Learning to Rank Images with Cross-Modal Graph Convolutions

Thibault Formal¹, Stéphane Clinchant^{1(✉)}, Jean-Michel Renders¹,
Sooyeol Lee², and Geun Hee Cho²

¹ Naver Labs Europe, Meylan, France

{`thibault.formal`,`stephane.clinchant`,`jean-michel.renders`}@naverlabs.com

² Naver Corporation - Search Intelligence Team - Image & Video,
Seongnam-si, South Korea

{`sooyeol.lee`,`geunhee.cho`}@navercorp.com

Abstract. We are interested in the problem of cross-modal retrieval for web image search, where the goal is to retrieve images relevant to a text query. While most of the current approaches for cross-modal retrieval revolve around learning how to represent text and images in a shared latent space, we take a different direction: we propose to generalize the cross-modal relevance feedback mechanism, a simple yet effective unsupervised method, that relies on standard information retrieval heuristics and the choice of a few hyper-parameters. We show that we can cast it as a supervised representation learning problem on graphs, using graph convolutions operating jointly over text and image features, namely cross-modal graph convolutions. The proposed architecture directly learns how to combine image and text features for the ranking task, while taking into account the context given by all the other elements in the set of images to be (re-)ranked. We validate our approach on two datasets: a public dataset from a **MediaEval** challenge, and a small sample of proprietary image search query logs, referred as **WebQ**. Our experiments demonstrate that our model improves over standard baselines.

Keywords: Cross-modal retrieval · Learning to rank · Graph convolutions

1 Introduction

This paper considers the typical image search scenario, where a user enters a text query, and the system returns a set of ranked images. More specifically, we are interested in re-ranking a subset of candidate images retrieved from the whole image collection by an efficient base ranker, following standard multi-stage ranking architectures in search engines [36]. Directly including visual features in the ranking process is actually not straightforward due to the semantic gap between text and images: this is why the problem has initially been addressed using standard text-based retrieval, relying for instance on text crawled from the image's

webpage (e.g. surrounding text, title of the page etc.). In order to exploit visual information, and therefore improve the quality of the results –especially because this text is generally noisy, and hardly describes the image semantic–, many techniques have been developed since. For instance, some works have focused on building similarity measures by fusing mono-modal similarities, using either simple combination rules, or more complex propagation mechanisms in similarity graphs. More recently, techniques have emerged from the computer vision community, where text and images are embedded in the same latent space (a.k.a. joint embedding), allowing to directly match text queries to images. The latter are currently considered as state-of-the-art techniques for the cross-modal retrieval task. However, they are generally evaluated on artificial retrieval scenarios (e.g. on MSCOCO dataset [34]), and rarely considered in a re-ranking scenario, where mechanisms like pseudo-relevance feedback (PRF) [31] are highly effective.

We propose to revisit the problem of cross-modal retrieval in the context of re-ranking. Our first contribution is to derive a general formulation of a differentiable architecture, drawing inspiration from cross-modal retrieval, learning to rank, neural information retrieval and graph neural networks. Compared to joint embedding approaches, we tackle the problem in a different view: instead of learning new (joint) embeddings, we focus on designing a model that *learns to combine* information from different modalities. Finally, we validate our approach on two datasets, using simple instances of our general formulation, and show that the approach is not only able to reproduce PRF, but actually outperform it.

2 Related Work

Cross-Modal Retrieval. In the literature, two main lines of work can be distinguished regarding cross-modal retrieval: the first one focuses on designing effective cross-modal similarity measures (e.g. [2, 10]), while the second seeks to learn how to map images and text into a shared latent space (e.g. [15, 18, 19, 54]).

The first set of approaches simply combines different mono-media similarity signals, relying either on simple aggregation rules, or on unsupervised cross-modal PRF mechanisms, that depend on the choice of a few but critical hyper-parameters [2, 10, 11, 45]. As it will be discussed in the next section, the latter can be formulated as a two-step PRF propagation process in a graph, where nodes represent multi-modal objects and edges encode their visual similarities. It has been later extended to more general propagation processes based on random walks [28].

Alternatively, joint embedding techniques aim at learning a mapping between textual and visual representations [15, 18, 19, 23, 52–55, 61]. Canonical Correlation Analysis (CCA) [17] and its deep variants [5, 27, 58], as well as bi-directional ranking losses [8, 9, 52, 53, 55, 61] (or triplet losses) ensure that, in the new latent space, an image and its corresponding text are correlated or close enough w.r.t. to the other images and pieces of text in the training collection. Other objective functions utilize metric learning losses [35], machine translation-based measures [44] or even adversarial losses [51].

These approaches suffer from several limitations [61]: they are sensitive to the triplet sampling strategy as well as the choice of appropriate margins in the ranking losses. Moreover, constituting a training set that ensures good learning and generalization is not an easy task: the text associated to an image should describe its visual content (e.g. *“a man speaking in front of a camera in a park”*), and nothing else (e.g. *“the President of the US, the 10th of March”*, *“John Doe”*, *“joy and happiness”*).

Building a universal training collection of paired (image, text) instances, where text describes faithfully the content of the image in terms of elementary objects and their relationships, would be too expensive and time-consuming in practice. Consequently, image search engines rely on such pairs crawled from the Web, where the link between image and text (e.g. image caption, surrounding sentences etc.) is tenuous and noisy.

To circumvent this problem, query logs could be used but, unfortunately –and this is our second argument regarding the limitations–, real queries are never expressed in the same way as the ones considered when evaluating joint embedding methods (e.g. artificial retrieval setting on MSCOCO [34] or Flickr-30K [43] datasets, where the query is the full canonical textual description of the image). In practice, queries are characterised by very large intent gaps: they do not really describe the content of the image but, most of the time, contain only a few words, and are far from expressing the true visual needs. What does it mean to impose close representations for all images representing *“Paris”* (e.g. *“the Eiffel Tower”*, *“Louvre Museum”*), even if they can be associated to the same textual unit?

Neural Information Retrieval. Neural networks, such as RankNet and LambdaRank, have been intensively used in IR to address the learning to rank task [7]. More recently, there has been a growing interest in designing effective IR models with neural models [1, 12, 13, 20, 25, 26, 37, 38, 41, 56], by learning the features useful for the ranking task directly from text.

While standard strategies focus on learning a global ranking function that considers each query-document pair in isolation, they tend to ignore the difference in distribution in the feature space for different queries [4]. Hence, some recent works have been focusing on designing models that exploit the context induced by the re-ranking paradigm, either by explicitly designing differentiable PRF models [32, 40], or by encoding the ranking context –the set of elements to re-rank–, using either RNNs [4] or attention mechanisms [42, 62]. Consequently, the score for a document takes into account all the other documents in the candidate list. Because of their resemblance with structured problems, this type of approaches could benefit from the recent body of work around graph neural networks, which operate on graphs by learning how to propagate information to neighboring nodes.

Graph Neural Networks. Graph Neural Networks (GNNs) are extensions of neural networks that deal with structured data encoded as a graph. Recently,

Graph Convolutional Networks (GCNs) [30] have been proposed for semi-supervised classification of nodes in a graph. Each layer of a GCN can generally be decomposed as: (i) node features are first transformed (e.g. linear mapping), (ii) node features are convolved, meaning that for each node, a differentiable, permutation-invariant operation (e.g. sum, mean, or max) of its neighbouring node features is computed, before applying some non-linearity, (iii) finally, we obtain a new representation for each node in the graph, which is then fed to the next layer. Many extensions of GCNs have been proposed (e.g. GraphSAGE [21], Graph Attention Network [50], Graph Isomorphism Network [57]), some of them directly tackling the recommendation task (e.g. PinSAGE [59]). But to the best of our knowledge, there is no prior work on using graph convolutions for the (re-)ranking task.

3 Learning to Rank Images

3.1 Cross-Modal Similarity Measure

Our goal is to extend and generalize simple yet effective unsupervised approaches which have been proposed for the task [2, 3, 10, 11, 45], that can be seen as an extension of pseudo-relevance feedback methods for multi-modal objects. Let $d \in D$ denote a document to re-rank, composed of text and image. We denote by $s_V(.,.)$ a normalized similarity measure between two images, and by $s_T(q, d)$ the textual relevance score of document d w.r.t. query q . The cross-modal similarity score is given by:

$$\forall d \in D, s_{CM}(q, d) = \sum_{d_i \in NN_T^K(q)} s_T(q, d_i) s_V(d_i, d) \quad (1)$$

where $NN_T^K(q)$ denotes the set of K most relevant documents w.r.t. q , based on text, i.e. on $s_T(q, .)$. The model can be understood very simply: similarly to PRF methods in standard information retrieval, the goal is to boost images that are visually similar to *top* images (from a text point of view), i.e. images that are likely to be relevant to the query but were initially badly ranked (which is likely to happen in the web scenario, where text is crawled from source page and can be very noisy).

Despite showing good empirical results, cross-modal similarities are fully unsupervised, and lack some dynamic behaviour, like being able to adapt to different queries. Moreover, they rely on a single relevance score $s_T(q, .)$, while it could actually be beneficial to learn how to use a larger set of features such as the ones employed in learning to rank models.

3.2 Cross-Modal Graph Convolution

In [3], the authors made a parallel between the cross-modal similarity from Eq. (1) and random walks in graphs: it can be seen as a kind of multimodal

label propagation in a graph. This motivates us to tackle the task using graph convolutions. We therefore represent each query $q \in \mathcal{Q}$ as a graph $\mathcal{G}_q$, as follows:

- The set of nodes is the set of candidate documents d_i to be re-ranked for this query: typically from a few to hundreds of documents, depending on the query.
- Each node i is described by a set of n learning to rank features $x_{q,d_i} \in \mathbb{R}^n$.
- $v_i \in \mathbb{R}^d$ denotes the (normalized) visual embedding for document d_i .
- As we do not have an explicit graph structure, we consider edges given by a k -nearest neighbor graph, based on a similarity between the embeddings v_i ¹.
- We denote by $\mathcal{N}_i$ the neighborhood of node i , i.e. the set of nodes j such that there exists an edge from j to i .
- We consider edge weights, given by a similarity function between the visual features of its two extremity nodes $f_{ij} = \mathbf{g}(v_i, v_j)$.

Our goal is to learn how to propagate features in the above graph. Generalizing convolution operations to graphs can generally be expressed as a message passing scheme [16]:

$$h_i^{(l+1)} = \gamma(h_i^{(l)}, \sum_{j \in \mathcal{N}_i} \phi(h_i^{(l)}, h_j^{(l)}, f_{ij})) \quad (2)$$

where γ and ϕ denote differentiable functions, e.g. MLPs (Multi Layer Perceptron). By choosing $\phi(h_i^{(l)}, h_j^{(l)}, f_{ij}) = \tau(h_i^{(l)})\mathbf{g}(v_i, v_j)$ and $\gamma(x, y) = y$, Eq. (2) reduces to:

$$h_i^{(l+1)} = \sum_{j \in \mathcal{N}_i} \tau(h_j^{(l)})\mathbf{g}(v_i, v_j) \quad (3)$$

This graph convolution can be reduced to the cross-modal similarity in Eq. (1). Indeed, assuming that $h_i := s_T(q, d_i)$, $\tau(\cdot)$ is a top- k filtering function, $\mathbf{g}(v_i, v_j) := s_V(d_i, d)$, and $\mathcal{N}_i := \mathcal{N}$ is the whole set of candidates to re-rank, then:

$$\sum_{j \in \mathcal{N}} \tau_k(s_T(q, d_i))\mathbf{g}(v_i, v_j) = \sum_{d_i \in NN_T^K(q)} s_T(q, d_i)s_V(d_i, d) \quad (4)$$

In other words, one layer defined with Eq. (3) includes the standard cross-modal relevance feedback as a special case. Equation (3) is more general, and can easily be used as a building block in a differentiable ranking architecture.

3.3 Learning to Rank with Cross-Modal Graph Convolutions

In the following, we derive a simple convolution layer from Eq. (3), and we introduce the complete architecture –called DCMM for Differentiable Cross-Modal

¹ With the special case of considering that all the nodes are connected to each other, i.e. $k = |\mathcal{G}_q| - 1$.

Model—, summarized in Fig. 1. Learning to rank features x_{q,d_i} are first encoded with an $\text{MLP}(\cdot; \theta)$ with ReLU activations, in order to obtain node features h_i^0 . Then, the network splits into two branches:

- The first branch simply projects linearly each $h_i^{(0)}$ to a real-valued score $s_T(q, d_i) = \mathbf{w}_0^T h_i^{(0)}$, that acts as a pure text-based score².
- The second branch is built upon one or several layer(s) of cross-modal convolution, simply defined as:

$$h_i^{(l+1)} = \text{ReLU}\left(\sum_{j \in \mathcal{N}_i} \mathbf{W}^{(l)} h_j^{(l)} \mathbf{g}(v_i, v_j)\right) \quad (5)$$

For the edge function $\mathbf{g}$, we consider two cases: the cosine similarity $g_{\cos}(v_i, v_j) = \cos(v_i, v_j)$, defining the first model (referred as **DCMM-cos**), and a simple learned similarity measure parametrized by a vector $\mathbf{a}$ such that $\mathbf{g}_{\text{edge}}(v_i, v_j) = v_i^T \text{diag}(\mathbf{a}) v_j$, defining our second model (referred as **DCMM-edge**).

After the convolution(s), the final embedding for each node $h_i^{(L)}$ is projected to a real-valued score $s_{\text{conv}}(q, d_i)$, using either a linear layer ($s_{\text{conv}}(q, d_i) = \mathbf{w}_L^T h_i^{(L)}$) or an MLP ($s_{\text{conv}}(q, d_i) = \text{MLP}(h_i^{(L)}, \omega)$). Finally, the two scores are combined to obtain the final ranking score:

$$s(q, d_i) = \mathbf{w}_0^T h_i^{(0)} + s_{\text{conv}}(q, d_i)$$

Fig. 1. Differentiable Cross-Modal Model architecture - schematic view. Note that the upper part is a standard MLP-based LTR model.

The model is trained using backpropagation and any standard learning to rank loss: pointwise, pairwise or listwise. It is worth to remark that, by extending PRF mechanisms for cross-modal re-ranking, our model is actually closer to listwise context-based models introduced in Sect. 2 than current state-of-the-art

² In addition to improve the results, keeping a separate branch focusing on learning to rank images solely from input nodes (i.e. learning to rank features) actually stabilizes the training, thanks to the shared input transformation.

cross-modal retrieval models. It is listwise by design³: an example in a batch is not a single image in isolation, but all the candidate images for a given query, encoded as a graph, that we aim to re-rank together in a one shot manner. In our experiments, we used the pairwise BPR loss [46], from which we obtained the best results⁴. Let's consider a graph (i.e. the set of candidate documents for query q) in the batch, and all the feasible pairs of documents $D_q^{+,-}$ for this query (by feasible, we mean all the pairs that can be made from positive and negative examples in the graph). Then the loss is defined:

$$\mathcal{L}(\theta, w_0, \gamma_{\text{conv}}, \omega) = - \sum_{d^+, d^- \in D_q^{+,-}} \log \sigma(s(q, d^+) - s(q, d^-))$$

Note that contrary to previous works on listwise context modeling, we consider a *set* of objects to re-rank, and not a sequence (for instance in [4], a RNN encoder is learned for re-ranking). In other words, we discard the rank information of the first ranker into the re-ranking process: we claim that the role of the first retriever is to be recall-oriented, and not precision-oriented. Thus, using initial order might be a too strong prior, and add noise information. Moreover, in the case of implicit feedback (clicks used as weak relevance signals), using rank information raises the issue of biased learning to rank (sensitivity to position and trust biases). It is also worth to emphasize that, contrary to most of the works around graph convolution models, our graph structure is somehow implicit: while edges between nodes generally indicate a certain relationship between nodes (for instance, connection between two users in a social network), in our case a connection represents the visual similarity between two nodes.

4 Experiments

In the following, we introduce the two datasets we used to validate our approach—a public dataset from a MediaEval⁵ challenge, and an annotated set of queries sampled from image search logs of Naver, the biggest commercial search engine in Korea—, as well as our experimental strategy. We emphasize on the fact that we restrict ourselves to two relatively small datasets and few features as input for the models. Even though the formulation from Eq. (3) is very general, our claim is that a simple model, i.e. containing few hundreds to thousands parameters, should be able to reproduce PRF mechanisms introduced in Sect. 3. When adapting the approach to larger datasets, the model capacity can be adjusted accordingly, in order to capture more complex relevance patterns. Note that

³ Note the difference between a model that has been trained using a listwise loss function but uses a pointwise scoring function (i.e. the score depends only on the document itself), and a model that directly uses a listwise scoring function.

⁴ While it seems appealing to directly use listwise losses, we actually obtained slightly worse performance when experimenting with the AttRank loss from [4] and the ApproxNDCG loss from [6].

⁵ <http://www.multimediaeval.org/mediaeval2019/>.

we did not consider in our study standard datasets generally used to train joint embeddings such as MSCOCO [34] or Flickr30k [43], because the retrieval scenario is rather artificial, compared to web search: there are no explicit queries, and a text is only relevant to a single image. Furthermore, we have tried to obtain the Clickture [24] dataset without success⁶, and therefore cannot report on it.

4.1 Datasets

MediaEval. We first conduct experiments on the dataset from the “MediaEval17, Retrieving Diverse Social Images Task” challenge⁷. While this challenge also had a focus on diversity aspects, we solely consider the standard relevance ranking task. The dataset is composed of a ranked list of images (up to 300) for each query, retrieved from Flickr using its default ranking algorithm. The queries are general-purpose queries (e.g. $q = \textit{autumn color}$), and each image has been annotated by expert annotators (binary label, i.e. relevant or not). The goal is to refine the results from the base ranking. The training set contains 110 queries for 33340 images, while the test set contains 84 queries for 24986 images.

While we could consider any number of learning to rank features as input for our model, we choose to restrict ourselves to a very narrow set of weak relevance signals, in order to remain comparable to its unsupervised counterpart, and ensure that the gain does not come from the addition of richer features. Hence, we solely rely on four relevance scores, namely tf-idf, BM25, Dirichlet smoothed LM [60] and DESM score [39], between the query and each image’s text component (the concatenation of the image title and tags). We use an Inception-ResNet model [48] pre-trained on ImageNet to get the image embeddings ($d = 1536$).

WebQ. In order to validate our approach on a real world dataset, we sample a set of 1000 queries⁸ from the image search logs of Naver. All images appearing in the top-50 candidates for these queries within a period of time of two weeks have been labeled by three annotators in terms of relevance to the query (binary label). Because of different query characteristics (in terms of frequency, difficulty etc.), and given the fact that new images are continuously added to/removed from the index, the number of images per query in our sample is variable (from around ten to few hundreds). Note that, while we actually have access to a much larger amount of click logs, we choose to restrict the experiments to this small sample in order keep the evaluations simple. Our goal here is to show that we are able to learn and reproduce some PRF mechanisms, without relying on large amount of data. Moreover, in this setting, it is easier to understand model’s behaviour, as we avoid to deal with click noise and position bias. After removing queries without relevant images (according to majority voting among the three annotators), our sample includes 952 queries, and 43064 images, indexed through

⁶ As of today, the data link seems broken and we got no response from the person in charge of this dataset.

⁷ <http://www.multimediaeval.org/mediaeval2017/diverseimages/>.

⁸ Our sample includes head, torso and tail queries.

various text fields (title of the page, image caption etc.). We select seven of such fields, that might contain relevant pieces of information, and for which we compute two simple relevance features w.r.t. query q : BM25 and DESM [39] (using embeddings trained on a large query corpus from an anterior period). We also add an additional feature, which is a mixture of the two above, on the concatenation of all the fields. Image embeddings ($d = 2048$) are obtained using a ResNet-152 model [22] pre-trained on ImageNet.

4.2 Evaluation Methodology

Given the limited number of queries in both collections, we conducted 5-fold cross-validation, by randomly splitting the queries into five folds. The model is trained on 4 folds (with 1 fold kept for validation, as we use early stopping on nDCG), and evaluated on the remaining one; this procedure is repeated 5 times. Then, the average validation nDCG is used to select the best model configuration. Note that for the **MediaEval** dataset, we have access to a separate test set, so we modify slightly the evaluation methodology: we do the above 5-fold cross-validation on the training set, without using a validation fold (hence, we do not use early stopping, and the number of epochs is a hyperparameter to tune). Once the best model has been selected with the above strategy, we re-train it on the full training set, and give the final performance on the test set. We report the nDCG, MAP, P@20, and nDCG@20 for both datasets.

We train the models using stochastic gradient descent with the Adam optimizer [29]. We set the batch size (i.e. number of graphs per batch) to $bs = \{5, \{32\}$ for respectively **MediaEval** and **WebQ**, so that training fits on a single NVIDIA Tesla P100 GPU. The hyper-parameters we tune for each dataset are: (1) the learning rate $\in \{1e-3, 1e-4, 5e-5\}$, (2) the number of layers $\in \{2, 3\}$ for the input MLP, as well as the number of hidden units $\in \{4, 8, 16, 32\}, \{8, 16, 32, 64\}$, (2) the dropout rate [47] in the MLP layers $\in \{0, 0.2\}$, (4) the number of graph convolutions $\in \{1, 2, 3, 4\}$ as well as the number of hidden units $\in \{4, 8, 16\}, \{8, 16, 32\}$, (5) the dropout rate of the convolution layers $\in \{0, 0.2, 0.5\}$ and (6) the number of visual neighbors to consider when building the input graph, $\in \{1, 3, 5, 10, 20, 50, 80, 100, 120, |\mathcal{G}| - 1\}$, $\{1, 3, 5, 10, 15, 20, 30, |\mathcal{G}| - 1\}$ for respectively **MediaEval** and **WebQ**. For **MediaEval**, we also tune the number of epochs $\in \{50, 100, 200, 300, 500\}$, while for **WebQ**, we set it to 500, and use early stopping with patience set to 80. All node features are query-level normalized (mean-std normalization). The models are implemented using PyTorch and PyTorch geometric⁹ [14] for the message passing components.

4.3 Baselines

In order to be fair, we want to compare methods with somewhat similar feature sets. Obviously, for the supervised methods, results can be improved by either

⁹ https://github.com/rusty1s/pytorch_geometric.

adding richer/more features, or increasing models’ capacity. For both datasets, we compare our **DCMM** model to the following baselines:

- A learning to rank model only based on textual features (**LTR**).
- The cross-modal similarity introduced in Sect. 3.1 [2, 3, 10, 11, 45] (**CM**).
- The above LTR model with the cross-modal similarity as additional input feature (**LTR+CM**), to verify that it is actually beneficial to learn the cross-modal propagation in DCMM in a end-to-end manner.

For the cross-modal similarity, we use as proxy for $s_T(q, \cdot)$ a simple mixture of term-based relevance score (Dirichlet-smoothed LM and BM25 for respectively **MediaEval** and **WebQ**) and DESM score, on a concatenation of all text fields. From our experiments, we observe that it is actually beneficial to recombine the cross-modal similarity with the initial relevance $s_T(q, \cdot)$, using a simple mixture. Hence, three parameters are tuned (the two mixture parameters, and the number of neighbors for the query), following the evaluation methodology introduced in Sect. 4.2¹⁰. The LTR models are standard MLPs: they correspond to the upper part of architecture Fig. 1 (text branch), and are tuned following the same strategy.

We do not compare our models with joint embedding approaches on those datasets for the reasons mentioned in Sect. 2, but also due to our initial experiments on **Medieval** which gave poor results. For the sake of illustration, on **MediaEval**, 64% of the queries have no lemmas in common with training queries (and 35% for **WebQ**): given the relatively small size of these datasets, the models cannot generalize to unseen queries. This illustrates an “extreme” example of the generalization issues –especially on tail queries– of joint embedding techniques. In the meantime, as our model is fed with learning to rank features, especially term-based relevance scores like BM25, it could be less sensitive to generalization issues, for instance on new named entities. However, we want to emphasize that both approaches are not antagonist, but can actually be complementary. As our model can be seen as an extension of listwise learning to rank for bi-modal objects (if edges are removed, the model reduces to a standard MLP-based learning to rank), it can take as input node features matching scores from joint embeddings models. The model being an extension of PRF, we actually see the approaches at different stages of ranking.

4.4 Results and Analysis

Table 1 gathers the main results of our study. Without too much surprise, going from pure text ranker to a model using both media types improves the results by a large margin (all the models are significantly better than the text-based LTR model, so we do not include these tests on Table 1 for clarity). Moreover, results

¹⁰ Note that, when used as a LTR feature, we obviously do not recombine the CM score with initial relevance score, as it will be redundant with other text features. Hence, we directly use score from Eq. (1), and tune only two parameters.

Table 1. Comparison of the methods on both datasets (test metrics). Significant improvement w.r.t. the cross-modal similarity (CM sim) is indicated with * (p -value < 0.05). The number of trained parameters are indicated for the convolution models: ranging from few hundreds to few thousands, i.e. orders of magnitude less than joint embeddings models.

		# params	P@20	nDCG@20	nDCG	MAP
MediaEval'17	LTR		0.758	0.767	0.912	0.707
	CM [11]		0.843	0.857	0.939	0.784
	LTR + CM		0.852	0.868	0.942	0.789
	DCMM-cos	268	0.871	0.876	0.944	0.803
	DCMM-edge	3314	0.861	0.871	0.944	0.806*
WebQ	LTR		0.69	0.801	0.884	0.775
	CM [11]		0.724	0.84	0.901	0.815
	LTR + CM		0.724	0.839	0.901	0.813
	DCMM-cos	1868	0.729	0.847	0.905	0.821
	DCMM-edge	15522	0.738*	0.857*	0.91*	0.83*

indicate that combining initial features with the unsupervised cross-modal similarity in a LTR model allows to slightly improve results over the latter (not significantly though) for the **MediaEval** dataset, while it has no effect on **WebQ**: this is likely due to the fact that features are somehow redundant in our setting, because of how $s_T(q, \cdot)$ is computed for the cross-modal similarity; the same would not hold if we would consider a richer set of features for the LTR models. Furthermore, the DCMM-cos model outperforms all the baselines, with larger margins for **MediaEval** than for **WebQ**; the only significant result (p -value < 0.05) is obtained for the MAP on **MediaEval**. Nevertheless, it shows that this simple architecture –the most straightforward extension of cross-modal similarity introduced in Sect. 3.1–, with a handful of parameters (see Table 1) and trained on small datasets, is able to reproduce PRF mechanisms. Interestingly, results tend to drop as we increase the number of layers (best results are obtained with a single convolution layer), no matter the number of neighbors chosen to define the visual graph. While it might be related to the relative simplicity of the model, it actually echoes common observations in PRF models (e.g. [3]): if we propagate too much, we also tend to diffuse information too much. Similarly, we can also make a parallel with over-smoothing in GNNs [33], which might be more critical for PRF, especially considering the simplicity of this model.

The DCMM-edge shows interesting results: on **WebQ**, we manage to improve results significantly w.r.t. to CM sim, while on **MediaEval**, results are slightly worse than DCMM-cos (except for the MAP). It might be due to the fact that images in the latter are more alike to the ones used to train image signatures, compared to the (noisy) web images in **WebQ**; hence, learning a new metric between images has less impact. Interestingly, for both datasets, best results

are obtained with more than a single layer; we hypothesize that the edge function plays the role of a simple filter for edges, allowing to propagate information from useful nodes across more layers. Note that the number of layers needed for the task is tied with how we define our input graph: the less neighbors we consider for each node, the more layers might be needed, in order for each node to gather information from useful nodes. In Fig. 2, we observe that if the number of neighbors is too small (e.g. 3 or 5), then the model needs more layers to improve performance. On the other side, when considering too many neighbors (e.g. 20 or all), the nodes already have access to all the useful neighbors, hence adding layers only reduces performances. We need to find the right balance between the number of neighbors and the number of convolution layers, so that the model can learn to propagate relevant signals (e.g. 10 neighbors and 3 layers for WebQ).

Fig. 2. Impact of the number of convolutions layers and top-k neighbors for WebQ.

5 Conclusion

In this paper, we have proposed a reformulation of unsupervised cross-modal PRF mechanisms for image search as a differentiable architecture relying on graph convolutions. Compared to its unsupervised counterpart, our novel approach can integrate any set of features, while providing a high flexibility in the design of the architecture. Experiments on two datasets showed that a simple model derived from our formulation achieved comparable –or better– performance compared to cross-modal PRF.

There are many extensions and possible directions stemming from the relatively simple model we have studied. Given enough training data (e.g. large amount of click logs), we could for instance learn to dynamically filter the visual similarity by using an attention mechanism to choose which nodes to attend,

similarly to Graph Attention Networks [50] and Transformer model [49], discarding the need to set the number of neighbors in the input graph. Finally, our approach directly addressed the cross-modal retrieval task, but its application to the more general PRF problem in IR remains possible.

References

1. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In: ACM International Conference on Information and Knowledge Management (CIKM), October 2013. <https://www.microsoft.com/en-us/research/publication/learning-deep-structured-semantic-models-for-web-search-using-clickthrough-data/>
2. Ah-Pine, J.M., Cifarelli, C.M., Clinchant, S.M., Csurka, G.M., Renders, J.M.: XRCE's Participation to ImageCLEF 2008. In: 9th Workshop of the Cross-Language Evaluation Forum (CLEF 2008), Aarhus, Denmark, September 2008. <https://hal.archives-ouvertes.fr/hal-01504444>
3. Ah-Pine, J., Csurka, G., Clinchant, S.: Unsupervised visual and textual information fusion in CBMIR using graph-based methods. *ACM Trans. Inf. Syst.* **33**(2), 9:1–9:31 (2015). <https://doi.org/10.1145/2699668>
4. Ai, Q., Bi, K., Guo, J., Croft, W.B.: Learning a deep listwise context model for ranking refinement. *CoRR* abs/1804.05936 (2018). <http://arxiv.org/abs/1804.05936>
5. Andrew, G., Arora, R., Bilmes, J., Livescu, K.: Deep canonical correlation analysis. In: International Conference on Machine Learning, pp. 1247–1255 (2013)
6. Bruch, S., Zoghi, M., Bendersky, M., Najork, M.: Revisiting approximate metric optimization in the age of deep neural networks. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), pp. 1241–1244 (2019)
7. Burges, C.J.: From ranknet to lambdarank to lambdamart: an overview. Technical report, June 2010. <https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/>
8. Carvalho, M., Cadène, R., Picard, D., Soulier, L., Thome, N., Cord, M.: Cross-modal retrieval in the cooking context: learning semantic text-image embeddings. *CoRR* abs/1804.11146 (2018). <http://arxiv.org/abs/1804.11146>
9. Chen, K., Bui, T., Chen, F., Wang, Z., Nevatia, R.: AMC: attention guided multi-modal correlation learning for image search. *CoRR* abs/1704.00763 (2017). <http://arxiv.org/abs/1704.00763>
10. Clinchant, S., Renders, J.-M., Csurka, G.: Trans-media pseudo-relevance feedback methods in multimedia retrieval. In: Peters, C., et al. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 569–576. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85760-0_71
11. Csurka, G., Ah-Pine, J., Clinchant, S.: Unsupervised visual and textual information fusion in multimedia retrieval - a graph-based point of view. *CoRR* abs/1401.6891 (2014). <http://arxiv.org/abs/1401.6891>
12. Dai, Z., Xiong, C., Callan, J., Liu, Z.: Convolutional neural networks for soft-matching n-grams in ad-hoc search. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. WSDM 2018, pp. 126–134. ACM, New York (2018). <https://doi.org/10.1145/3159652.3159659>

13. Fan, Y., Guo, J., Lan, Y., Xu, J., Zhai, C., Cheng, X.: Modeling diverse relevance patterns in ad-hoc retrieval. CoRR abs/1805.05737 (2018). <http://arxiv.org/abs/1805.05737>
14. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
15. Frome, A., et al.: DeViSE: a deep visual-semantic embedding model (2013)
16. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. CoRR abs/1704.01212 (2017). <http://arxiv.org/abs/1704.01212>
17. Gong, Y., Ke, Q., Isard, M., Lazebnik, S.: A Multi-View Embedding Space for Modeling Internet Images, Tags, and Their Semantics. Int. J. Comput. Vis. **106**(2), 210–233 (2013). <https://doi.org/10.1007/s11263-013-0658-4>
18. Gong, Y., Wang, L., Hodosh, M., Hockenmaier, J., Lazebnik, S.: Improving image-sentence embeddings using large weakly annotated photo collections. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 529–545. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_35
19. Gordo, A., Larlus, D.: Beyond instance-level image retrieval: leveraging captions to learn a global visual representation for semantic retrieval (2017)
20. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. CoRR abs/1711.08611 (2017). <http://arxiv.org/abs/1711.08611>
21. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. CoRR abs/1706.02216 (2017). <http://arxiv.org/abs/1706.02216>
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). <http://arxiv.org/abs/1512.03385>
23. Hu, P., Zhen, L., Peng, D., Liu, P.: Scalable deep multimodal learning for cross-modal retrieval. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 635–644 (2019)
24. Hua, X.S., et al.: Clickage: towards bridging semantic and intent gaps via mining click logs of search engines, pp. 243–252, October 2013. <https://doi.org/10.1145/2502081.2502283>
25. Hui, K., Yates, A., Berberich, K., de Melo, G.: A position-aware deep model for relevance matching in information retrieval. CoRR abs/1704.03940 (2017). <http://arxiv.org/abs/1704.03940>
26. Hui, K., Yates, A., Berberich, K., de Melo, G.: RE-PACRR: a context and density-aware neural information retrieval model. CoRR abs/1706.10192 (2017). <http://arxiv.org/abs/1706.10192>
27. Kan, M., Shan, S., Chen, X.: Multi-view deep network for cross-view classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4847–4855 (2016)
28. Khasanova, R., Dong, X., Frossard, P.: Multi-modal image retrieval with random walk on multi-layer graphs. In: 2016 IEEE International Symposium on Multimedia (ISM), pp. 1–6 (2016)
29. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014). <http://arxiv.org/abs/1412.6980>
30. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. CoRR abs/1609.02907 (2016). <http://arxiv.org/abs/1609.02907>
31. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 2001, pp. 120–127. ACM, New York (2001). <https://doi.org/10.1145/383952.383972>

32. Li, C., et al.: NPRF: a neural pseudo relevance feedback framework for ad-hoc information retrieval. CoRR abs/1810.12936 (2018). <http://arxiv.org/abs/1810.12936>
33. Li, Q., Han, Z., Wu, X.: Deeper insights into graph convolutional networks for semi-supervised learning. CoRR abs/1801.07606 (2018). <http://arxiv.org/abs/1801.07606>
34. Lin, T., et al.: Microsoft COCO: common objects in context. CoRR abs/1405.0312 (2014). <http://arxiv.org/abs/1405.0312>
35. Liong, V.E., Lu, J., Tan, Y.P., Zhou, J.: Deep coupled metric learning for cross-modal matching. IEEE Trans. Multimedia **19**(6), 1234–1244 (2016)
36. Liu, S., Xiao, F., Ou, W., Si, L.: Cascade ranking for operational e-commerce search. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 2017 (2017). <https://doi.org/10.1145/3097983.3098011>
37. Mitra, B., Craswell, N.: An updated duet model for passage re-ranking. CoRR abs/1903.07666 (2019). <http://arxiv.org/abs/1903.07666>
38. Mitra, B., Diaz, F., Craswell, N.: Learning to match using local and distributed representations of text for web search. CoRR abs/1610.08136 (2016). <http://arxiv.org/abs/1610.08136>
39. Mitra, B., Nalisnick, E.T., Craswell, N., Caruana, R.: A dual embedding space model for document ranking. CoRR abs/1602.01137 (2016). <http://arxiv.org/abs/1602.01137>
40. Nogueira, R., Cho, K.: Task-oriented query reformulation with reinforcement learning. CoRR abs/1704.04572 (2017). <http://arxiv.org/abs/1704.04572>
41. Pang, L., Lan, Y., Guo, J., Xu, J., Xu, J., Cheng, X.: DeepRank: a new deep architecture for relevance ranking in information retrieval. CoRR abs/1710.05649 (2017). <http://arxiv.org/abs/1710.05649>
42. Pei, C., et al.: Personalized context-aware re-ranking for e-commerce recommender systems. CoRR abs/1904.06813 (2019). <http://arxiv.org/abs/1904.06813>
43. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. CoRR abs/1505.04870 (2015). <http://arxiv.org/abs/1505.04870>
44. Qi, J., Peng, Y.: Cross-modal bidirectional translation via reinforcement learning. In: IJCAI, pp. 2630–2636 (2018)
45. Renders, J.M., Csurka, G.: NLE@MediaEval’17: combining cross-media similarity and embeddings for retrieving diverse social images. In: MediaEval (2017)
46. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. UAI 2009, pp. 452–461. AUAI Press, Arlington (2009). <http://dl.acm.org/citation.cfm?id=1795114.1795167>
47. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014). <http://jmlr.org/papers/v15/srivastava14a.html>
48. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-ResNet and the impact of residual connections on learning. CoRR abs/1602.07261 (2016). <http://arxiv.org/abs/1602.07261>
49. Vaswani, A., et al.: Attention is all you need. CoRR abs/1706.03762 (2017). <http://arxiv.org/abs/1706.03762>
50. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018). <https://openreview.net/forum?id=rJXMpikCZ>

51. Wang, B., Yang, Y., Xu, X., Hanjalic, A., Shen, H.T.: Adversarial cross-modal retrieval. In: Proceedings of the 25th ACM International Conference on Multimedia. MM 2017, pp. 154–162 (2017)
52. Wang, L., Li, Y., Huang, J., Lazebnik, S.: Learning two-branch neural networks for image-text matching tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(2), 394–407 (2018)
53. Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5005–5013 (2016)
54. Weston, J., Bengio, S., Usunier, N.: WSABIE: scaling up to large vocabulary image annotation (2011)
55. Wu, Y., Wang, S., Huang, Q.: Learning semantic structure-preserved embeddings for cross-modal retrieval. In: Proceedings of the 26th ACM International Conference on Multimedia. MM 2018, pp. 825–833. ACM (2018)
56. Xiong, C., Dai, Z., Callan, J., Liu, Z., Power, R.: End-to-end neural ad-hoc ranking with kernel pooling. *CoRR* abs/1706.06613 (2017). <http://arxiv.org/abs/1706.06613>
57. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? *CoRR* abs/1810.00826 (2018). <http://arxiv.org/abs/1810.00826>
58. Yan, F., Mikolajczyk, K.: Deep correlation for matching images and text. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3441–3450 (2015)
59. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. *CoRR* abs/1806.01973 (2018). <http://arxiv.org/abs/1806.01973>
60. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 2001, pp. 334–342. ACM, New York (2001). <https://doi.org/10.1145/383952.384019>
61. Zhang, Y., Lu, H.: Deep cross-modal projection learning for image-text matching. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 707–723. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_42
62. Zhu, L., Chen, Y., He, B.: A domain generalization perspective on listwise context modeling. *CoRR* abs/1902.04484 (2019). <http://arxiv.org/abs/1902.04484>