Aggregating Image and Text Quantized Correlated Components Supplementary Material

Thi Quynh Nhi Tran
CEA, LIST and CEDRIC-CNAM
thiquynhnhi.tran@cea.fr

Hervé Le Borgne
CEA, LIST
Gif-sur-Yvettes, France
herve.le-borgne@cea.fr

Michel Crucianu
CEDRIC-CNAM
Paris, France
michel.crucianu@cnam.fr

1. Introduction

This is the supplementary material to the article *Aggregating Image and Text Quantized Correlated Components* published at CVPR 2016. While not necessary to understand the work exposed in the paper, it reports some complementary results or “interesting negative results”.

Last section focuses on providing implementation details to reproduce the experiments, in particular regarding the material coming from other works.

2. Image classification on Pascal VOC07

2.1. Grid Search results

The paper presents in Table 5 exhaustive results obtained with the default values for the KCCA parameters (kernel standard deviation $\sigma = 0.2$ and regularization parameter $\kappa = 0.1$), also employed in other references [3]. However, to compare to the state of the art, we report in Tables 3 and 4 of the paper the mAP on the test set with the parameters of the best configuration found on the validation set. We describe below the detailed procedure we used for obtaining these results.

To explore the parameter space more thoroughly, we performed a Grid Search using a split of the original 5011 training documents into 4011 documents for performing the KCCA and a validation set of 1000 documents for selecting the best parameters. This split is not exactly the one employed in the literature because we wanted to use more data to learn the KCCA space. The Grid Search leads to use a kernel standard deviation $\sigma = 0.2$, a regularization parameter $\kappa = 0.5$ and a $d = 500$ dimensional KCCA space.

We apply these parameters to perform a bi-modal classification on the 1000 testing data and obtain the performances shown in Table 1. Some are slightly better than those given in the paper with the default parameters (90.37), but nevertheless very close. The best results on the validation set are obtained for $n = 5$ and $k = 32$, leading to report a score of 90.12 to compare to the state of the art, while the best result we could obtain is 90.5 with $n = 5$ and $k = 8$.

	$k = 8$	$k = 16$	$k = 32$
$n = 5$	90.50	90.15	90.12
$n = 16$	-	90.28	90.42
$n = 32$	-	-	90.24

Table 1. Pascal VOC07 bi-modal classification mAP(%) on the test set for different values of k and n on the KCCA common space, when the KCCA parameters are determined on the validation set by grid search.

Globally, it is worth noting that the results obtained with the different values of (n, k) on the test set are all above 90, showing a remarkable robustness of our approach to the parameter settings. In fact, with a KCCA optimised on the validation set, the results appear to be even more stable than those reported in the paper with the default parameters.

2.2. Cross-modal classification with a “naive” completion method

As mentioned in Section 3.3 of the paper, in order to complete an uni-modal document, a “naive” choice would be to use a vector obtained from its μ nearest neighbors among the projections from the other modality. Consequently, the set of contributors to the “modality complement” would be defined in this case as follows for a textual-only document p^T:

$$\mathcal{M}_c(p^T) = NN^{\mu}_{\mathcal{A}^I}(p^T) \quad (1)$$

and for a visual-only document p^I:

$$\mathcal{M}_c(p^I) = NN^{\mu}_{\mathcal{A}^T}(p^I) \quad (2)$$

We report in Table 2 the results obtained in the cross-modal classification task on Pascal VOC07 when using this “naive” method of completion (MACC$_n$). The performances are significantly lower than those obtained with the completion method proposed in the paper. While the highest mAP using the “naive” completion is only 16.53% for the Text-Image task and 14.83% for the Image-Text task, we obtain respectively 82.18% and 79.18% using the completion method proposed in the paper.

μ	mAP(%) Text-Image	mAP(%) Image-Text
Random	7.76	7.33
MACC$_n(\mu = 0)$	12.03	10.04
MACC$_n(\mu = 1)$	12.48	13.98
MACC$_n(\mu = 3)$	13.59	14.65
MACC$_n(\mu = 5)$	15.29	13.28
MACC$_n(\mu = 8)$	16.41	14.83
MACC$_n(\mu = 10)$	16.53	14.79

Table 2. PascalVOC07 cross-modal classification with the direct completion with μ nearest neighbors of the missing view.

As explained in the paper, this is due to the separation of the modalities in the KCCA space. This issue is directly addressed by the contribution we proposed with regard to the completion of data for cross-modal problems.

3. Image retrieval on FlickR 8K and FlickR 30K

3.1. Grid Search for KCCA parameters on FlickR 8K

We detail the Grid Search employed in our paper to select the best parameters for image retrieval on FlickR 8K. For that, several KCCA common spaces are learnt from the 6000 training data of Flickr 8K using different values of the Gaussian kernel standard deviation σ, regularization parameter κ and dimension of projected space d. In this experiment, σ is chosen among 4 possible values {0.2, 0.5, 1, 2} and κ among {0.1, 0.2, 0.5, 1}. For the dimension d of the KCCA space we test 10 possible values in the range [10, 3000]. Image retrieval is performed on these KCCA spaces, using the 1000 validation documents as queries. In the paper, we select the parameters which yield the highest R@1 on the validation set. This leads to use a Gaussian kernel with standard deviation $\sigma = 0.2$, a regularization parameter $\kappa = 1$ and a KCCA space of dimension $d = 50$. The corresponding R@1 obtained on the validation set is 25.1%.

3.2. Retrieval test on FlickR 30K with sub-optimal KCCA space

In this experiment, we examine the behavior of our method when the Grid Search is not employed. For that, the FlickR 30K dataset is randomly split into five training datasets of 6000 images each and the same parameters are used to learn the KCCA each time (hence, we do not perform Grid Search). None of the five datasets is exactly that of FlickR 8K employed in the original experiment, but we nevertheless reuse the parameter setting obtained for FlickR 8K. Specifically, we employ a Gaussian kernel with a standard deviation $\sigma = 2$, a KCCA regularization parameter $\kappa = 1$ and only $d = 50$ dimensions for the projected space. For the MACC representation, we also use a codebook of $k = 64$ words and soft coding $(n = k = 64)$ is performed. To complete the uni-modal data, we consider $\mu = 64$ nearest neighbors in the missing modality. In all cases, the auxiliary dataset $\mathcal{A}$ is always the full training set of FlickR 30K, which includes 29783 images and their captions.

	R@1	R@5	R@10
Chen et al. [2]	18.5	45.7	58.1
MACC	33.9	65.5	77.5
MACC$_{noGS}$	18.2 ± 0.4	44.6 ± 2.2	59.2 ± 1.7

Table 3. FlickR 30K image retrieval results with sub-optimal KCCA learning on 5 different training sets.

In Table 3, we report the averages of image retrieval recall rates on the 1000 testing data over the 5 dataset. It is not surprising that the results obtained on the five KCCA spaces built from FlickR 30K training data (line MACC$_{noGS}$) are weaker than those on the KCCA space built from the FlickR 8K training set (line MACC). This is because the parameters found by Grid Search on the 6000 FlickR 8K training set were reused on the five samples from FlickR 30K, for which they may not be optimal. On the other hand, it is interesting to note that the results obtained on the five KCCA spaces remain in line with the current state-of-the-art on FlickR 30K [2], even without optimizing the KCCA parameters.

4. Reproductible research

Some material to reproduce the method described in the paper (and the experiments) will be released at `http://cedric.cnam.fr/vertigo/macc/`. In the meantime, below are some hints to find concrete implementation of the methods we used from other people. When necessary (substential modification of the original code) and when the licence of the program allows it, we will release our modified version of them. Else, we will provide the material produced by the program or a script to reproduce exactly the experiment.

4.1. programs

The original KCCA code of D. Hardoon has been recently removed from its webpage and he requires from now on to send him a mail to access it[1]. A version can also still be found in other projects[2,3].

[1]http://www.davidroihardoon.com/code.html
[2]https://github.com/aalto-ics-kepaco/DeepBiosphere
[3]http://vision.cs.utexas.edu/sungju/pascal_twkim.zip

For word2vec we used the implementation provided at `https://code.google.com/archive/p/word2vec/` and modified the program `get_top_neighbors.c` to implement the process described in our paper.

For the visual descriptors, we used the Caffe framework[4] and the VGG-16 model provided at `https://gist.github.com/ksimonyan/211839e770f7b538e2d8`.

All the SVM classification were performed using the Leon Bottou's SvmAsgd which implementation can be found at `http://leon.bottou.org/projects/sgd`. See *e.g.* [1] for detail.

4.2. data

Most of data is easily accessible on its original website, either for pascalVOC 2007[5], Flick8k[6] and FlickR30k[7].

For PascalVOC07, we used the tags collected by Hwang and Grauman in [4] and provided on their group's webpage[8]. Moreover, this package contains a version of the Hardoon's KCCA code. Some tags are nevertheless noisy: for instance, some tags are concatenated and result into a unique (non exeisting) word (*e.g. sunsetoverthesea*), naturally absent from the Word2Vec vocabulary. To improve the quality of textual features, we automatically separate the words (producing *e.g. sunset over the sea*) before employing Word2Vec. For this, each tag is matched to the tag dictionary and we retain only the valid largest substrings. The exact process is described in Figure 1 (Python code).

References

[1] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. NIPS Foundation (http://books.nips.cc), 2008.

[2] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, June 2015.

[3] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, pages 853–899, 2013.

[4] S. J. Hwang and K. Grauman. Reading between the lines: Object localization using implicit cues from image tags. *TPAMI*, 34(6):1145–1158, June 2012.

[4]https://github.com/BVLC/caffe/

[5]http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html

[6]http://cs.stanford.edu/people/karpathy/deepimagesent/flickr8k.zip

[7]http://cs.stanford.edu/people/karpathy/deepimagesent/flickr30k.zip

[8]http://vision.cs.utexas.edu/sungju/pascal_twkim.zip

```
dict_file = ’Final_Tag_List.txt’
tags_file = ’allTags.txt’
final_tags_file = ’allTags.txt.cleaned’

# collect words in dictionary
dict = []
with open(dict_file, ”r”) as file:
    for line in file:
        dict.append(line.strip())
file.close()
print len(dict)

# clean each row of tags
with open(tags_file, ”r”) as file, open(final_tags_file,”w”) as outfile:

    for line in file:
        tags = line.strip().split(’ ’)
        print line

        final_tags = []
        for t in tags:
            # for each tag string, candidate list contains
            # all possible words (from dict) that may be
            # present in tag string
            candidate = []
            for d in dict:
                if d in t:
                    candidate.append(d)
            # we take only the longest word among possible
            # words in candidate
            # (for ex: to avoid the case ”sunglasses” decomposes
            # into ”sun”, ”glasses”, ”sunglasses”)
            for c in candidate:
                if any((c in d) and (len(c)!=len(d)) for d in candidate):
                    print()
                else:
                    final_tags.append(c)

        for f in (set(final_tags)):
            outfile.write(’%s ’%f)
        outfile.write(’\n’)
file.close()
outfile.close()
```

Figure 1. Python program to clean the tags