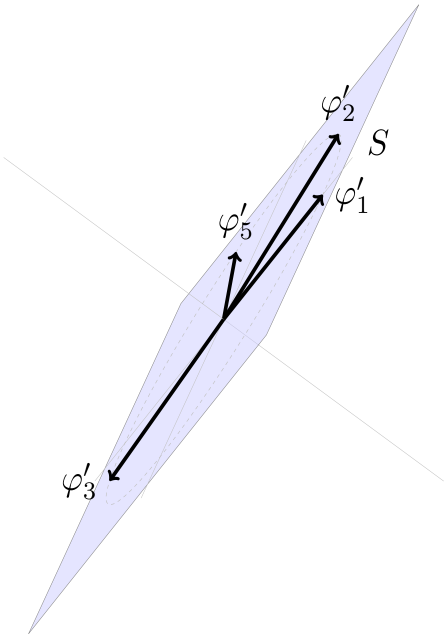
Q for "quantum probability"

Mixture of states $\mathcal{V}$

$$Pr(S|\mathcal{V}) = \sum_{\varphi} Q(S|\varphi) Pr(\varphi|\mathcal{V})$$

Updating distributions

The posterior probability distribution is obtained by projection



$$\mathcal{V} \triangleright S$$

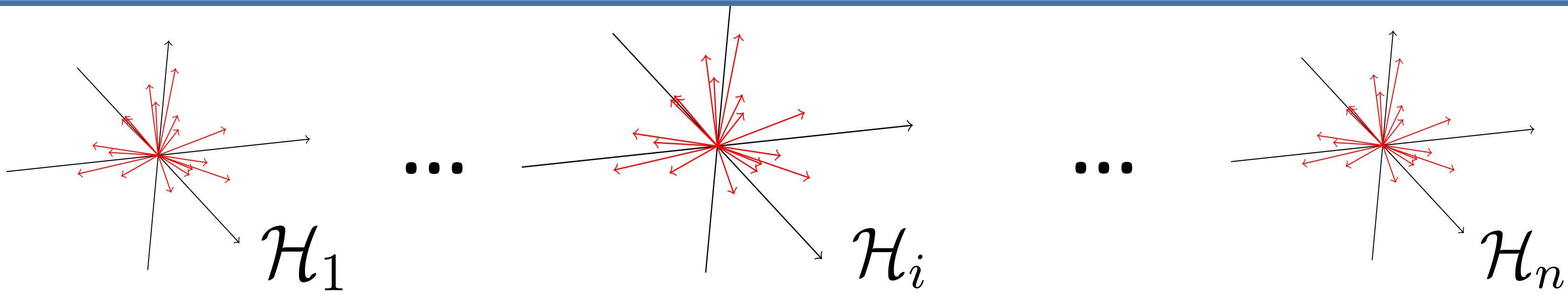
$$Pr(\varphi|\mathcal{V} \triangleright S) = K \sum_{\varphi' / \varphi' = \varphi \triangleright S} Q(S|\varphi') Pr(\varphi'|\mathcal{V})$$

φ' is the normalised projection of φ

Monday, 28 February 2011

Multi-part systems

Tensor products of Hilbert space (generalisation of a product of probability spaces) define a space for multi-part systems



Tensor space $\mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_i \otimes \ldots \otimes \mathcal{H}_n$

Probabilities $$Q\left(\bigotimes_i S_i \middle| \bigotimes_i \varphi_i\right) = \prod_i Q\left(S_i | \varphi_i\right)$$

The Quantum IA framework

DENSITIES AND EVENTS IN THE TOPICAL SPACE

Hypotheses

- There exists a space of *atomic topics, e.g.* "density in a topical Hilbert space", where each aspect correspond to a unit vector
- Topicality can be measured through Quantum Probabilities
  – Is this fragment of text about **X**?

Fragments representation

are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

2.2 Profile Updating and Matching

The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

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

are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

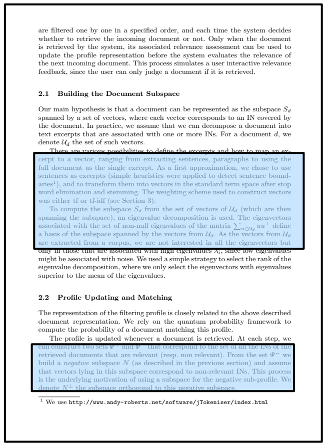
2.2 Profile Updating and Matching

The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

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

Fragments representation



are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

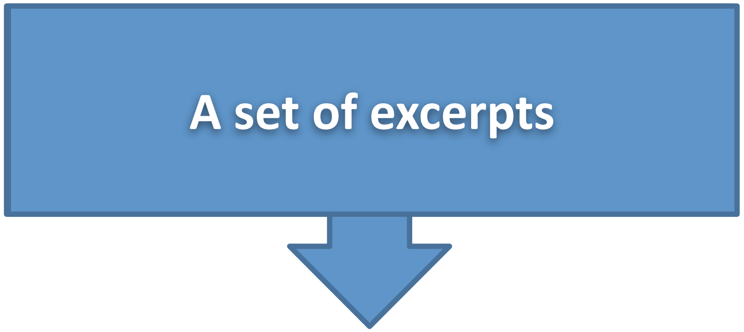
To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

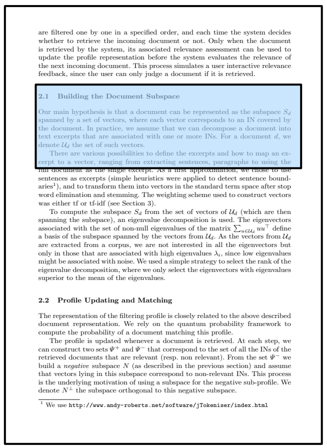
2.2 Profile Updating and Matching

The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

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





are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

2.2 Profile Updating and Matching

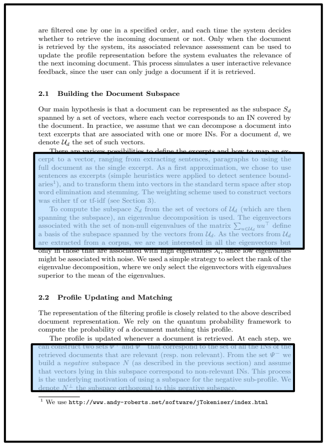
The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

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

Monday, 28 February 2011

Fragments representation



are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

2.2 Profile Updating and Matching

The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

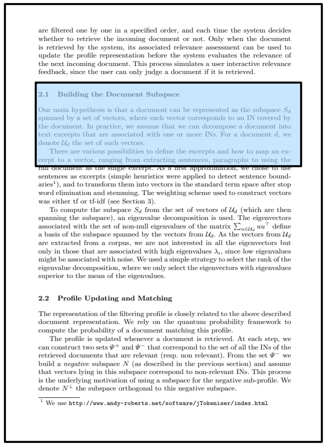
[1] We use `http://www.andy-roberts.net/software/jTokeniser/index.html`

A set of excerpts

↓

An excerpt is an atomic topic

↓



are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

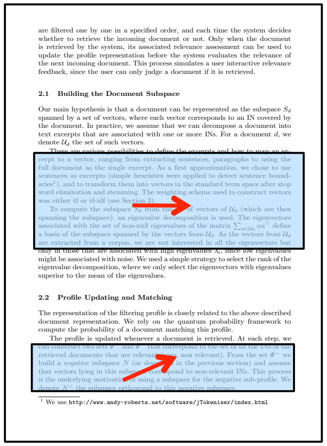
2.2 Profile Updating and Matching

The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

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

Fragments representation



are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

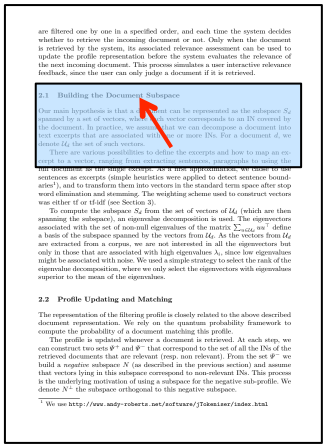
To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

2.2 Profile Updating and Matching

The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

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



are filtered one by one in a specified order, and each time the system decides whether to retrieve the incoming document or not. Only when the document is retrieved by the system, its associated relevance assessment can be used to update the profile representation before the system evaluates the relevance of the next incoming document. This process simulates a user interactive relevance feedback, since the user can only judge a document if it is retrieved.

2.1 Building the Document Subspace

Our main hypothesis is that a document can be represented as the subspace S_d spanned by a set of vectors, where each vector corresponds to an IN covered by the document. In practice, we assume that we can decompose a document into text excerpts that are associated with one or more INs. For a document d, we denote $\mathcal{U}_d$ the set of such vectors.

There are various possibilities to define the excerpts and how to map an excerpt to a vector, ranging from extracting sentences, paragraphs to using the full document as the single excerpt. As a first approximation, we chose to use sentences as excerpts (simple heuristics were applied to detect sentence boundaries[1]), and to transform them into vectors in the standard term space after stop word elimination and stemming. The weighting scheme used to construct vectors was either tf or tf-idf (see Section 3).

To compute the subspace S_d from the set of vectors of $\mathcal{U}_d$ (which are then spanning the subspace), an eigenvalue decomposition is used. The eigenvectors associated with the set of non-null eigenvalues of the matrix $\sum_{u \in \mathcal{U}_d} uu^\top$ define a basis of the subspace spanned by the vectors from $\mathcal{U}_d$. As the vectors from $\mathcal{U}_d$ are extracted from a corpus, we are not interested in all the eigenvectors but only in those that are associated with high eigenvalues λ_i, since low eigenvalues might be associated with noise. We used a simple strategy to select the rank of the eigenvalue decomposition, where we only select the eigenvectors with eigenvalues superior to the mean of the eigenvalues.

2.2 Profile Updating and Matching

The representation of the filtering profile is closely related to the above described document representation. We rely on the quantum probability framework to compute the probability of a document matching this profile.

The profile is updated whenever a document is retrieved. At each step, we can construct two sets Ψ^+ and Ψ^- that correspond to the set of all the INs of the retrieved documents that are relevant (resp. non relevant). From the set Ψ^- we build a *negative* subspace N (as described in the previous section) and assume that vectors lying in this subspace correspond to non-relevant INs. This process is the underlying motivation of using a subspace for the negative sub-profile. We denote $N^\perp$ the subspace orthogonal to this negative subspace.

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

A set of excerpts

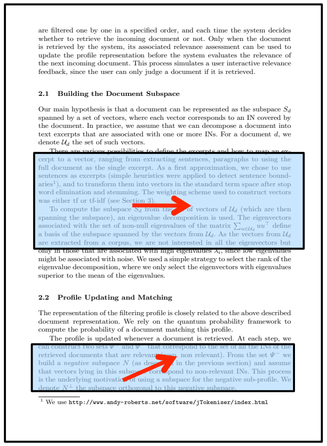
↓

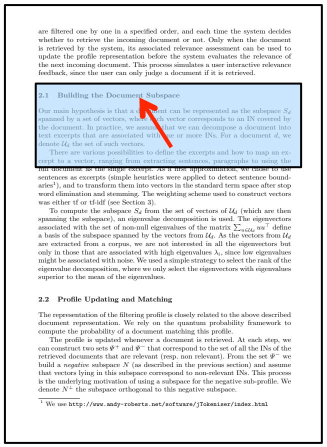
An excerpt is an atomic topic

↓

We can associate to each excerpt a unit vector in the topical space and hence a probability distribution over atomic topics for a fragment

Fragments representation





A set of excerpts

↓

An excerpt is an atomic topic

↓

We can associate to each excerpt a unit vector in the topical space and hence a <u>probability distribution over atomic topics</u> for a fragment

<u>In practice</u>:

1. Each vector is in a term space
2. Non null components correspond to words in the fragment
3. Weighting scheme can be tf, tf-idf, etc.

Two views over extracted topics

DENSITY

We can use any distribution over the extracted vectors (in practice, uniform), i.e.

$$Pr(\varphi|\mathcal{V})$$

equals the probability of one fragment corresponding to φ

Two views over extracted topics

DENSITY

We can use any distribution over the extracted vectors (in practice, uniform), i.e.

$$Pr(\varphi|\mathcal{V})$$

equals the probability of one fragment corresponding to φ

SUBSPACE

We want $Q(S|\varphi) = 1$

for any extracted aspect…

but no more, i.e. $Q(S|\varphi) = 0$

for orthogonal aspects

S = subspace spanned by the extracted aspects

The Quantum IR framework

AD-HOC INFORMATION RETRIEVAL

The Quantum IR (QIR) formalism

Information needs often composed of *several aspects*

TREC-8 topic 408: "What tropical storms (hurricanes and typhoons) have caused significant property damage and loss of life?"

We suppose there exists an **aspect space** (Hilbert space)

aspect = weighted set of aspects vectors

The *Information Need* space

- An IN is defined as a multi-part system in a tensor product of **aspect spaces**
  - (text) Topical (i.e., term space)
  - (image) Colour, shape, etc.
  - (metadata) Reviews, rating, etc.

We suppose there exists an **information need space** (Hilbert space) made of one or more **aspect spaces**

Information need = weighted set of tensor product of aspect vectors

Motivations

- Multi-dimensional and geometric representation of
  - Queries
  - Documents
- Principled way of
  - Capturing interaction
  - Deal with novelty & diversity

Multi-dimensional representation

- Documents can have multiple topics
  - A *new* topic = a new dimension
  - No document length normalisation needed
- INs are naturally diverse (ensemble of states)
  - Interaction
  - Negative information

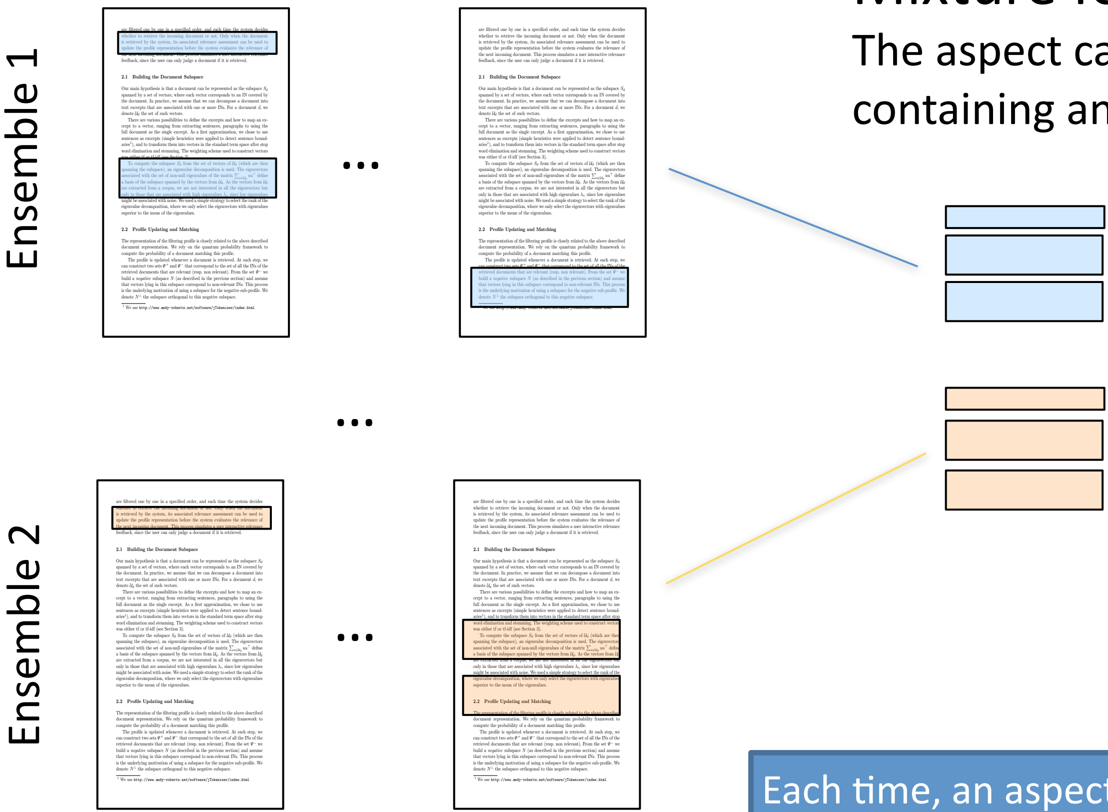
Representation in topical space(s)

- Documents = subspace defined by the extracted fragments
- Terms (next slides):
  - Mixture
  - Mixture of superposition
  - Tensor

Mixture (M)

Mixture for *2* ensembles

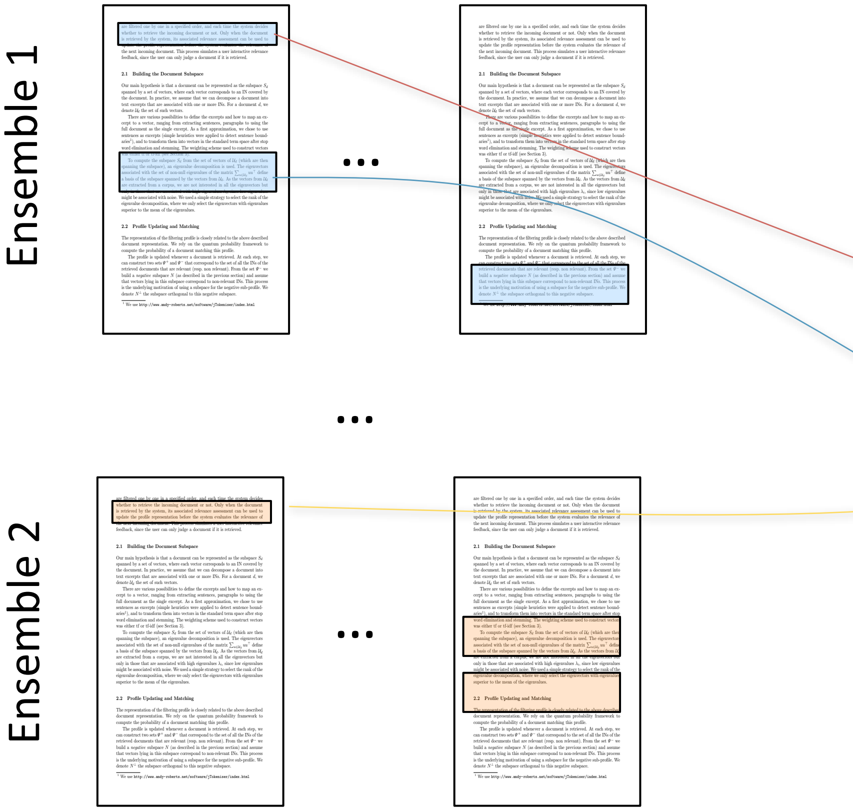
The aspect can be any of the fragment containing any of the terms



Each time, an aspect must be satisfied for a document to be relevant

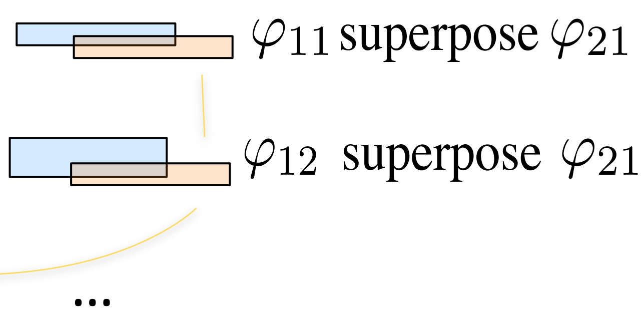
The more aspects are satisfied in the document, the more the document is relevant

Mixture of superpositions (MS)



Mixture of superpositions for *2* ensembles

The aspect can be any of the superposition of fragments from each set



Each time, a superposition of aspects must be satisfied for a document to be relevant

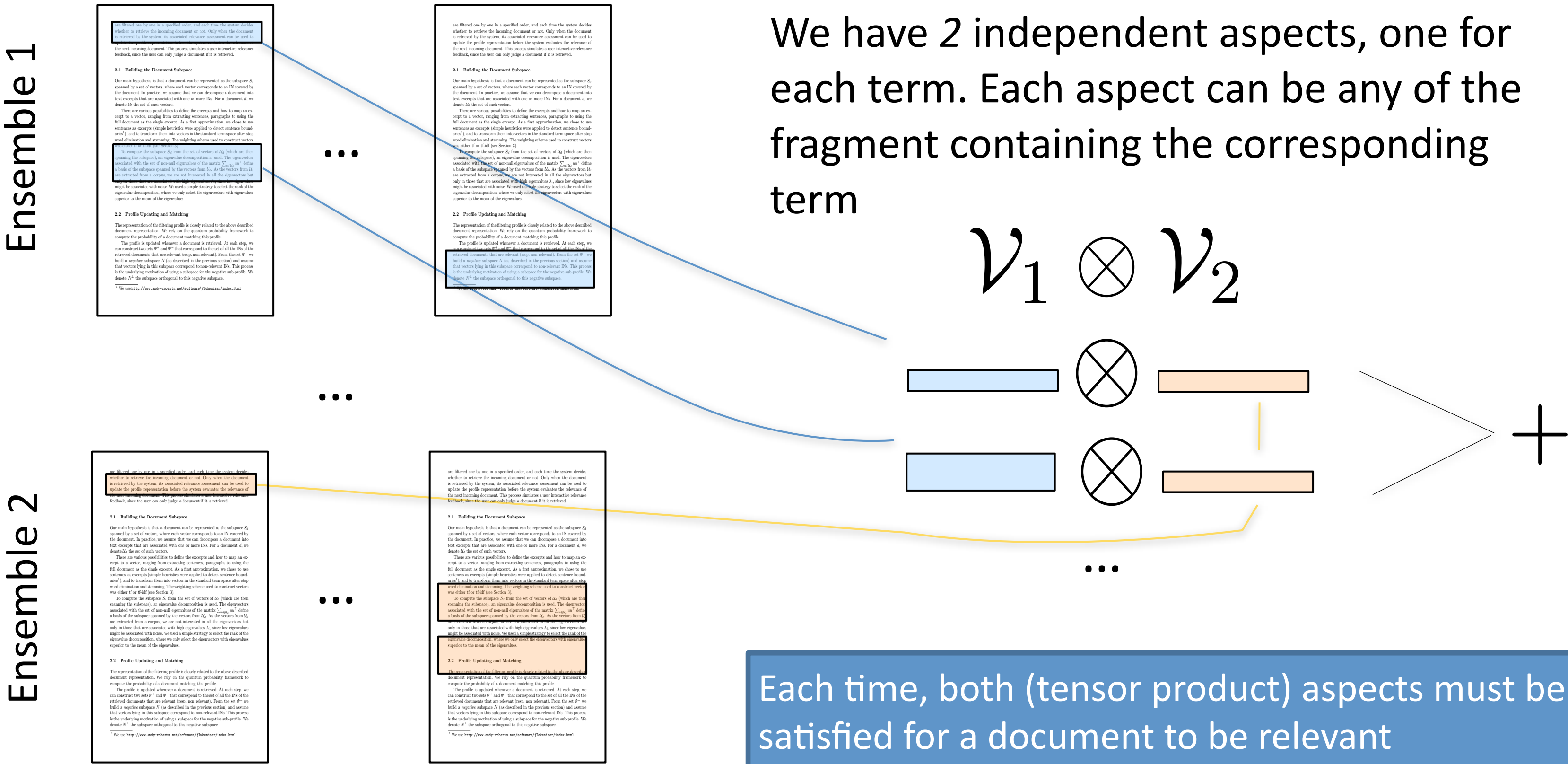
The more superpositions of aspects are satisfied in the document, the more the document is relevant

Tensor product (MS)

Tensor product for *2 sets*

We have *2* independent aspects, one for each term. Each aspect can be any of the fragment containing the corresponding term

$$\mathcal{V}_1 \otimes \mathcal{V}_2$$



Each time, both (tensor product) aspects must be satisfied for a document to be relevant

The more tensor products of aspects are satisfied in the document, the more a document is relevant

Experiments

Collections

- TREC 1, 2, 3, 5, 6, 7, 8
- Basic ad hoc scenario
- **Re-ranking** of the top 1,500 documents found by BM25

Extracting topical aspects

- Sliding window of span 5 for documents
- Window of span 5 for terms
- Binary weighting scheme

Process

- Compute the term representation for each term
- Compute the query representation from term representation
- Evaluation with Mean Average Precision

Results with TREC

	TREC 1	TREC 2	TREC 3	TREC 5	TREC 6	TREC 7	TREC 8
BM25	0.230	0.209	0.282	0.148	0.224	0.182	0.236
M	0.205*	0.184*	0.226*	0.115*	0.173*	0.142*	0.165*
MS	0.209*	0.167*	0.206*	0.112*	0.157*	0.117*	0.159*
T	0.232	0.195*	0.281	0.148	0.214	0.182	0.234

For a query $q = \{t_1, \ldots, t_n\}$

(M) mixture of ensembles

(MS) mixture of superpositions of ensembles

(T) Tensor product of ensembles

$$\mathcal{V}_{t_1}, \ldots, \mathcal{V}_{t_n}$$

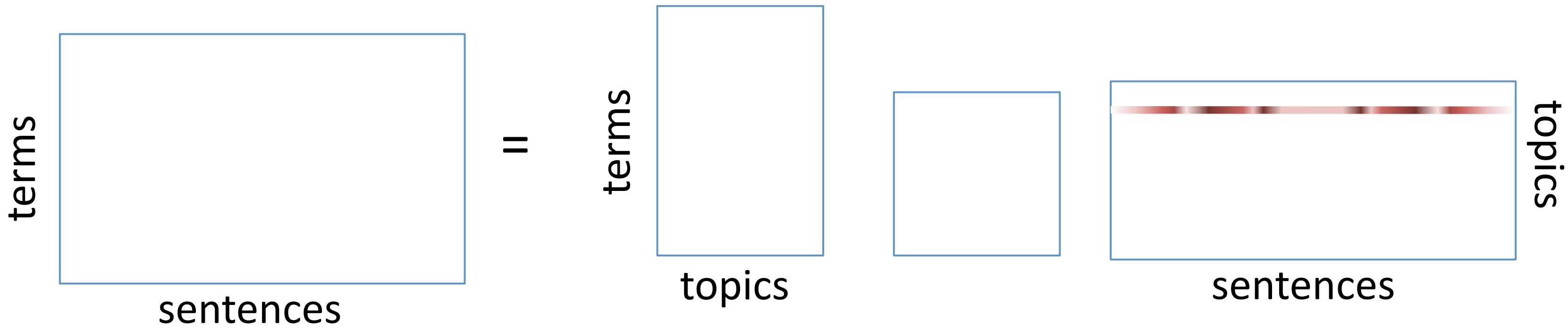
* significance at a 0.05 level (paired t-test)

The Quantum IA framework

SUMMARISATION

Motivations

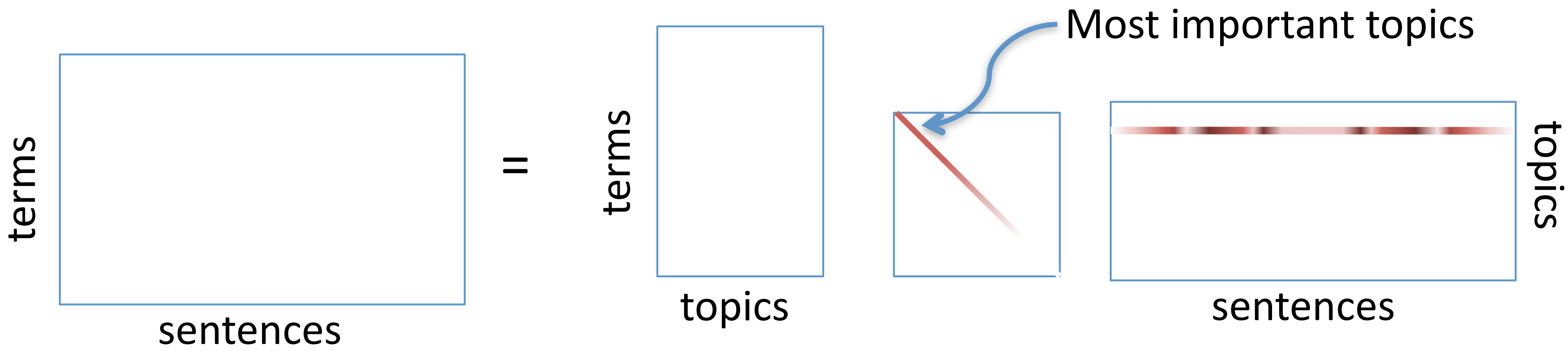
- LSA-based summarisation...



- ... bears similarities with quantum probabilities

Motivations

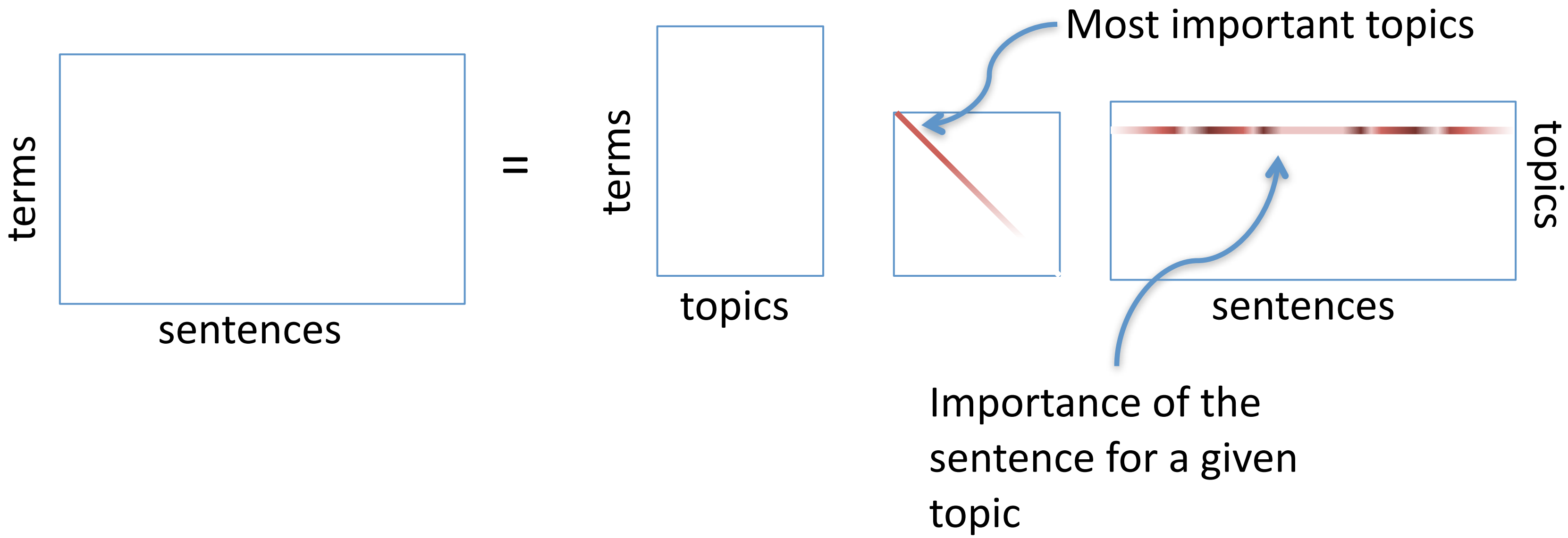
- LSA-based summarisation...



- ... bears similarities with quantum probabilities

Motivations

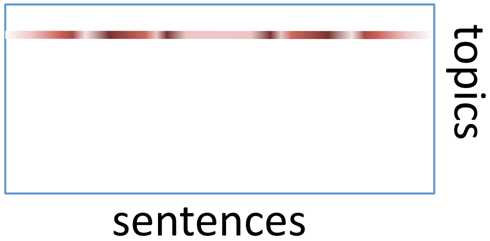
- LSA-based summarisation...



- ... bears similarities with quantum probabilities

Analysis

- We assume
  - one fragment = one sentence
  - set of all sentences = density over topics
- Past methods:
  - Probability of being of a given topic (Gong et al., 2001)
  - Probability of being about one of the topics (Steinberger et al., 2004)



Limitations of previous approaches

- Limitations
  - Multi-document summarisation
  - A sentence can “belong” to different topics (Gong)
  - How to select the dimension (Steinberger)?
  - The same topic can be sampled again and again (Steinberger)

Proposed method

- The documents define the density over atomic topics (uniform distribution over documents)
- Select the set of sentences that cover the topics

$$S^* = \underset{s_1,\ldots,s_n}{\operatorname{argmax}}\, q\left(\mathcal{S}_{s_1,\ldots,s_n} | D\right)$$

- Advantages:
  - Gives a probabilistic interpretation of previous work
  - Address the limitations of previous methods
  - Copes with multiple documents
  - Can be extended to topic-biased summarisation

Results (Rouge-2)

	2005	2006	2007
Best in DUC	**0.07431**	**0.09513**	**0.12285**
Gong (*)	0.072948	0.089894	0.112884
Murray (*)	0.072874	0.089882	0.112107
Ozsoy (*)	0.066421	0.082974	0.101842
QIA	**0.073412 (2)**	**0.090414 (2)**	**0.113445 (5)**

(*) Using document normalisation

COMPUTING QUANTUM PROBABILITIES

Computing probabilities

- We want to compute

$$Pr(S|\mathcal{V}) = \sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V}) q(S|\varphi) = \sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V}) \|\hat{S}\varphi\|^2$$

- This can be expressed using a "density operator"

$$Pr(\mathcal{S}|\mathcal{V}) = \sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V}) \varphi^\top \hat{\mathcal{S}} \varphi$$

$$= \sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V}) \ \mathrm{tr}(\hat{\mathcal{S}} \varphi \varphi^\top)$$

$$= \mathrm{tr}(\hat{\mathcal{S}} \sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V}) \varphi \varphi^\top)$$

Density

Low rank approximation

Low rank approximation

- We compute a low rank eigenvalue decomposition

$$\sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V}) \approx \sum_{i=1}^{K} \sigma_i \psi_i \psi_i^\top$$

  - The vectors *u* span the subspace of the vectors in the ensemble
  - Useful to get rid of noise

Kernel approach

- Motivations
  - Easier to manipulate:
    - semantic kernels (e.g. using WordNet)
    - composite kernels
  - The "real" feature space is too big
    - tensor spaces

- A kernel just needs to define an inner product in some Hilbert space, e.g.

$$k(\varphi_1^{(1)} \otimes \varphi_2^{(1)}, \varphi_1^{(2)} \otimes \varphi_2^{(2)}) = \varphi_1^{(1)} \cdot \varphi_1^{(2)} \times \varphi_2^{(1)} \cdot \varphi_2^{(2)}$$

Kernel approach

- Problem
  - We need to approximate $\sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V})\varphi\varphi^\top$
- Solution
  - Kernel-EVD

$$\sum_{\varphi \in \mathcal{V}} p(\varphi|\mathcal{V})\varphi\varphi^\top \approx \sum_{i=1}^{K} \sigma_i \left(\sum_{\varphi \in \mathcal{V}} \alpha_{i,\varphi}\varphi\right)\left(\sum_{\varphi \in \mathcal{V}} \alpha_{i,\varphi}\varphi\right)^\top$$

  - All the "quantum probabilities" can be computed using only the kernel

ONGOING & FUTURE WORK
CONCLUSION

Ongoing work

- Theoretic & general themes
  - Kernel approach
  - Validating the hypothesis of an IN space
  - Topical space
- ad-hoc IR
  - Query representation
  - Interaction, Diversity & novelty
- Other
  - Summarisation
  - Image retrieval (tensor space)

Limitations

- No principled framework for the definition of topical aspects and the computation of a query representation
- Low-rank approximations might be a problem for high frequency / very ambiguous terms in ad-hoc IR
- Computational complexity

Potential & Interesting things

- Links probability and geometric approaches in Information Access
  - Cosine similarity in IR
  - LSA-based approaches in summarisation
- Multi-dimensionality = diverse topics (or information needs)
  - interaction, diversity, novelty
- Kernel approaches will strengthen the QIA framework

Selected related works

- Driving idea: C. J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, 2004.
- Other "quantum inspired" works:
  - M. Melucci. *A basis for information retrieval in context*. ACM TOIS, 26(3), 2008.
  - G. Zuccon and L. Azzopardi. *Using the Quantum Probability Ranking Principle to Rank Interdependent Documents*. In ECIR. Springer, 2010.
- Multi-dimensional documents:
  - L. Che, J. Zen, and N. Tokud. *A "stereo" document representation for textual information retrieval*. JASIST, 5, 2006.
  - G. Murray, S. Renals, and J. Carletta. *Extractive summarization of meeting recordings*. In ECSCT, 2005.
- Negative RF and multiple query vectors: X. Wang, H. Fang, and C. Zhai. *A study of methods for negative relevance feedback*. In SIGIR. ACM, 2008.
- HAL spaces: C. Burgess, K. Livesay, and K. Lund. Explorations in context space: Words, sentences, discourse. Discourse Processes, 2-3, 1998.
- Subspace representation: P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE TPAMI, 19(7), 1997.

Thanks! More information?

CIKM 2010 proceedings

B. Piwowarski, I. Frommholz, M. Lalmas, and K. van Rijsbergen.
What quantum theory can bring to IR?

My Website

http://www.bpiwowar.net/quantum-ir/

Source code available at

http://sourceforge.net/projects/qir/