Deep Neural Architecture for Multi-Modal Retrieval based on Joint Embedding Space for Text and Images

Saeid Balaneshin-kordan
Wayne State University
Detroit, MI, USA
saeid@wayne.edu

Alexander Kotov
Wayne State University
Detroit, MI, USA
kotov@wayne.edu

ABSTRACT

Recent advances in deep learning and distributed representations of images and text have resulted in the emergence of several neural architectures for cross-modal retrieval tasks, such as searching collections of images in response to textual queries and assigning textual descriptions to images. However, the multi-modal retrieval scenario, when a query can be either a text or an image and the goal is to retrieve both a textual fragment and an image, which should be considered as an atomic unit, has been significantly less studied. In this paper, we propose a gated neural architecture to project image and keyword queries as well as multi-modal retrieval units into the same low-dimensional embedding space and perform semantic matching in this space. The proposed architecture is trained to minimize structured hinge loss and can be applied to both cross- and multi-modal retrieval. Experimental results for six different cross- and multi-modal retrieval tasks obtained on publicly available datasets indicate superior retrieval accuracy of the proposed architecture in comparison to the state-of-art baselines.

CCS CONCEPTS

• **Information systems → Image search**;

KEYWORDS

Multi-Modal IR, Cross-Modal IR, Deep Neural Networks

ACM Reference Format:
Saeid Balaneshin-kordan and Alexander Kotov. 2018. Deep Neural Architecture for Multi-Modal Retrieval based on Joint Embedding Space for Text and Images. In *Proceedings of WSDM 2018: The Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3159652.3159735

1 INTRODUCTION

Images and text are an integral part of the Web, from photo sharing and social media platforms to on-line encyclopedias. However, Web search systems still consider images as a separate vertical from text and provide only text-to-image (T→I) search functionality. Yet, the spectrum of information needs of Web search system users goes well beyond text-to-image searches and includes the search tasks, in which pairs of a textual fragment and an image form *atomic retrieval units*, such as in *image-to-image and text* (I→IT) and *text-to-image and text* (i.e. T→IT) retrieval scenarios. Images rarely exist without text and, as illustrated in Figure 1, often convey complementary information. Therefore, it is natural to consider both text and image as one retrieval unit.

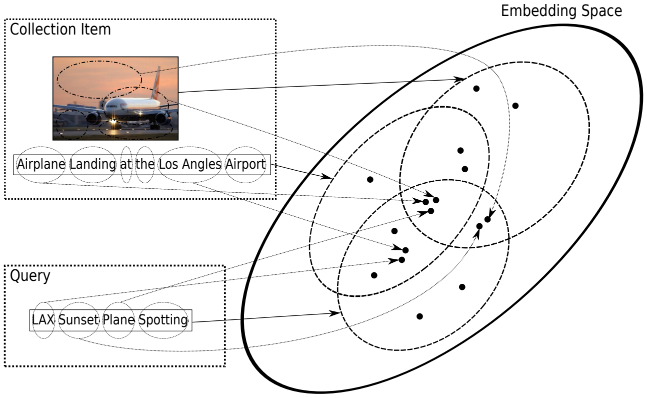

Embedding Space
Collection Item
Airplane Landing at the Los Angles Airport
Query
LAX Sunset Plane Spotting

Figure 1: Projection of textual and visual components of an example query and multi-modal retrieval unit into the space of concept embeddings. The query term "plane" can be matched in both textual and visual components of a given retrieval unit, the query term "LAX" can be matched only in its textual component, while the term "sunset" can only be matched in its visual component.

These multi-modal retrieval scenarios are facing the same fundamental problem of semantic matching of queries to retrieval units, as textual information retrieval (IR). In the case of textual IR, this problem is typically addressed by projecting sparse bag-of-words representations of queries and retrieval units (e.g., documents, passages, sentences) onto dense continuous representations (i.e., embedding vectors), which capture their semantics in a low-dimensional space, and matching them in this space [7, 50]. Shallow neural architectures [37] trained with the goal of making embeddings of words that frequently appear in the same context to be close to each other have emerged as a computationally efficient way of obtaining word embeddings. At the same time, methods utilizing word embeddings for document [12] and query [2, 31] expansion as well as in pseudo-relevance feedback [9, 59] have demonstrated their effectiveness in addressing the problem of vocabulary gap

WSDM 2018, February 5–9, 2018, Marina Del Rey, CA, USA

ACM ISBN 978-1-4503-5581-0/18/02…$15.00
https://doi.org/10.1145/3159652.3159735

in textual IR. Furthermore, several neural architectures based on Convolutional Neural Network (CNN) [33] and Long-short Term Memory Network (LSTM) [20], which take word embedding based representations of queries and documents as input to estimate their relevance, have been recently proposed for Web search [38], adhoc document retrieval [15], microblog retrieval [41] and question answering [51].

Deep neural architectures based on CNNs [33] have also been successfully applied to unsupervised feature extraction and achieved state-of-the-art performance for many computer vision tasks, such as image classification [30, 47] and image recognition [18]. Significant progress in utilizing neural networks for image classification and learning of word embeddings led to the emergence of hybrid neural architectures for cross-modal retrieval tasks, such as generating image descriptions (i.e., captions) [10, 24–27, 36, 44, 49], some of which surpassed human performance for this task. A substantial amount of research also focused on efficient cross-modal retrieval. In particular, hashing-based methods [3, 4, 23, 34, 53, 55] apply hashing to transform different modalities into the same Hamming space and learn quantizers to convert the isomorphic latent features into compact binary codes, which provide a compromise between efficiency and accuracy of cross-modal retrieval. However, the problem of *multi-modal retrieval* (when retrieval units include different modalities) has been significantly less studied.

Drawing inspiration from the success of image captioning, we propose a *gated neural architecture* called JEMR (**J**oint-**E**mbedding for **M**ulti-modal **R**etrieval) to represent image and keyword queries as well as multi-modal retrieval units in the space of word embeddings and semantically match them in that space, as illustrated in Figure 1. The proposed architecture consists of the embedding and relevance matching layers. In the *embedding layers*, the feature vector extracted by a deep CNN for a query or a collection item image is used as the initial hidden state of LSTM to generate the embedding vectors corresponding to image descriptions, which are then used as input to the *relevance matching layers*. Both embedding and relevance matching layers are *jointly trained to minimize a structured hinge loss*. The proposed architecture also includes adaptive gating units that regulate information flow between the embedding and matching layers.

Retrieval is done based on a nearest neighbor search method to efficiently find collection items that are the most similar to a given query, as measured by the output of the matching layers. The proposed architecture is highly modular and can be easily adapted to many cross- and multi-modal retrieval scenarios.

The remainder of this paper is organized as follows. In Section 2, we provide a brief overview of the relevant prior work. The proposed deep neural architecture along with the search method are discussed in Section 3. Experimental results are presented and analyzed in Section 4. Section 5 concludes the paper.

2 RELATED WORK

Prior to discussing the details of the proposed method, we provide an overview of the recent research on neural architectures in textual IR, learning multi-modal representations and cross-modal IR.

Neural architectures in textual IR. Deep learning methods are rapidly gaining popularity in textual IR and related fields, such as question answering. As the first step of these methods, documents and queries are typically transformed into different representations, such as letter trigrams [22, 42], word embeddings [37] or matching histograms [15]. These representations are then used as input to fully connected feed-forward [15, 22], convolutional [42] or recurrent [51] neural networks for estimating relevance of documents to queries. Neural architectures for textual information retrieval tasks can also include gating units [15], which allow to directly incorporate additional relevance signals and heuristics, such as the importance weight or inverse document frequency of a query term.

Learning multi-modal representations. Similar to text retrieval, the first step of cross-modal retrieval methods typically involves obtaining dense representations for textual and visual modalities. The state-of-the-art way to obtain embedding of an image is to use activations in a penultimate layer of deep neural architectures for object recognition [30], which typically consist of several layers of convolutional filtering, local contrast normalization and max-pooling followed by several fully connected layers, after training those architectures on large image collections, such as ImageNet [40]. A variety of methods can be used to obtain dense representations for textual modality. Besides word embedding methods [37, 39], textual modality in multi-modal retrieval tasks can also be represented using a letter-trigram matrix [10]. Alternatively, handcrafted textual features, such as bag-of-words counts and LDA topics, can be passed through several fully connected layers to obtain word embeddings [23].

Linear [57] or non-linear [11, 52] mappings can be learned to convert independently obtained embeddings of images and words into multi-modal embeddings in the same semantic space for a particular cross-model task, such as image captioning or retrieval. Alternatively, zero-shot learning methods train direct mappings of image embeddings into the space of word embeddings [43] and vice versa [6]. Image representations have also been incorporated into the skip-gram model [37] for learning word embeddings enriched with perceptual information [19, 32].

Cross-modal IR. Cross-modal (image-to-text and text-to-image) retrieval methods can be categorized into correlation-, semantic-, and hashing-based ones. Correlation-based methods utilize Canonical Correlation Analysis (CCA) [17] and its variants, such as kernel CCA [21] and normalized CCA [14], to capture linear and/or non-linear correlations between textual and visual modalities for bi-directional ranking of images and captions. Semantic methods leverage dense multi-modal representations and deep neural architectures. For example, a neural architecture for cross-modal retrieval, which combines one CNN for image representation and one CNN for calculating word-level, phrase-level and sentence-level matching scores between an image and a sentence, was proposed in [35].

A similar task to cross-modal retrieval is image captioning or associating textual descriptions (e.g. sentences [24, 44] or sentence fragments [25]) either with entire images [49] or their fragments [24, 25]. An image captioning method proposed in [10] ranks textual fragments for a given image and vice versa based on the cosine similarity between their embeddings. Captions can also be generated by sampling from a log-bilinear language model conditioned on the embeddings of already generated caption words and the image feature vector [26].

Neural network architectures proposed for image captioning are typically based on the encoder-decoder framework. For example, a deep CNN can be used as an encoder and LSTM as a decoder [49]. A combination of CNN and LSTM was used as an encoder and multiplicative neural language model incorporating linguistic structure was used as a decoder in [27]. Image feature vector obtained by a deep CNN can also be directly incorporated into an RNN [36] for caption generation.

Finally, hashing-based cross-modal retrieval methods [3, 4, 23, 34, 53, 55] learn hash functions that map images and text in the original space into a Hamming space of binary codes, such that the similarity between the objects in the original space is preserved in the Hamming space. Some hashing-based methods [3, 4, 23] also leverage deep CNNs for creating dense representations of images.

Approximate Nearest Neighbor (ANN) [1] search algorithm accompanied by a proper technique to index collection items enables fast and accurate retrieval in a Hamming space. For this reason, ANN is frequently used in cross-modal hashing methods to rank collection items in the order of their similarity to a query. For example, [23] used ANN to accelerate retrieval of binary hashes obtained using CNN, while [58] used ANN coupled with a sensitive Jaccard similarity metric to efficiently search in sparse and high-dimensional space of cross-modal codes.

3 METHOD

3.1 Proposed Neural Architecture

Without loss of generality, the proposed neural architecture is discussed for the case of text-to-image and text (T→IT) retrieval task (i.e., retrieving a multi-modal collection item with a textual modality d_t and a visual modality d_v given a textual keyword query q). However, we would like to emphasize that the proposed architecture is general and can be easily adapted to other cross- and multi-modal retrieval tasks (e.g. when collection items have only textual or visual modality).

The proposed neural architecture consists of two types of layers. The embedding layers (illustrated in Figure 2 for images) extract concept and topic embeddings from textual and visual modalities of queries and collection items, while the relevance matching layers (illustrated in Figure 3) calculate the relevance score of a query to a collection item.

3.1.1 Embedding layers. The goal of these layers is to create dense low-dimensional representations of a query (q) and different modalities of a collection item (d_t and d_v). The output of these layers consists of the two sets of matrices of low-dimensional representations $\mathcal{S}'$ and $\mathcal{S}''$ that are used later in estimating the relevance of a collection item to a query. Both of these matrices contain embeddings of concepts from a controlled vocabulary. In our experiments, this controlled vocabulary consists of the words in the titles of all English Wikipedia articles. Each embedding vector in these matrices is a representation of a concept, which can be a word in a query or collection item's text (textual concept) or an object in a collection item's image (visual concept).

Table 1: Summary of notation

Var.	Description
q	query
d	multi-modal collection item
d_t	textual modality of d
d_v	visual modality of d
$\mathbf{q}_i$	i^th concept embedding vector of q
$\mathbf{d}_{\mathrm{t},i}$	i^th concept embedding vector of d_t
$\mathbf{d}_{\mathrm{v},i}$	i^th concept embedding vector of d_v
$\mathbf{Q}'$	concept embedding matrix for q
$\mathbf{D}'_\mathrm{t}$	concept embedding matrix for d_t
$\mathbf{D}'_\mathrm{v}$	concept embedding matrix for d_v
$\mathbf{Q}''_i$	i^th topic embedding matrix for q
$\mathbf{D}''_{\mathrm{t},i}$	i^th topic embedding matrix for d_t
$\mathbf{D}''_{\mathrm{v},i}$	i^th topic embedding matrix for d_v
$p(d\|q)$	probability of d being relevant to q

The first set, $\mathcal{S}' = \{\mathbf{Q}', \mathbf{D}'_\mathrm{t}, \mathbf{D}'_\mathrm{v}\}$, consists of *concept embedding matrices* that are used for computing the matching scores *at the concept level*. The matrices in this set contain embeddings of concepts in a query, collection item's text or collection item's image, respectively. The second set, $\mathcal{S}'' = \{\mathbf{Q}''_1, \mathbf{Q}''_2, \ldots, \mathbf{D}''_{\mathrm{t},1}, \mathbf{D}''_{\mathrm{t},2}, \ldots, \mathbf{D}''_{\mathrm{v},1}, \mathbf{D}''_{\mathrm{v},2}, \ldots\}$, consists of *topic embedding matrices* that are used for computing the matching scores *at the topic level*. We obtain each of the matrices in this set by clustering the embedding vectors of all concepts in a given modality of a query or collection item (e.g. clustering embeddings of words in a keyword query). We use cosine similarity as a measure of semantic similarity of concept vectors. For the sake of notational simplicity, in Figure 3, we denote the matrices $\mathbf{Q}'$, $\mathbf{D}'_\mathrm{t}$ and $\mathbf{D}'_\mathrm{v}$ in the set $\mathcal{S}'$ as QTC (Query Text Concepts), CTC (Collection Text Concepts) and CIC (Collection Image Concepts), respectively, and the matrices $\mathbf{Q}''_i$, $\mathbf{D}''_{\mathrm{t},i}$ and $\mathbf{D}''_{\mathrm{v},i}$ in the set $\mathcal{S}''$ as QTT$_i$ (Query Text Topic i), CTT$_i$ (Collection Text Topic i) and CIT$_i$ (Collection Image Topic i).

If the number of concepts in q, d_t and d_v is $|\mathbf{Q}'|$, $|\mathbf{D}'_\mathrm{t}|$ and $|\mathbf{D}'_\mathrm{v}|$, and k is the size of the embedding vector representing each concept, then the dimensions of $\mathbf{Q}'$, $\mathbf{D}'_\mathrm{t}$ and $\mathbf{D}'_\mathrm{v}$ are $k \times |\mathbf{Q}'|$, $k \times |\mathbf{D}'_\mathrm{t}|$ and $k \times |\mathbf{D}'_\mathrm{v}|$, respectively. In our experiments, we set k to 300. On the other hand, if $\mathbf{Q}'$, $\mathbf{D}'_\mathrm{t}$ and $\mathbf{D}'_\mathrm{v}$ have $|\mathbf{Q}''|$, $|\mathbf{D}''_\mathrm{t}|$ and $|\mathbf{D}''_\mathrm{v}|$ clusters (topics), then $\mathbf{Q}''$, $\mathbf{D}''_\mathrm{t}$ and $\mathbf{D}''_\mathrm{v}$ contain embedding vectors of size k that correspond to each of the topics in q, d_t and d_v.

word2vec [37] embeddings were used to represent the concepts in textual modality of queries that exist in the adopted controlled vocabulary. Embeddings of concepts in the visual modality were obtained by adopting the neural architecture for image captioning proposed in [49] (illustrated in Figure 2), which combines a deep CNN [30] for image feature extraction and LSTM [20] for caption generation. Considering all the words in the adopted controlled vocabulary as candidate visual concepts, we use LSTM to model $p(\mathbf{d}_{\mathrm{v},i+1}|\mathbf{d}_{\mathrm{v},1}, \ldots, \mathbf{d}_{\mathrm{v},i})$, which is the probability of the $(i+1)^\mathrm{th}$ word embedding vector $\mathbf{d}_{\mathrm{v},i+1}$ to be used for representing a visual concept in an image of the collection item, given the word embedding

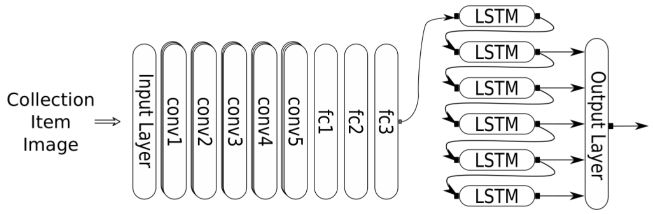


Figure 2: Image captioning layers in the proposed deep neural architecture for $T \rightarrow TI$ task. A combination of convolutional layers conv1-conv5 and fully connected layers fc1-fc3 is used for image feature extraction. LSTM is used for caption generation.

vectors $\mathbf{d}_{v,1}, \ldots, \mathbf{d}_{v,i}$ that have already been generated as visual concepts for the image, and select the word embedding $\mathbf{d}_{v,i+1}$ that maximizes the probability $p(\mathbf{d}_{v,i+1}|\mathbf{d}_{v,1}, \ldots, \mathbf{d}_{v,i})$. This criterion ensures that the selected visual concept best describes a given image in conjunction with previously selected i concepts. As can be seen from Figure 2, in this iterative approach, the first concept vector ($\mathbf{d}_{v,1}$) is generated by maximizing the probability computed directly from the feature vector obtained from the CNN layers ($\mathbf{d}_{v,0}$). We repeat this process until LSTM generates a pre-defined number of visual concept embedding vectors for a given collection item image.

Although hybrid neural architectures have been studied for image captioning [49], the objectives of these architectures are different from this component of our proposed architecture for multi-modal retrieval task. For the image captioning task, LSTM and CNN layers are trained with the goal of generating image descriptions that are the most understandable by humans, while for the multi-modal retrieval task, these layers are trained with the goal of producing the image descriptions that maximize retrieval accuracy.

3.1.2 *Relevance matching layers.*

The goal of these layers is to calculate $p(d|q)$, the probability of a collection item d to be relevant to query q. We further decompose $p(d|q)$ into topic and concept relevance matching scores as:

$$\begin{aligned} p(d|q) \approx\, & \rho(\mathbf{Q}', \mathbf{D}'_t)p(\mathbf{D}'_t|\mathbf{Q}') + \rho(\mathbf{Q}', \mathbf{D}'_v)p(\mathbf{D}'_v|\mathbf{Q}') + \\ & \sum_{j,k} \rho(\mathbf{Q}''_k, \mathbf{D}''_{t,j})p(\mathbf{D}''_{t,j}|\mathbf{Q}''_k) + \rho(\mathbf{Q}''_k, \mathbf{D}''_{v,j})p(\mathbf{D}''_{v,j}|\mathbf{Q}''_k) \quad (1) \end{aligned}$$

where $\rho(\cdot,\cdot)$ computes the prior probabilities for the concept and topical relevance, which are obtained by the gating network described in Section 3.1.3. A similar idea of lexical and semantic matching has been shown to be effective in textual IR [28, 29, 54].

The probabilities $p(\mathbf{D}'_t|\mathbf{Q}')$, $p(\mathbf{D}'_v|\mathbf{Q}')$, $p(\mathbf{D}''_{t,j}|\mathbf{Q}''_k)$ and $p(\mathbf{D}''_{v,j}|\mathbf{Q}''_k)$ are computed as a combination of Bidirectional Long Short-Term Memory (BiLSTM) [48] units, max-pooling and cosine similarities, as shown in Figure 3.

3.1.3 *Gating Network.*

To account for the semantic similarity of individual concept vectors, we use multiple gates to regulate the concept and topic relevance probabilities. In other words, similar to LSTM [20], Highway or Residual Networks [18, 45], we provide connections from the input layer of the relevance matching component to its last layer through the gating units that regulate the information flow from these layers. The gate function $\rho(\Phi, \Psi)$ depends on the sum of L^2-norms of the distances between the elements of $\Phi = [\phi_1, \phi_2, \ldots]$ and $\Psi = [\psi_1, \psi_2, \ldots]$ as:

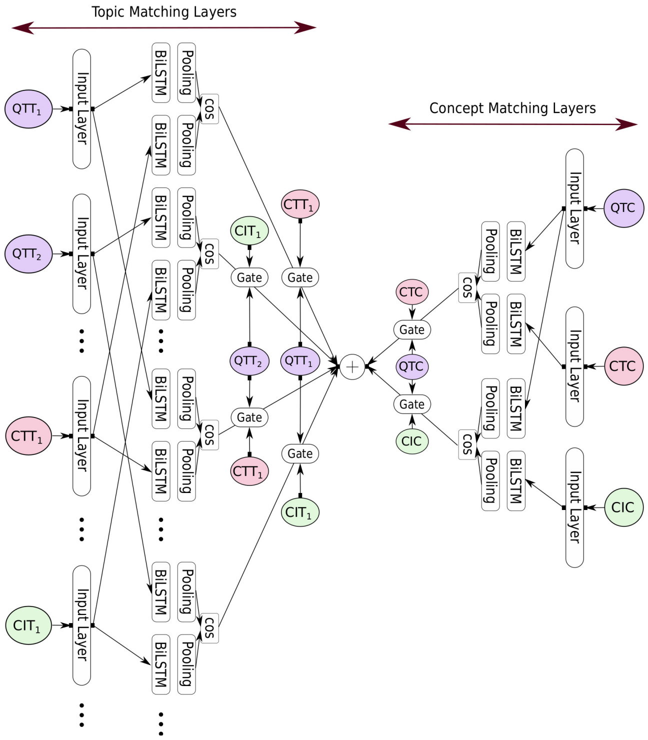


Figure 3: Projection layers in the proposed neural network architecture for $T \rightarrow TI$ task. $\mathbf{QTT}_i$, $\mathbf{CTT}_i$ and $\mathbf{CIT}_i$ are low-dimensional representations of the i-th topic in the query's text, collection item's text and collection item's image, respectively. QTC, CTC and CIC are low-dimensional representations of concepts in query's text, collection item's text and collection item's image.

$$\rho(\Phi, \Psi) \approx 1 - \left(\frac{\sum_l \min_i \|\phi_i - \psi_l\|_2^2}{2|\Psi|} + \frac{\sum_i \min_l \|\phi_i - \psi_l\|_2^2}{2|\Phi|} \right), \quad \Phi, \Psi \in \{\mathbf{Q}', \mathbf{D}'_t, \mathbf{D}'_v, \mathbf{Q}''_k, \mathbf{D}''_{t,j}, \mathbf{D}''_{v,j}\} \quad (2)$$

where $|\Phi|$ and $|\Psi|$ are the number of embedding vectors in Φ and Ψ. Since ϕ_i and ψ_l are normalized vectors, finding $\min_i \|\psi_i - \phi_l\|_2^2$ is equivalent to finding a concept vector in Φ that has the highest cosine similarity with ψ_j. The score $\sum_l \min_i \|\phi_i - \psi_l\|_2^2$ is the sum of distances between all embedding vectors in Ψ and their most similar embeddings in Φ. The second term in (2) accounts for the cases when $|\Psi| \neq |\Phi|$. It can be easily shown that this gate function always has values between zero and one, and it computes similarity between the most similar pairs in two sets of embedding vectors Φ and Ψ. Considering these gating units, the estimated relevance probability of two matrices of embeddings Φ and Ψ is obtained as the product of $p(\Psi|\Phi)$ calculated by the relevance matching layers and $\rho(\Phi, \Psi)$ calculated by the gating units.

3.1.4 Extensions to other Cross- and Multi-modal Retrieval Tasks. Besides T⟶IT (shown in Figure 3), the proposed method is evaluated in Section 4 for five other cross- and multi-modal retrieval tasks (T⟶I, I⟶T, I⟶I, T⟶T and I⟶TI). The architectures for these tasks can be straightforwardly obtained from the one for the T⟶IT task. For example, the neural architecture for the I⟶T task has CNN and LSTM layers associated with the query and an embedding layer associated with the collection item. Parallel projection and matching layers that share the same set of weights can be added to the proposed architecture to extend this architecture to the cases when more than one textual or visual modality is associated with a query or a collection item.

3.2 Training

The training data for the proposed architecture consists of triplets of related and unrelated collection items to a given query. If we designate one of these triplets as (q, d_i^+, d_i^-) and a vector of parameters as θ, then finding θ involves minimizing the following hinge loss:

$$\min_{\theta} \left(\frac{\lambda_o}{2} ||\theta||_2^2 + \sum_{(q,d^+,d^-)\in\mathcal{T}} \max(0, p(d^+|q) - p(d^-|q) + \beta) \right) \quad (3)$$

where $\mathcal{T}$ is the set of triplets in the training data, λ_0 is a constant, and β is a desired margin between the relevance probabilities of relevant and non-relevant collection items with respect to a query. The second term in the objective function of the above optimization problem is our training loss ($\mathcal{L}(\theta)$) which enforces $p(d_i^+|q) > p(d_i^-|q)$. The loss function can also be written in terms of the concept and topic relevance probabilities as:

$$\begin{aligned}\mathcal{L}(\theta) = \sum_{(q,d^+,d^-)\in\mathcal{T}} \Bigg(& \rho(Q', D'_\mathrm{t}) \max(0, p(D'^+_\mathrm{t}|Q') - p(D'^-_\mathrm{t}|Q') + \beta'_\mathrm{t}) \\ & + \rho(Q', D'_\mathrm{v}) \max(0, p(D'^+_\mathrm{v}|Q') - p(D'^-_\mathrm{v}|Q') + \beta'_\mathrm{v}) \\ & + \sum_{j,k} \rho(Q''_k, D''_{\mathrm{t},j}) \max(0, p(D''^+_{\mathrm{t},j}|Q''_k) - p(D''^-_{\mathrm{t},j}|Q''_k) + \beta''_\mathrm{t}) \\ & + \sum_{j,k} \rho(Q''_k, D''_{\mathrm{v},j}) \max(0, p(D''^+_{\mathrm{v},j}|Q''_k) - p(D''^-_{\mathrm{v},j}|Q''_k) + \beta''_\mathrm{v}) \Bigg)\end{aligned}$$

where the prior probabilities and the margin parameters β'_t, β'_v, β''_t and β''_v are independent of the neural network.

The structured loss function in the above equation can be viewed as a sum of four different loss functions. The first two loss functions depend on the probability of relevance of different modalities of a collection item and a query. The last two loss functions depend on the probability of topical relevance of a collection item and a query.

The proposed architecture is trained in several stages. In the first training stage, a shallow neural network is trained by using the skip-gram model [37] over the chosen controlled concept vocabulary. In the second training stage, the image feature extraction network is trained. In the third stage, the parameters of the LSTM layers that generate visual concept vectors are trained. Next, the parameters of the BiLSTM layers are trained, and finally, in the last training stage, the parameters of the relevance matching layers are trained.

In the first training stage, we utilize word2vec vectors pre-trained for 3 million words and phrases on a Google News corpus[1] and prune the table of word embeddings to keep the ones with a word or a phrase that exists in our controlled vocabulary. We use the pruned table as the table of concept vectors. In Stage 2, we train weights for conv1–5 and fc1–3 layers on ILSVRC-2012 dataset[2]. In Stage 3, we use the training data from 2015 MS COCO Image Captioning Challenge [49] to train LSTM layers. In Stage 4, to train the BiLSTM layers, we use the training data from Yahoo Question answering dataset, which is described in Section 4. Finally, as shown in Figure 4, in the last stage, we use the training data from multi-modal retrieval datasets, which are described in Section 4. The parameters of the fc2-fc3 and LSTM and BiLSTM layers are fine-tuned in the last training stage.

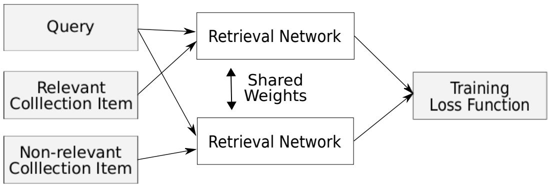


Figure 4: Procedure to train the parameters of the neural network in the last training stage. The upper and lower networks are the same as the network shown in Figure 3.

3.3 Search

We considered the inverse of the relevance matching score between a collection item and a query computed by the proposed architecture as a distance and adopted two distance-based search methods, a brute-force k-nearest neighbor search (k-NN) and approximate k-nearest neighbor (ANN) [16] to find collection items that have the maximum relevance matching scores (or minimum distance) with respect to a given query.

4 EXPERIMENTS

We evaluate our proposed architecture using two datasets, which were chosen with the goals of (1) investigating whether JEMR overfits the training data, (2) evaluating JEMR on a dataset without explicit bridge information between textual and visual modalities of collection items, and (3) evaluating JEMR on a dataset with out-of-sample images and texts. We compare the performance of our proposed method with two state-of-the-art unsupervised and three supervised baselines that leverage deep neural networks. We also investigate the effect of image feature extractors on the performance of our proposed architecture by examining the cases of using image feature extractors other than AlexNet. Finally, we examine the effect of a number of acceleration methods on the speed of the proposed method. The proposed method is compared with the baselines based on Mean Average Precision (MAP) and Precision-Recall curves.

4.1 Datasets

For all experiments in this paper, we use the two datasets based on the ones in [4] and [55]. **NUS-WIDE**[5], the first dataset, is a

[1] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/

[2] https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet

multi-modal dataset with 269,648 image-tags pairs, 5,018 tags and a ground truth for 81 concept categories. Each image in the **NUS-WIDE** or ImageNet dataset [8] belongs to a category. **NUS-WIDE** and ImageNet datasets have 16 of these categories in common. We prune images from **NUS-WIDE** that do not belong to any of these common categories. **NUS-WIDE** contains one textual document for each image, which is obtained by aggregating all its corresponding tags. We obtain the query set for **NUS-WIDE** by randomly selecting 2000 textual documents and 2000 visual documents from **NUS-WIDE**.

The other dataset is **ImageNet-YahooQA** [4], which consists of 10 million images from ImageNet [8] and 300, 000 texts obtained by using Yahoo Query Language API[3]. This dataset is generated by using the tags from the **NUS-WIDE** dataset, i.e., 5,018 tags in **NUS-WIDE** are used as keyword queries to obtain the set of Yahoo QAs. The relevant answers to questions in Yahoo QAs dataset are considered as the textual documents in **ImageNet-YahooQA**. The goal of multi-modal retrieval using this dataset is to find answers from Yahoo QAs dataset that are semantically related to an image query randomly chosen from the ImageNet dataset and vice versa. In other words, only T⟶I and I⟶T retrieval tasks can be examined on this dataset. Multi-modal dataset similar to **ImageNet-YahooQA** is MIRFLICKR-Yahoo Answer dataset introduced in [55]. Similar to the **NUS-WIDE**, we prune **ImageNet-YahooQA** by removing the images in ImageNet that do not have any of 5,018 tags in **NUS-WIDE**. Similar to **NUS-WIDE**, we obtain a query set by randomly selecting 2000 texts and 2000 images from **ImageNet-YahooQA**.

In the last training stage described in Section 3.2, we use the training data from **NUS-WIDE** for experiments on both **ImageNet-YahooQA** and **NUS-WIDE** datasets. To obtain the ground truth for evaluation, we assumed that any pair of query and collection items (either containing image, text or both) are relevant, if both of them belong to at least one of the 16 categories that NUS-WIDE shares with ImageNet.

4.2 Experimental Setup

We used TensorFlow version 1.0.1[4] to implement and train our deep neural architecture on a Linux server with a NVIDIA Tesla K10 GPU with batch size 32 for 100 epochs. We use back-propagation with stochastic gradient descent to train the parameters of our proposed deep neural network. To speed up mini-batch learning, RMSProp was used with a decay of 0.9, and $\epsilon = 1.0$. Hyper-parameters (e.g. ρ, β) were optimized using coordinate ascent based on three-fold cross-validation.

As mentioned earlier, besides using AlexNet as the image feature extractor, we also experimented with ResNet 152 [18], Inception V3 [47] and Inception-ResNet-v2 [46]. We use the pre-trained weights of these CNNs, which are publicly available for Tensorflow[5] and adapt the weights of AlexNet that are publicly available in Caffe[6] to Tensorflow.

4.3 Baselines

In the experiments, we consider the following two unsupervised and three supervised baselines:

`CCA-MV` [13]: extends canonical correlation analysis (`CCA`) to the case of having multiple views of visual, textual and semantic features obtained by clustering words. We use a three-view CCA ("CCA (V+T+C)" in [13]). We use the same list of features adopted in [13] to implement this method.

`CCQ` [34]: is an unsupervised cross-modal hashing based retrieval method that adopts a unified optimization framework to jointly learn the latent space and similarity preserving composite quantization that maximize correlation. Unlike `JEMR`, `CCQ` does not rely on neural networks.

`DSM` [56]: similar to `JEMR` provides real-valued representations for visual and textual modalities. However, `DSM` also uses CNN and hand-crafted features to obtain representations of visual and textual modalities. The hand-crafted features are extracted using topic models and bag-of-words.

`DVSH` [3]: adopts a two-sided deep CNN-LSTM network for joint representation learning and hash coding. One side applies CNN to project images and LSTM to project text into a common subspace and another LSTM network computes the matching score of these projected modalities. The other side, utilizes CNN and LSTM networks to encode the textual and visual modalities of collection items and then it computes the similarity between the generated isomorphic hash codes. The network is trained according to the computed matching score and the similarity of the generated hash codes.

`THN` [4]: adopts CNN and multilayer perceptrons and has a similar training process to `JEMR`, as both architectures utilize the training data from different datasets to train different components of the network. The main goal of using diverse training data is to enable the retrieval system to process queries in a collection that has a different distribution. For example, given a query selected from Yahoo QAs dataset, it allows the retrieval system to obtain relevant images in ImageNet dataset.

`MCNN` [35]: adopts one CNN to learn image representations and Skip-gram method to learn text representations, and another CNN to compute the multi-modal matching scores between a query and collection items. This method performs word-level, phrase-level and sentence-level matching using a matching CNN.

Similar to `JEMR`, `DSM`, `DVSH`, `THN`, and `MCNN` adopt deep neural networks to extract features from query and collection items. Specifically, `DVSH` is similar to `JEMR` in that it uses a hybrid CNN-LSTM network. However, unlike `JEMR`, `DVSH` and `THN` both employ hashing methods in approximating nearest neighbor search. We used 16 bits for hashes created by both `DVSH` and `THN`.

4.4 Results and Discussion

Table 2 summarizes the performance of the proposed architecture (`JEMR`) and six state-of-the-art baselines in terms of MAP for six multi-modal retrieval scenarios (I⟶T, T⟶I, I⟶IT, T⟶IT, I⟶I, and T⟶T) on two datasets (**NUS-WIDE** and **ImageNet-YahooQA**). In **ImageNet-YahooQA** dataset, for the tasks T⟶T and T⟶IT, we obtain the textual query from a set of related answers in Yahoo QAs dataset and the multi-modal collection items from the ImageNet

[3] https://developer.yahoo.com/yql/
[4] https://github.com/tensorflow/tensorflow/releases/tag/v1.0.1
[5] https://github.com/tensorflow/models/tree/master/slim
[6] https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet

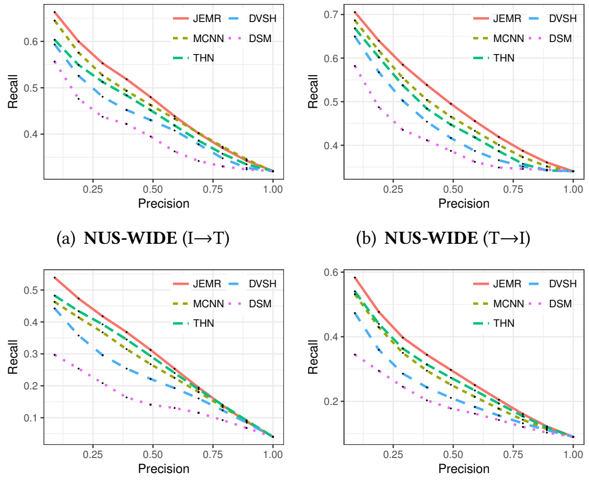
dataset. However, since Yahoo QAs dataset does not contain any images, we do not evaluate the methods for I⟶I and I⟶IT retrieval tasks on the **ImageNet-YahooQA** dataset.

As follows from Table 2, JEMR outperforms all baselines in all of the six retrieval tasks with statistically significant difference. This table also indicates that, on average, the performance improvement of JEMR relative to DSM, DVSH, THN, and MCNN is higher for the I⟶IT, T⟶IT, I⟶I, and T⟶T, than for I⟶T and T⟶I retrieval tasks. This can be explain by the fact that DSM, DVSH, THN, and MCNN do not provide any mechanisms to create embeddings or encodings shared by both modalities for I⟶IT and T⟶IT retrieval tasks.

By experimenting with the two datasets (**NUS-WIDE** and **ImageNet-YahooQA**), we can evaluate the effect of existence of explicit relationships between query and collection items in training of the proposed network. Table 2 reveals that the methods leveraging deep neural networks, i.e., DSM, DVSH, THN, MCNN, and JEMR have higher MAP values than the unsupervised baselines, i.e., CCA-MV and CCQ. It can be also concluded that, on average, these improvements are higher on **ImageNet-YahooQA** than on **NUS-WIDE**. This is because deep neural networks can better learn from diverse training data than the shallow baselines. In particular, we can also observe that the percentage of improvement of JEMR over its best performing baselines is greater on **ImageNet-YahooQA** than on **NUS-WIDE**, which can be an indication of superior ability of JEMR to generalize to the collection items that do not exist in the multimodal retrieval training data. Although DVSH and JEMR both use hybrid deep CNN-LSTM Networks, JEMR has a higher MAP value than DVSH [3]. This can be attributed to the gated structure of JEMR, which enables computing the matching functions via considering local similarity of word embedding vectors as well as global similarity of a collection item to a query.

The results for I⟶I and T⟶T tasks were included in Table 2 in order to analyze the influence of an additional textual or visual modality (i.e., I⟶I, and T⟶T tasks) on the performance of JEMR and its baselines. Based on this table, we can conclude that the proposed architecture and baseline methods have on average 2.8% higher MAP values for the T⟶IT task than for T⟶T task on **NUS-WIDE** and **ImageNet-YahooQA** datasets. This improvement is statistically significant for all methods and we can deduce that considering additional visual modality can improve the quality of the T⟶T retrieval task. Following the same reasoning, we can conclude that with on average 4.2% higher MAP values on **NUS-WIDE** dataset, considering additional textual modality for the I⟶I task also provides statistically significant improvement for all methods.

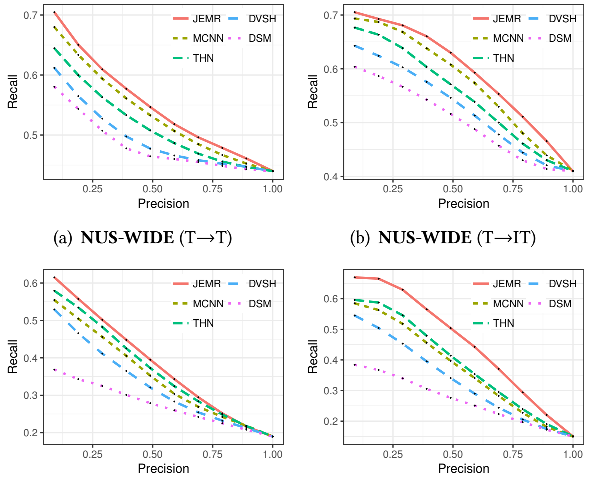
Similar to the observations made from Table 2, the precision-recall curves in Figures 5 and 6 indicate superior performance of JEMR over its best-performing baselines for I⟶T, T⟶I, T⟶T, and T⟶IT tasks on **NUS-WIDE** and **ImageNet-YahooQA** datasets. These figures indicate that JEMR and THN have greater performance improvement in comparison to the other baselines on the **ImageNet-YahooQA** dataset than on **NUS-WIDE**, which is an indication of better generalization of these two methods to the unseen data. Figure 6 is provided to compare precision-recall curves when the retrieval method can get access to an additional visual component of the collection items, in addition to their textual components (by comparing T⟶T and T⟶IT tasks). Figure 6 indicates that JEMR demonstrates greater improvement over the baselines for T⟶IT than for T⟶T task, which is because JEMR takes into account local information between modalities of collection items.



(a) **NUS-WIDE** (I⟶T) (b) **NUS-WIDE** (T⟶I)

(c) **ImageNet-YahooQA** (I⟶T) (d) **ImageNet-YahooQA** (T⟶I)

Figure 5: Precision-recall curves for the proposed method and the baselines for the NUS-WIDE and ImageNet-YahooQA datasets for I⟶T and T⟶I tasks.



(a) **NUS-WIDE** (T⟶T) (b) **NUS-WIDE** (T⟶IT)

(c) **ImageNet-YahooQA** (T⟶T) (d) **ImageNet-YahooQA** (T⟶IT)

Figure 6: Precision-recall curves for the proposed method and the baselines for the NUS-WIDE and ImageNet-YahooQA datasets for T⟶T and T⟶IT tasks.

To investigate the effectiveness of applying accelerated search on performance of the proposed method, we combined JEMR with k-NN and approximate nearest neighbors (ANN) search methods [16] and report the results in Table 3. JEMR+KNN uses a brute-force search

Table 2: Performance of the proposed architecture (JEMR) and the baselines, as measured by MAP. $\star$ and † indicate statistically significant improvements according to Fisher's randomization test with $p < 0.05$ over the best performing baselines THN [4] and MCNN [35], respectively. The percentage improvement of JEMR over THN [4] and MCNN [35] are shown in parenthesis.

Dataset	Task	CCA-MV [13]	CCQ [34]	DSM [56]	DVSH [3]	THN [4]	MCNN [35]	JEMR
NUS-WIDE	**I→T**	0.4261	0.4877	0.6415	0.7236	0.7268	0.7381	**0.7538***(3.71% / 2.13%)
	T→T	0.5381	0.6261	0.6801	0.7109	0.7321	0.7402	**0.7833***†(6.99% / 5.82%)
	I→IT	0.4563	0.5115	0.6759	0.6983	0.7494	0.7528	**0.7984***†(6.54% / 6.06%)
	T→IT	0.5449	0.6317	0.6949	0.7532	0.7649	0.7693	**0.8092***†(5.79% / 5.19%)
	I→I	0.4258	0.4652	0.6568	0.6745	0.7093	0.7113	**0.7563***†(6.63% / 6.33%)
	T→I	0.4481	0.5165	0.6782	0.7468	0.7572	0.7795	**0.7825***(3.34% / 0.38%)
ImageNet-YahooQA	**I→T**	0.1145	0.2152	0.4142	0.5631	0.6132	0.6092	**0.6557***†(6.93% / 7.63%)
	T→I	0.1361	0.2389	0.4209	0.5938	0.6341	0.6278	**0.6804***†(7.30% / 8.38%)
	T→IT	0.2595	0.3725	0.5665	0.6102	0.6383	0.6303	**0.6929***†(8.55% / 9.93%)
	T→T	0.2437	0.3571	0.4869	0.6029	0.6314	0.6282	**0.6872***†(8.84% / 9.39%)

method to find all nearest neighbors, while JEMR+ANN applies approximations, which result in a smaller number of collection items than k-NN that have to be scored with the relevance matching layers of the proposed architecture. In JEMR+ANN and JEMR+KNN, the collection items are represented by a single concept vector that is an average of all of its concept vectors. In this experiment, we set the size of the neighborhood and the maximum number of iterations to be 1000 for both JEMR+ANN and JEMR+KNN.

Table 3 indicates that, although JEMR uses a brute-force search method and examines all collection items, its MAP is, on average, around 5% higher than MAP of JEMR+ANN. On the other hand, JEMR+KNN has, on average, around 5% higher MAP than JEMR+ANN. Without considering the time to locate objects of collection items in the embedding space, we observed that, on average, JEMR is around **1200** times and JEMR+KNN is around **150** times slower than JEMR+ANN. Therefore, we can conclude that using an approximate search method can substantially decrease the search time with a negligible degradation in accuracy.

Table 3: MAP of JEMR on NUS-WIDE dataset when different search methods are used.

Task	I→T	T→I	I→IT	T→IT
JEMR+ANN	0.7381	0.7303	0.7723	0.7926
JEMR+KNN	0.7477	0.7791	0.7921	0.8035
JEMR	0.7538	0.7825	0.7984	0.8092

Table 4 reflects the impact of state-of-the-art deep CNN architectures, such as ResNet 152[18], Inception V3[47] and Inception-ResNet-v2[46] on performance of JEMR, if they are utilized for image feature extraction instead of AlexNet. Similar to the training of AlexNet, the parameters of these three networks are pre-trained using the training data from ImageNet dataset with the parameters of their last two layers fine-tuned based on the training data, once using MS-COCO and once using **NUS-WIDE** datasets. Table 4 indicates that utilizing Inception-ResNet-v2, ResNet 152 and Inception V3 results in higher MAP than AlexNet, with Inception-ResNet-v2 producing a statistically significant improvement in MAP over the other networks, according to the Fisher's randomization test with $p < 0.05$.

Table 4: MAP of JEMR on NUS-WIDE dataset when different CNN networks are used in the proposed architecture.

CNN	I→T	T→I	I→IT	T→IT
AlexNet	0.7538	0.7825	0.7984	0.8092
ResNet 152	0.7932	0.8053	0.8178	0.8223
Inception V3	0.7986	0.8114	0.8250	0.8377
Inception-ResNet-v2	0.8049	0.8230	0.8287	0.8424

5 CONCLUSIONS

This paper presents a novel neural architecture for multi-modal retrieval when the query has a single modality and collection items can have multiple modalities. The proposed architecture utilizes a hybrid LSTM-CNN network to project the visual modalities and the skip-gram model to project the textual modalities into a common subspace, which contains embeddings of words in the textual modalities and embeddings of words that describe the visual modalities. The proposed architecture also includes a gating network to regulate the information flow by accounting for concept level and topic level matching scores. The experiments on heterogeneous datasets indicate that the proposed method outperforms state-of-the-art baselines. We hypothesize that the proposed architecture can also be successfully applied to multi-modal e-commerce search and leave validation of this hypothesis to future work.

REFERENCES

[1] Alexandr Andoni and Ilya Razenshteyn. 2015. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of STOC*. 793–801.

[2] Saeid Balaneshin-kordan and Alexander Kotov. 2017. Embedding-based Query Expansion for Weighted Sequential Dependence Retrieval Model. In *Proceedings of ACM SIGIR*. 1213–1216.

[3] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S Yu. 2016. Deep Visual-Semantic Hashing for Cross-Modal Retrieval. In *Proceedings of ACM SIGKDD*. 1445–1454.

[4] Zhangjie Cao, Mingsheng Long, and Qiang Yang. 2016. Transitive Hashing Network for Heterogeneous Multimedia Retrieval. *arXiv preprint arXiv:1608.04307* (2016).

[5] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of ICIVR*.

[6] Guillem Collell, Ted Zhang, and Marie-Francine Moens. 2017. Imagined Visual Representations as Multimodal Embeddings. In *Proceedings of AAAI*. 4378–4384.

[7] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology* (1990), 391–407.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE CVPR*. 248–255.

[9] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891* (2016).

[10] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, and others. 2015. From captions to visual concepts and back. In *Proceedings of IEEE CVPR*. 1473–1482.

[11] Andrea Frome, Greg S Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A deep visual-semantic embedding model. In *Proceedings of NIPS*. 2121–2129.

[12] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth JF Jones. 2015. Word embedding based generalized language model for information retrieval. In *Proceedings of ACM SIGIR*. 795–798.

[13] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision* (2014), 210–233.

[14] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. 2014. Improving image-sentence embeddings using large weakly annotated photo collections. In *Proceedings of ECCV*. 529–545.

[15] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of ACM CIKM*. 55–64.

[16] Kiana Hajebi, Yasin Abbasi-Yadkori, Hossein Shahbazi, and Hong Zhang. 2011. Fast approximate nearest-neighbor search with k-nearest neighbor graph. In *Artificial Intelligence Journal*. 1312–1317.

[17] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation* (2004), 2639–2664.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of IEEE CVPR*. 770–778.

[19] Felix Hill and Anna Korhonen. 2014. Learning Abstract Concept Embeddings from Multi-Modal Data: Since You Probably Can't See What I Mean. In *Proceedings of EMNLP*. 255–265.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47 (2013), 853–899.

[22] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using click-through data. In *Proceedings of ACM CIKM*. 2333–2338.

[23] Qing-Yuan Jiang and Wu-Jun Li. 2016. Deep Cross-Modal Hashing. *arXiv preprint arXiv:1602.02255* (2016).

[24] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of IEEE CVPR*. 3128–3137.

[25] Andrej Karpathy, Armand Joulin, and Fei Fei Li. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of NIPS*. 1889–1897.

[26] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. 2014. Multimodal neural language models. In *Proceedings of ICML*. 595–603.

[27] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539* (2014).

[28] Alexander Kotov, Vineeth Rakesh, Eugene Agichtein, and Chandan K Reddy. 2015. Geographical Latent Variable Models for Microblog Retrieval. In *Proceedings of ECIR*. 635–647.

[29] Alexander Kotov, Yu Wang, and Eugene Agichtein. 2013. Leveraging geographical metadata to improve search over social media. In *Proceedings of WWW*. 151–152.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of NIPS*. 1097–1105.

[31] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of ACM CIKM*. 1929–1932.

[32] Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining Language and Vision with a Multimodal Skip-gram Model. In *Proceedings of NACL-HLT*. 153–163.

[33] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of IEEE* (1998), 2278–2324.

[34] Mingsheng Long, Yue Cao, Jianmin Wang, and Philip S Yu. 2016. Composite Correlation Quantization for Efficient Multimodal Retrieval. In *Proceedings of ACM SIGIR*. 579–588.

[35] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. 2015. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of IEEE CVPR*. 2623–2631.

[36] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. 2014. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090* (2014).

[37] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*. 3111–3119.

[38] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proceedings of WWW*. 1291–1299.

[39] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*. 1532–1543.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and others. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* (2015), 211–252.

[41] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of ACM SIGIR*. 373–382.

[42] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of ACM CIKM*. 101–110.

[43] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proceedings of NIPS*. 935–943.

[44] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *TACL* (2014), 207–218.

[45] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Proceedings of NIPS*. 2377–2385.

[46] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261* (2016).

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE CVPR*. 2818–2826.

[48] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. LSTM-based deep learning models for non-factoid answer selection. *Proceedings of ICLR* (2016).

[49] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2016. Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016), 652–663.

[50] Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of ACM SIGIR*. 363–372.

[51] Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of ACL*. 707–712.

[52] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of IEEE CVPR*. 5005–5013.

[53] Wei Wang, Xiaoyan Yang, Beng Chin Ooi, Dongxiang Zhang, and Yueting Zhuang. 2016. Effective deep learning-based multi-modal retrieval. *VLDB* (2016), 79–101.

[54] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of ACM SIGIR*. 178–185.

[55] Ying Wei, Yangqiu Song, Yi Zhen, Bo Liu, and Qiang Yang. 2014. Scalable heterogeneous translated hashing. In *Proceedings of ACM SIGKDD*. 791–800.

[56] Yunchao Wei, Yao Zhao, Canyi Lu, Shikui Wei, Luoqi Liu, Zhenfeng Zhu, and Shuicheng Yan. 2016. Cross-modal retrieval with cnn visual features: A new baseline. *IEEE Transactions on Cybernetics* 47, 2 (2016), 449–460.

[57] Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning* 81, 1 (2010), 21–35.

[58] Fei Wu, Zhou Yu, Yi Yang, Siliang Tang, Yin Zhang, and Yueting Zhuang. 2014. Sparse multi-modal hashing. *Proceedings of MM* (2014), 427–439.

[59] Hamed Zamani and W Bruce Croft. 2017. Relevance-based Word Embedding. In *Proceedings of ACM SIGIR*. 505–514.