Cross Modal Deep Model and Gaussian Process Based Model for MSR-Bing Challenge*

Jian Wang, Cuicui Kang, Yonghao He, Shiming Xiang, Chunhong Pan
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{jian.wang, cckang, yhhe, smxiang, chpan}@nlpr.ia.ac.cn

ABSTRACT

In the MSR-Bing Image Retrieval Challenge, the contestants are required to design a system that can score the query-image pairs based on the relevance between queries and images. To address this problem, we propose a regression based cross modal deep learning model and a Gaussian Process scoring model. The regression based cross modal deep learning model takes the image features and query features as inputs respectively and outputs the relevance scores directly. The Gaussian Process scoring model regards the challenge as a ranking problem and utilizes the click (or pseudo click) information from both the training set and the development set to predict the relevance scores. The proposed models are used in different situations: matched and miss-matched queries. Experiments on the development set show the effectiveness of the proposed models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

Keywords

Image Retrieval; Deep Network; Gaussian Process; Convolutional Neural Network; Clickthrough Data

1. INTRODUCTION

The arrival of the big data era encourages the researchers to develop more creative methods so as to facilitate the real-word applications. Image retrieval is one of the hot topics in the field of multimedia. This challenge focuses on the task of image retrieval. The traditional image retrieval [12] is roughly divided into two categories: keyword based and content based. The keyword based image retrieval takes advantages of extra data such as captions, tags and descriptions to the images. Differently, the content based image retrieval takes the images as the input queries and performs the retrieval via measuring the visual similarities between the images. However, the challenge puts forward a new scenario: directly scoring the query-image pairs without any extra information but the clickthrough data [6]. In the challenge, we are given a training set and a development set. The training set consists of triads: ⟨query, image, click count⟩, and the development set is used to validate the performance of the system.

In the previous challenge, some methods [16, 15, 3, 13, 11] are proposed. In general, these methods concentrate on several points: image content based model, text based model and some other models. In the image content based model [16, 3, 11], the basic idea is that the visually similar images will have similar queries. The model first retrieves images and their associated queries by computing the visual similarities between the images, and then computes the textual similarities between the queries. The final relevance scores are based on the textual similarities. However, the model is difficult to retrieve reliable visually similar images in the big image collection with a crowd of noisy samples. The text based model [16, 15, 3, 13] can be said as an inverse process of the image content based model: 1) retrieve semantically similar queries and their related images; 2) compute the visual similarities between the images. Then the scores are given by the visual similarities.

Beyond these algorithms, in [16] a modified PageRank model is proposed, which achieved the first place in the last challenge. The model follows the assumption that the majority images under the same query are relevant to the query. We found that this assumption holds in the development set. The higher similarities to the other images, the higher relevance score the image will get. However, this method abandons the semantics of queries. The name and face correspondences are also studied in [16]. For those names appearing in the training set, the authors train identity models using corresponding faces. Additionally, some names are missing in the training set, so another strategy is used in [16]: the final score is decided by the number of names and faces detected in the query and the image. When the numbers are identical, the score will be higher than those in other cases.

In this paper, we propose two models to deal with the challenge: the regression based cross modal deep learning model and the Gaussian Process scoring model. The image features and the query features are fed into the regression based cross modal deep learning model separately, and then

*This work was supported in part by the National Basic Research Program of China under Grant 2012CB316304, and the National Natural Science Foundation of China under Grants 61272331, 61203277, 61331018, and 91338202.

MM'14, November 3–7, 2014, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2647868.2656400.

the deep network outputs the relevance scores directly, providing an end-to-end scoring fashion. Although the previous works [14, 10] focus on the multi-modal deep learning, their deep networks are learned in the form of generative models and are not adaptive in this case. The details of the deep network will be described in Section 2. We use this model only when the incoming query exists (or has highly semantical one) in the training set. In [7], the Gaussian Process has been used to rank the images of the same query by requiring that all the images must have click counts. The proposed Gaussian Process scoring model makes a modification based on the original one in [7]. If the incoming query has identical or highly semantical query in the training set, the images of the incoming query as well as the images of the identical query in the training set are all used in Gaussian Process scoring model. Since those images of the incoming query have no click counts (they are what we want to predict), their click counts are set to the same value. As for the queries that are not matched in the training set, only the images of the incoming query are used, and the pseudo click counts of these images are generated based on an assumption: the images highly related to the query are in the same cluster, whereas the other images scatter in the feature space. Accordingly, we use meanshift [1] to find the center of the cluster and assign higher click counts to those images near the center and lower click counts to those far away from the center. Notice that the Gaussian Process scoring model can handle all the incoming queries (matched and miss-matched), but the semantics of miss-matched queries are neglected. The experiments are conducted on the development set, and the results show the effectiveness of our models.

2. THE PROPOSED MODELS

In this section, the proposed models are introduced in detail. In subsection 2.1, there is a brief description of image feature extraction. Then, subsection 2.2 depicts the query match strategy. The regression based cross modal deep learning model and the Gaussian Process scoring model are described in subsection 2.3 and subsection 2.4, respectively.

2.1 CNN Feature for Images

The deep neural network has received popular attentions because it shows the remarkable capacity of feature learning in recent years. The convolutional neural network (CNN) [8] is one of the widely used architectures for image feature extraction. In this challenge, we only use deep CNN features for all images. The source code is available at the authors' website [2]. The network contains five convolution layers and three fully-connected layers, which has been trained on the ILSVRC-2012 database of 1000 categories. The outputs of the $6th$ layer are used as the image features, which is 4096-dimensional.

2.2 Query Matching Strategy

As mentioned in [5], the original queries suffer some problems: meaningless words, such as "picture" and "image"; misspelling, such as "girl" vs "gril"; near-duplicate queries, such as "cat", "cats" and "cat pictures". To deal with these problems, we first remove meaningless words, then get the stems of each query by using the OpenNLP tool[1]. In this way, some triads are merged to a single triad, reducing 23094592 triads to 9195452 triads in the training set.

For each unigram in the training set, we collect all the queries that contain the unigram. In this way, it is easy to get a query list and the frequency of unigram. The importance of each unigram is measured by its frequency reciprocal. The amount of information contained in a query is defined as the sum of the importance of its unigrams. For an incoming query, first its amount of information is calculated. Then we start to search the similar queries in the query list of unigrams. In order to speed up this process, the search is from the list of unigram with the highest importance to the lowest. The ratio of the amount of information is used to measure the similarity between queries. We set a large threshold, say 0.9, to determine whether the two queries are matched or not. By using the strategy, in the development set, there are 438 queries that are matched and the rest are not matched.

[1]https://opennlp.apache.org/index.html

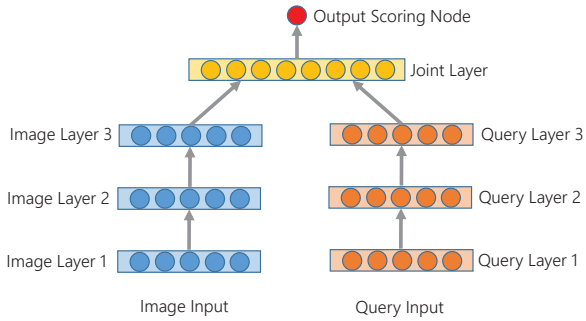

Figure 1: Structure of the proposed regression based cross modal deep learning model.

2.3 Regression based Cross Modal Deep Learning Model

The architecture of the proposed deep model is presented in Fig. 1. For each modal, there is an input layer, followed by two hidden layers. The joint layer merges the information from two modals and connects to the scoring node. Since the features of queries are difficult to extract, we represent query features in the image feature space by using linear weighting via clickthrough data. In the training set, assume that the image set is $\mathcal{I} = \{\mathbf{x}_i\}_{i=1}^{N}$, N is the number of images, and the query set is $\mathcal{Q} = \{q_i\}_{i=1}^{M}$, M is the number of queries. For each query q_i, its associated image set is $\mathcal{I}_i = \{\mathbf{x}_{i,j}\}_{j=1}^{n_i}$ and the click count set is $\mathcal{C}_i = \{c_{i,j}\}_{j=1}^{n_i}$, where n_i is the number of images of query q_i. Thus, the feature of query q_i can be formulated as:

$$\mathbf{f}_i = \sum\nolimits_{j=1}^{n_i} \hat{c}_{i,j}\mathbf{x}_{i,j}, \tag{1}$$

where $\hat{c}_{i,j} = c_{i,j} / \sum_{j=1}^{n_i} c_{i,j}$ is the normalized count. In this way, each query in the training set is allocated to a feature point in the image feature space. In our view, such kind of representations of the queries can obtain a good property that image features around a query point are semantically related to the query.

All the image and query features are mean-centered and normalized to unit length. The input layers are real-valued and the other layers are binary. The output scoring node rates the relevance via the sigmoid activation. The click

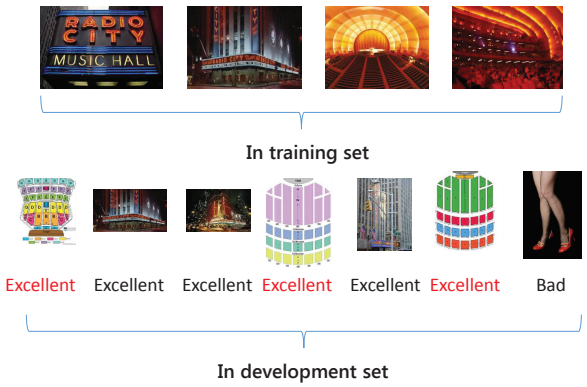

Figure 2: Images of query "radio city music hall" in both training set and development set. We can see that the images in the training set depict the building itself, but there are three images (labeled with red "Excellent") describing the inside layout of the building.

counts c are scaled into the float points by:

$$\tilde{c} = \frac{2}{1 + exp(-c)} - 1, \tag{2}$$

which are used as the true similarity scores in the deep learning. Evidently, $\tilde{c} \in [0, 1)$. In the training process, the squared error is used for the output scoring node. All the parameters are pre-trained using Contrastive Divergence [4] based on the Restricted Boltzmann Machine (RBM) (Gaussian RBM for the input layers and Binary RBM for the other layers). The backpropagation algorithm is used to fine-tune the parameters. The numbers of nodes in each layers are set as follows: image input layer and query input layer are 4096, 1000 nodes are for the image layers 2-3 and query layers 2-3, and the joint layer contains 2000 nodes.

It should be pointed out that the queries in the training set are self matched, hence the learned deep model is only suitable for the matched queries. As a result, there are only 438 queries in the development set that can be processed by the deep model. If all the queries can be semantically represented by feature vectors through an image-independent method[2], the deep learning model may be better learned and can tackle all the queries.

2.4 Gaussian Process Scoring Model

The key assumption in Gaussian Process scoring model is that the more click counts an image received, the more relevant it will be to its query. The basic form of Gaussian Process based ranking is formulated as follows [7]:

$$y(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X}_c)[\mathbf{K}(\mathbf{X}_c, \mathbf{X}_c) + \sigma^2\mathbf{I}]^{-1}\mathbf{y}_c, \tag{3}$$

where $y(\mathbf{x})$ is the relevance prediction of image $\mathbf{x}$, $\mathbf{K}$ is a kernel function (here Gaussian kernel is adopted), $\mathbf{X}_c$ is a matrix collecting all clicked image feature vectors, and $\mathbf{y}_c$ is the normalized click counts of all images. In Gaussian Process, the clicked images of the queries are preconditions. For those matched queries, they have clicked images in the training set. However, we found that the clicked images are very limited for these queries and cannot fully represent the semantics of the queries, such as the example in Fig. 2. Thus, Gaussian Process with the training images dose not perform well. Considering this, we regard the challenge as a ranking problem. Therefore, the images in the development set under the test query can be also involved into Gaussian Process by assigning them the same pseudo click count ($\frac{1}{5}$ of the max count in those of training images). The formulation in (3) is rewritten as:

$$y(\mathbf{x}) = \mathbf{K}\left(\mathbf{x}, [\mathbf{X}_c^{tr}; \mathbf{X}^{dev}]\right) \cdot \left[\mathbf{K}\left([\mathbf{X}_c^{tr}; \mathbf{X}^{dev}], [\mathbf{X}_c^{tr}; \mathbf{X}^{dev}]\right) + \sigma^2\mathbf{I}\right]^{-1} [\mathbf{y}_c^{tr}; \mathbf{y}^{dev}],$$

where $\mathbf{X}_c^{tr}$ are feature vectors from training set, $\mathbf{X}^{dev}$ are feature vectors from development set, $\mathbf{y}_c^{tr}$ and $\mathbf{y}^{dev}$ are normalized click counts of images from training set and development set, respectively. Each feature vector in $\mathbf{X}^{dev}$ is assigned to $\mathbf{x}$ to obtain a refined relevance score. We apply this Gaussian Process scoring model to those matched queries (438 queries) in the development set.

As for the queries that are not matched, a pseudo click count generation strategy is employed. In our opinion, the majority of images that are related to the query are in the same cluster with relatively small distances with each other, whereas those unrelated images scatter in feature space. Based on this assumption, we use meanshift [1] to find the center of the cluster and set higher click counts to the images that are near the center. Specifically, a radius threshold is set, when the distances are smaller than the radius, the pseudo click counts are set to 2, otherwise 1. After completing this, we can get the relevance scores using Eqn. (3).

3. EXPERIMENT

The Challenge provides two datasets, a training set for model learning and a development set for evaluation. The training set consists of 1M images with 23M click logs, and the development set contains more than 70K query-image pairs of 1000 unique queries and around 80K images.

As for the evaluation, the Discounted Cumulated Gain at 25 (DCG_{25}) for the same query is used. Then the average of all queries is computed as the result. The DCG_{25} for each query is computed as

$$DCG_{25} = 0.01757 \sum_{i=1}^{25} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

where the rel_i={Excellent=3,Good=2,Bad=0} is the manually judged relevance for each image with respect to the query, and 0.01757 is a normalizer to make the score for 25 Excellent results 1.

The experimental results on the development set are reported in Table 1. We also compared our method with those methods proposed in the previous challenges, and the results are shown in Table 2. From Table 1, we can observe that the ideal NDCG of matched queries is much higher than that of miss-matched queries, indicating that there are many miss-matched queries having less than 25 images which are "excellent" or "good" related to the queries. The Gaussian Process scoring model achieves the highest NDCG among three models with matched queries, particularly better than the original Gaussian Process model, and it is the only model that can be applied to all queries. Although the regression

[2]As far as we know, there exist some methods for extracting semantic feature vectors for a single word, such as word2vect [9]. However, the queries with short text are difficult to extract semantic features. We have tried some methods, but got unsatisfactory results.

Table 1: Performances of three models, namely regression based cross modal deep learning model, original Gaussian Process and Gaussian Process scoring model. The original and ideal NDCG scores are also presented for each part of development set. "-" means that it is not available.

	Matched Queries (438)	Miss-mathced Queries (562)	All Queries (1000)
Original NDCG	0.603	0.362	0.469
Ideal NDCG	0.800	0.580	0.684
Deep Network	0.660	-	-
Original GP	0.656	-	-
Our GP	0.678	0.433	0.540

Table 2: Comparisons between our method and other proposed methods on the entire development set. We take Gaussian Process scoring model as the final model.

	Our method	MPM [16]	CQRA [15]	EDM [3]	GLP [11]	BA [13]
NDCG	0.540	0.537	0.529	0.503	0.505	0.487

based cross modal deep learning model shows a lower performance than that of Gaussian Process scoring model, it directly bridges two different modalities and promotes the original NDCG from 0.603 to 0.660, making it worthy to be explored further. In Table 2, Gaussian Process scoring model outperforms the other methods.

The final test set of the challenge contains 147346 images and 9449 queries. The feature extraction of CNN costs about 2 hours and 40 minutes on our PC with one intel i7-4770 CPU and one Intel GTX780 GPU. As for the scoring stage, the proposed algorithm just need 43ms to process an image-query pair on average. The performance of the proposed models are listed below:

- Random: 0.3906
- Ideal: 0.5266
- Deep learning model (matched queries) + GP scoring model (miss-matched queries): 0.4877
- GP scoring model: 0.4965

4. CONCLUSIONS

In this work, we propose two models: regression based cross modal deep learning model and Gaussian Process scoring model, to assign relevance scores to the query-image pairs. The regression based cross modal deep learning model takes separated image features and query features as inputs and outputs the relevance scores directly. We represent the query features in the image feature space by using linear weighting via the clickthrough data. This model can only tackle the queries that have highly semantical or identical queries in the training set. The Gaussian Process scoring model regards the challenge as a ranking problem. It generates the relevance scores by using images with click counts in the training set and images with pseudo click counts in the development set, and we adapt it to those miss-matched queries. The experimental results on development set demonstrate the effectiveness of the proposed models.

5. REFERENCES

[1] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on PAMI*, 17(8):790–799, 1995.

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. Software available at https://github.com/UCB-ICSI-Vision-Group/decaf-release/wiki.

[3] Q. Fang, H. Xu, R. Wang, S. Qian, T. Wang, J. Sang, and C. Xu. Towards MSR-Bing Challenge: Ensemble of diverse models for image retrieval. In *MSR-Bing IRC 2013 Workshop*, 2013.

[4] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[5] X. Hua. Looking into "MSR-Bing Challenge on Image Retrieval". 2013.

[6] X. Hua, L. Yang, J. Wang, J. Wang, M. Ye, K. Wang, Y. Rui, and J. Li. Clickage: Towards bridging semantic and intent gaps via mining click logs of search engines. In *Proceedings of ACM International Conference on Multimedia*, pages 243–252, 2013.

[7] V. Jain and M. Varma. Learning to re-rank: query-dependent image re-ranking using click data. In *Proceedings of International Conference on World Wide Web*, pages 277–286, 2011.

[8] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in NIPS*, pages 1106–1114, 2012.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal deep learning. In *Proceedings of International Conference on Machine Learning*, pages 689–696, 2011.

[11] Y. Pan, T. Yao, H. Li, and C. Ngo. USTC-CityU at MSR-Bing IRC: Image search by graph-based label propagation. In *MSR-Bing IRC 2013 Workshop*, 2013.

[12] Y. Rui, T. Huang, and S. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, 1999.

[13] A. Sayko and A. Slesarev. Report on a baseline approach to the 2nd MSR-Bing challenge on image retrieval. In *MSR-Bing IRC 2013 Workshop*, 2013.

[14] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2222–2230, 2012.

[15] L. Wang, S. Cen, H. Bai, C. Huang, N. Zhao, B. Liu, and Y. Feng. France telecom orange labs a MSR-Bing challenge on image retrieval 2013. In *MSR-Bing IRC 2013 Workshop*, 2013.

[16] C. Wu, K. Chu, Y. Kuo, Y. Chen, W. Lee, and W. Hsu. Search-based relevance association with auxiliary contextual cues. In *MSR-Bing IRC 2013 Workshop*, 2013.