Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization

Alessandro Sordoni, Yoshua Bengio and Jian-Yun Nie
DIRO, Université de Montréal
Montréal, Québec
{sordonia, bengioy, nie}@iro.umontreal.ca

Abstract

In web search, users queries are formulated using only few terms and term-matching retrieval functions could fail at retrieving relevant documents. Given a user query, the technique of query expansion (QE) consists in selecting related terms that could enhance the likelihood of retrieving relevant documents. Selecting such expansion terms is challenging and requires a computational framework capable of encoding complex semantic relationships. In this paper, we propose a novel method for learning, in a supervised way, semantic representations for words and phrases. By embedding queries and documents in special matrices, our model disposes of an increased representational power with respect to existing approaches adopting a vector representation. We show that our model produces high-quality query expansion terms. Our expansion increase IR measures beyond expansion from current word-embeddings models and well-established traditional QE methods.

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, Nie, and Bengio 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.

Related work

In this section, 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.

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, He, and Nie 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, clickthrough data has shown great success as it can naturally bridge the gap from queries terms to documents terms. Recently, Gao et al. (Gao and Nie 2012; Gao, He, and Nie 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).

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 (NMF) (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, Yang, and Yu 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.

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.

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, $\tilde{x}_\kappa = Ux_\kappa$, $\tilde{x}_\kappa \in \mathbb{R}^K$. 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:

$$\tilde{q} = Uq = Z_q^{-1} \sum_{\kappa \in Q} Ux_\kappa = Z_q^{-1} \sum_{\kappa \in Q} \tilde{x}_\kappa, \qquad (1)$$

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:

$$s^{SSI}(Q, D) = q^T(U^T U + I)d = \tilde{q}^T \tilde{d} + q^T d. \qquad (2)$$

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):

$$L_{\mathcal{D}}^{SSI}(U) = \sum_{l=1}^{L}[1 - s^{SSI}(Q_l, D_l) + s^{SSI}(Q_l, D_c)]_+ \quad (3)$$

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:

$$-\frac{\partial L_{\mathcal{D}}^{SSI}}{\partial \tilde{x}_q} \approx \tilde{d}_l - \tilde{d}_c, \quad -\frac{\partial L_{\mathcal{D}}^{SSI}}{\partial \tilde{x}_d}, \quad \approx \tilde{q}_l, -\frac{\partial L_{\mathcal{D}}^{SSI}}{\partial \tilde{x}_c} \approx -\tilde{q}_l, \quad (4)$$

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.

Quantum Entropy Minimization

In order to learn the latent embeddings, we stem from the computational framework proposed by the recent Quantum Language Modeling approach for IR (QLM) (Sordoni, Nie, and Bengio 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 matrices 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:

$$\mathcal{L}_Q(W) = \sum_{\kappa \in Q} \log \operatorname{Tr} W \Pi_\kappa, \quad (5)$$

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, Nie, and Bengio 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:

$$\mathcal{L}_Q^I(W) = \sum_{\kappa \in Q} \operatorname{Tr} W \widetilde{\Pi}_\kappa \quad (6)$$

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 low. 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. For the purpose of this work however, we found that this linear approximation works well in practice.

The maximization of Eq. 6 is performed by enforcing the unit-trace constraint $\operatorname{Tr} W = 1$ through a Langrangian multiplier λ. We have:

$$\mathcal{L}_Q^I(W) = \sum_{\kappa \in Q} \operatorname{Tr} W \widetilde{\Pi}_\kappa - \lambda (\operatorname{Tr} W - 1) \quad (7)$$

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:

$$\widetilde{W}_Q = {N_Q}^{-1} \sum_{\kappa \in Q} \widetilde{\Pi}_\kappa = {N_Q}^{-1} \sum_{\kappa \in Q} \tilde{x}_\kappa \tilde{x}_\kappa^T, \qquad (8)$$

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. 8, 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. 8, 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:

$$s(Q, D) = \operatorname{Tr} W_Q \log W_D, \qquad (9)$$

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. 9 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:

$$\log W \approx \log^{\mathrm{I}} W = W - I_K. \qquad (10)$$

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:

$$s^{QEM}(Q, D) = \operatorname{Tr} W_Q (W_D - I_K) \overset{rank}{=} \operatorname{Tr} W_Q W_D, \quad (11)$$

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:

$$L_{\mathcal{D}}^{QEM}(U) = \sum_{l=1}^{L} [m - s^{QEM}(Q_l, D_l) + s^{QEM}(Q_l, D_c)]_+. \qquad (12)$$

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. 8, we can rewrite our scoring function as:

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

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 $\tilde{x}_{q_l}, \tilde{x}_{d_l}$ and $\tilde{x}_{d_c}$ the embedding of a concept appearing in the query, in the relevant document and in the non relevant document respectively. Our updates are:

$$\begin{aligned} -\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_q} &\approx \tilde{x}_q^T (\widetilde{W}_{D_l} - \widetilde{W}_{D_c}), \\ -\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_d} \approx \tilde{x}_d^T \widetilde{W}_Q, \quad -\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_c} &\approx -\tilde{x}_c^T \widetilde{W}_Q, \end{aligned} \qquad (13)$$

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. 8:

$$-\frac{\partial L_{\mathcal{D}}^{QEM}}{\partial \tilde{x}_q} \approx N_{D_l}^{-1} \sum_{\kappa \in D_l} (\tilde{x}_\kappa^T \tilde{x}_q) \tilde{x}_\kappa - N_{D_c}^{-1} \sum_{\eta \in D_c} (\tilde{x}_\eta^T \tilde{x}_q) \tilde{x}_\eta. \quad (14)$$

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 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.

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

Table 1: Number of anchors, concepts, unigram and bigram concepts in the anchor log used in the experiments.

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	$\exp \mathrm{Tr}\, \widetilde{W}_Q\, \widetilde{\Pi}_\kappa \approx \exp \sum_{\eta\in Q} (\tilde{x}_\kappa^T \tilde{x}_\eta)^2$

Table 2: Explicit parameterizations of the probability of an expansion concept given the query for each of the models.

Experimental study

Experimental setup

All our experiments were conducted using the open source Indri search engine (http://www.lemurproject.org). 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, Allan, and Carterette 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.

Anchor log The studies asserting the efficiency of click-through data for QE nearly all make use of proprietary query logs (Gao and Nie 2012; Gao, He, and Nie 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 (http://www.wikipedia.org). 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 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:

$$s^{KL}(Q, D) = \sum_{\kappa\in\mathcal{V}} p(\kappa|\theta_Q) \log p(\kappa|\theta_D) \tag{15}$$

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:

$$p(\kappa|\tilde{\theta}_Q) = \lambda\, p(\kappa|\theta_Q) + (1-\lambda)\, p(\kappa|\theta_E), \tag{16}$$

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 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 $\mathrm{Tr}\, \widetilde{W}_Q \widetilde{\Pi}_\kappa$. 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. 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.

Results

Table 3 resumes all our experimental results. First of all, we note that all the expansion methods increase significantly on

Method	WT-10			WT-11			WT-12		
	nDCG@10	ERR@10	MAP	nDCG@10	ERR@10	MAP	nDCG@10	ERR@10	MAP
LM	.0850	.0443	.1069	.1341	.0613	.0894	.0738	.1087	.1047
CTM	.0954	.0494	.1128	.1278	.0611	.0936	.0837	.1144	.1095
SSI_{100}	.0877	.0437	.1123	.1331	.0624	.0882	**.1063**	.1475	.1200
QEM_{100}	**.1091**$_s$	**.0583**$_{cs}$	**.1137**	**.1514**$_{cs}$	**.0727**$_{cs}$	**.1002**$_s$	.1040$_c$	**.1488**$_c$	**.1210**$_c$
	(+14.3/+24.4)	*(+18/+33.4)*	*(+0.7/+1.2)*	*(+18.0/+13.7)*	*(+19.0/+16.5)*	*(+7.0/+13.6)*	*(+24.2/-0.2)*	*(+30.0/+.1)*	*(+10.5/+.08)*

Table 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.

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

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 choosing a more appropriate ranking loss such as proposed in (Weston, Bengio, and Usunier 2011) and found that the performance of QEM can be further increased by a significant amount with respect to classical CTM and SSI. We also found that varying the number of embedding dimensions did not help on this particular dataset. We argue that this would be useful for larger datasets and thus the automatic setting of appropriate dimensions will be an interesting research in the future.

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), an embedding model that allocates 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 interesting 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.

Acknowledgments

We would like to thank NSERC, Compute Canada, and Calcul Québec for providing computational resources.

References

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

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

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

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

Blei, D. M.; Ng, A. Y.; Jordan, M. I.; and Lafferty, J. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:2003.

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

Chapelle, O.; Metlzer, D.; Zhang, Y.; and Grinspan, P. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, 621–630. New York, NY, USA: ACM.

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

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

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

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

Gao, J., and Nie, J.-Y. 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, 1:1–1:10. New York, NY, USA: ACM.

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

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

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

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

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

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *CoRR* abs/1310.4546.

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

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

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

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

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

Smucker, M. D.; Allan, J.; and Carterette, B. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 623–632.

Sordoni, A.; Nie, J.-Y.; and Bengio, Y. 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, 653–662. New York, NY, USA: ACM.

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

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

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