**Introduction**

Problem definition

Our challenge was to develop a search engine. The input would be a set of natural language query/ies and the output ought to be a shortlisted list of images, rank ordered by likelihood of relevance. Thus, our output was expected to perform well on two relevance metrics, namely, Shortlisting images and Rank ordering.

Fundamentally, our problem consisted of building a text pre-processing function to process the query and extract relevant, usable features. This would be fed into a set of models that ingest these features and output the most relevant images for the query. Subsequently, these models would be ensembled to combine the shortlisted images from all the models. Finally, there would be a ranking mechanism to ensure that the final set of images is ordered by likelihood of relevance.

Data:

Our training set consisted of 10K images of size 224x224 and their corresponding fc1000 and pool5, which are features extracted from a CNN. These are compressed (down from 50,176 dimensions to 1,000 for fc1000 and 2,048 for pool5), noise free and therefore ideal for model development. Secondly, we also have a set of 5 sentence descriptions corresponding to each image, expounding on the content of the image. Thirdly, we have tags that have been provided in the format of category: subcategory. These human-labelled tags correspond to the objects or items in the image.

In the test set we have been provided 2K images with the same features as the training set, except that there was no correspondence between an image, its features and descriptions, tags. The relevance we desire in the output is informed by training set: therefore, our goal for model building was to capture the mapping between the images (their features) and their corresponding descriptions and tags given in the training set. In this paper, we present an ensemble of models that combine to give a boosted performance over each of the individual models, thereby validating our proposed ensemble as well as ranking technique.

Previous work

We reviewed papers on cross-modal retrieval i.e. techniques studying mapping query data to image data. *A comprehensive survey on cross-modal retrieval by Kaiye Wang et al.* suggests using PLS as an unsupervised method for real-valued representation learning.

We have also explored techniques and work in Latent Dirichlet allocation, for the purpose of compressing our bag of words of description. *Latent Dirichlet Allocation by David Blei et al.* explains how to model a collection as a finite mixture of over an underlying set of topic probabilities. We also use the nltk library: corpus to remove stop-words in pre-processing our queries to create bag of words. However, there were limitations of directly using this technique, such as the assumption of topic distribution to be of a Dirichlet prior, which means that there is sparsity in terms of the number of topics. This means that the documents are assumed to cover only a small set of topics and that topics use only a small set of words frequently- however, we have a wide variety of topics in our set of images.

**Reverse architecture**

In the reverse procedure, we leveraged our test and training set to construct a new set of features for the test set. This is explained further here.

For each image in the testing set, we did the following:

1) Pick up its fc1000 feature. Perform a 20-Nearest Neighbours with the training set, using their fc1000 features. This gives us the closest 20 training images to this test image.

The intention here is to leverage the stronger mapping in data structures of a similar type to bridge the heterogeneity map. In this case specifically, the mapping between the fc1000 features in expected to be stronger, thus a simple k-Nearest Neighbours selects relevant images from the training set, for our test image.

2) For these 20 training images, we pick their bag of words of description, and create a new bag of words vector by averaging all of these 20 bag of word vectors.

3) This new averaged bag-of-words vector is assigned to the test image. The idea is that this averaged bag-of-words description will be an ‘average’ description found from the closest images in the training set.

4) Given a natural language query, our pre-processing converted it to a bag of words description. We then performed a k-Nearest Neighbour between the input BoW and the new BoW description that we created for the test images. Again, our intention was to leverage the stronger mapping within the bag of words data structures.

We performed this process with the following permutation of parameters and evaluated as follows, changing one parameter ceteris paribus:

* pool5 < fc1000
* BoW\_nouns < BoW\_all
* We have also attempt this with the following number of neighbours from the training set in the k-NN: Top 10, Top 30 < Top 20

Therefore, Reverse ( fc1000, BoW\_all, Top 20 ) does the best job

**Conclusion**

We concluded by noting that the top performing models were the following:

1) PLS on descriptions to image features

2) PLS on tags to image features

3) Reverse mapping and k-Nearest Neighbours

We see that for this problem, our input would essentially be a bag of words which has to be mapped to abstract image features captured by the CNN.

PLS:

Since these are 2 disparate spaces, PLS does a marvellous job because, at the very essence, it is formulated to handle 2 matrices of different width i.e. heterogeneity gap, by projecting them to new spaces that are of equal dimensions.

Using PCA in conjunction with PLS:

Since PCA also accomplishes the goal of unsupervised compression while preserving variance of the data-set, we have attempted to apply it to the BoW matrix before using the PLS. We find that this considerably improves the running time, since PLS involves far greater number of operations to accomplish the same task.

Using image tags vs. image descriptions within PLS:

We find that while both are important features, image tags are of inherently different spaces, even in the BoW representation, since there are far fewer total number of categories and sub-categories than there are words within bag of words. This is why we built a separate model for handling BoW of tags as opposed to concatenating it with BoW of descriptions.

Assigning greater weight to sub-categories as opposed to categories was an effective approach to creating features, since they are more specific when it comes to identifying images.

We suggest exploring techniques to use tags for shortlisting or eliminating images in sequence with another regression model on BoW description as potential avenues to explore for future.

Ensemble approach:

We found that assigning weights to both the frequency and ranking of images given by our models is an effective strategy for both shortlisting and ranking images.

Ranking images:

We have attempted to apply a weighted page-rank to improve the ranking of our images, and this was not an effective approach. This could be because page-rank finds the degree of influence of images, while this is not necessarily the best degree of centrality when it comes to mapping BoW description to image features.

**Future work**

We would also like to try

Random forest regression BoW\_description to BoW\_Tags, BoW\_description to fc1000, pool 5

SVM multi-class classification on BoW\_description to Tags, with the 80 sub\_tags and 11 tags as categories.

**Acknowledgement**

We would like to thank Prof Serge, for teaching us the fundamentals the TA team including Xun, Trishala, Micheal who helped us with assignments and our doubts throughout the semester. Also, grateful to our classmates who shared their experiences with us in the duration of the course.

**References**

[1] A comprehensive survey on cross-modal retrieval by Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, Liang Wang

[2] <https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation>

[3] <https://en.wikipedia.org/wiki/Partial_least_squares_regression>

[4] <https://en.wikipedia.org/wiki/PageRank>

[5] Latent Dirichlet Allocation by David M. Blei, Andrew Y.Ng, Micheal I. Jordan

[6] <http://scikit-learn.org/stable/>