A Cross-Modal Concept Detection and Caption Prediction Approach in ImageCLEFcaption Track of ImageCLEF 2017

Md Mahmudur Rahman, Terrance Lagree, Martina Taylor

Computer Science Department,
Morgan State University, Baltimore, MD, USA
{md.rahman,telag1,matay15}@morgan.edu

Abstract. This article describes the participation of the Computer Science Department of Morgan State University, Baltimore, Maryland, USA in the ImageCLEFcaption under ImageCLEF 2017. The purpose of this research and participation is to be able to predict the caption and detect UMLS concepts of an unknown query (test) image by using Cross Modal Retrieval and Clustering techniques. In our approach, for each image (without any caption or concept information) in the test set, we find the closest matching image in the training set by applying similarity search (e.g., content based image retrieval) in a combined feature space of color, texture, and edge-related visual features. By linking the associated caption and UMLS concepts of the closest matched image, further processing are performed to extract terms (keywords/concepts) to form a text feature vector and finally return the top ranked terms as predicted concepts (caption) from the best matching cluster centroids which are previously generated by applying K-means clustering in a term-document matrix of the training set. In this article we present main objectives of experiments, overview of these approaches, resources employed, and describe our submitted runs and results with conclusions and future directions.

1 Introduction

This article describes the main objectives of cross modality matching approach based on our first year participation in ImageCLEF 2017 [1] for the ImageCLEFcaption track. This track consists of both Concept Detection and Caption Prediction Tasks [2]. For the Concept Detection, participating systems are tasked with identifying the presence of relevant UMLS concepts in images appeared in bio-medical journal articles (PubMed Central). For the Caption Prediction, participating systems are tasked with composing coherent captions for the entirety of an image based on the interaction of visual information content and the detected concepts from the first task.

Besides in clinical settings, bio-medical images are also sources of essential information for research and education in biomedical literature. For example, authors of journal articles frequently use images to elucidate the text and to illustrate important concepts or to highlight special cases as Region of Interests

(ROIs) [3]. Overall, biomedical literature incorporates an approximation of 100 million figures, whereas the biomedical open access literature of PubMed Central of National Library of Medicine (NLM) alone contained almost two million images in 2014 [4]. The images contained in biomedical articles are seldom self-evident, and much of the information required for their comprehension can be found in the text of the articles in which they appear. Figure captions, article titles, abstracts, and snippets of body text from within the articles all contribute to image understanding. Hence, biomedical images cannot be easily understood when they are removed from their original context. Given the rapid pace of scientific discovery in medical field, it poses significant challenges to transform of massive volumes of image and text data from biomedical articles into useful information and actionable knowledge in the form of effective and efficient search process [3].

There has been a lot of interest in information retrieval, computer vision, and multimedia community recently in developing cross and multi-modal image retrieval techniques with the massive explosion of multimedia content on the web. Multimedia contents, such as web pages, scientific publications, and document images convey information using multiple modalities, including text, layout/style and images. However, an intrinsic problem here is to investigate the semantic correlation amongst the text and image data. Different models have been proposed to learn the dependencies between the visual content of an image set and the associated text captions, then allowing for the automatic creation of semantic indexes for un-annotated images [5–8].

Motivated by these approaches in general domain, in the following sections, we describe our cross-modal search approach towards the concept detection and caption prediction tasks in ImageCLEFcaption for bio-medical images in journal articles. In the following sections, we describe our content-based visual search approach (Section 2) to link test images to closet match images in the training set, text feature extraction and K-means clustering in a term-document matrix (Section 3) from associated image concepts and captions and detection and prediction of concepts and captions respectively (Section 4) for test images based on using Python 3.5, with OpenCV 2.0, sklearn, scikit-image, NLTK and mahotas packages. Section 5 describes our submitted runs and results that we achieved and finally Section 6 provides our conclusions and future works.

2 Content-Based Visual Similarity Search Approach

We at first performed a content-based visual similarity search for each image in the test set as an example query image to a content-based image retrieval (CBIR) system where images in both training and validation sets are indexed at first by extracting several low-level color, texture, and edge related visual feature. The purpose of this similarity search is to find the closet matching image in the training (validation) set for each query (test) image and using its associated caption (concepts) for further processing (keyword extraction, clustering) for caption prediction and concept detection. So, the CBIR search is the first step of the pipeline of our cross-modal process to link unknown test images with associated captions and UMLS concepts of known images with captions (concepts). To save computational time, each image is resized (100 x 100 pixels) and the following features are extracted:

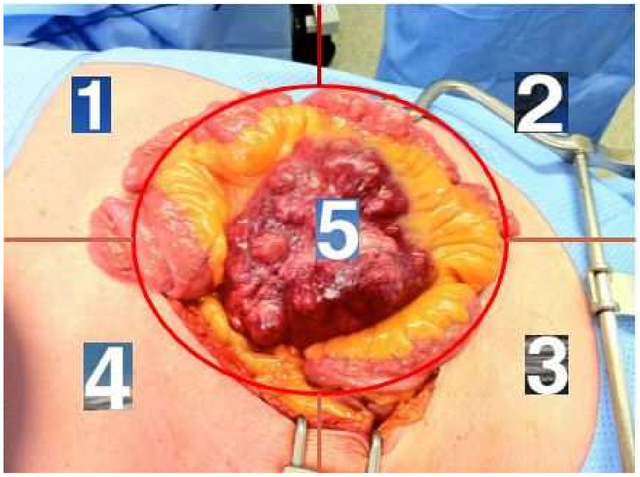

Fig. 1. Example of dividing our image into 5 different segments

Local Color Descriptor: Instead of computing a color histogram for the entire image, a 3D normalized HSV color histogram is computed for different regions (1) the top-left corner, (2) the top-right corner, (3) the bottom-right corner, (4) the bottom-left corner, and finally (5) the center of the image as shown in Fig. 1. Each region is represented by a histogram with $8x \times 12 \times 3 = 288$ entries where Hue, Saturation and Intensity values are quantized to 8, 12, and 3 bins respectively. Given 5 regions, the overall feature vector is $5 \times 288 = 1440$ dimensionality. Using regions-based histograms rather than global-histograms allows us to simulate locality in a color distribution.

In addition to the color descriptor, we extracted the well-known Local Binary Patterns (LBPs) [9] as a texture feature and Histogram of Oriented Gradients (HOG) [10] as an edge-related feature from each image. LBPs also compute a local representation of texture by comparing each pixel with its surrounding neighborhood of pixels. The first step in constructing the LBP texture descriptor is to convert the image to gray-scale. For each pixel in the gray-scale image, we select a neighborhood of size radius (r) surrounding the center pixel. A LBP value is then calculated for this center pixel and stored in the output 2D array with the same width and height as the input image. We initialized our LBP descriptor using a $numPoints = 24$ to store the number of points and $r = 8$ for the radius. The process at first generates a 2D array with the same width and height as our input image each of the values inside the array ranges from

$[0, numPoints + 2]$, a value for each of the possible $numPoints + 1$ possible rotation invariant prototypes along with an extra dimension for all patterns that are not uniform, yielding a total of $numPoints + 2$ unique possible values. Finally, our LBP feature vector as a normalized histogram ($numPoints + 2 - dimensional$), which counts the number of times each of the prototypes appears and normalized to the range of $[0, 1]$.

HOG is known as a keypoint descriptor in literature which expresses the local statistics of the gradient orientations around a keypoint [10]. The HOG feature can express object appearance due to the reason that the histogram process gives translational invariance the gradient orientations are strong to lighting changes. The HOG feature extraction process consists of three phases. In first phase first order derivatives in x and y directions are computed and the image is divided into $m \times n$ tiled regions. Gradient orientations quantized into n bins. Then, for each tiled region 1-D histogram of gradient orientations which is weighted by gradient magnitude is accumulated. Eventually, obtained feature vectors are normalized to provide robustness to illumination changes and HOG feature vectors are collected for all blocks over detection window.

For similarity matching, each feature is concatenated to form a combined feature vector and Euclidean distance measure is used for k-nearest neighbor image similarity where top matching images are ranked from a low to high scores in the range of $[0, 1]$ and only the top ranked (closest match) image is selected for each query (test) image.

3 Text Feature Extraction and Clustering

Our next step of the process is the text feature extraction and indexing (creating a document-term matrix for subsequent clustering) of associated image captions (concepts) of training (validation) images and perform clustering to form natural groups of images with similar (related) captions and concepts. For the caption prediction task, each associated caption of training images is converted to lower-case, all punctuation are removed and tokenized into its individual words. Stopwords are removed using NLTK's "english" stopword list and subsequently terms are removed from the vocabulary that occur in fewer than 10 captions, and finally the remaining words are reduced to their stems using NLTK's Snowball stemmer, which finally form the vocabulary list $T = \{t_1, t_2, \cdots, t_N\}$ of index terms or keywords of the image captions. Finally, a document-term matrix $\mathbf{M}$ is created based on T, where each caption is modeled as a vector of keywords (terms) as

$$\mathbf{f}_j^D = [\hat{w}_{1_j}, \cdots, \hat{w}_{i_j}, \cdots \hat{w}_{N_j}]^{\mathrm{T}} \tag{1}$$

where each $\hat{w}_{i_j}$ denotes the *tf-idf* weight of a keyword $t_i, 1 \leq i \leq N$ in the caption of image I_j. This weighting scheme amplifies the influence of terms, which occur often in a document (e.g., *tf* factor), but relative rarely in the whole collection of documents (e.g., *idf* factor[11]. The document-term matrix is converted to a sparse matrix in Python which only records non-zero entries to save memory space as we have significant number of entries that are zero.

However, with a size of $T > 20,000$, our matrix is still too large. Hence, to reduce the dimension further, Latent Semantic Analysis (LSA) is applied and further analysis (clustering) is performed in the LSA projected feature space by keeping explained variance above 90%. In many instances, LSA is found suitable to reduce dimensionality in a spare matrix and discover latent patterns in the data.

For the concept detection task, the text feature extraction and indexing approaches is more straightforward as we do not to perform any extra preprocessing steps, such as removal of stopwords, tokenization, stemming, LSA etc. In this case, we generate the vocabulary list T based on the presence of all the UMLS concepts in the images of the training set and generate the sparse matrix accordingly for further analysis in the next step of the process.

Our final goal is is to partition N data points (text feature vectors of training set) into K clusters by applying K-means clustering [12]. In K-means, each feature vector of image caption (concepts) is assigned to a cluster with the nearest mean where the mean of each cluster is called its "centroid" or "center". Overall, applying K-means yields K separate clusters of the original n data points. Data points inside a particular cluster are considered to be "more similar" to each other than data points that belong to other clusters. For this, we used the scikit-learn implementation of K-means in Python by experimenting with different number of clusters ($n\_clusters$) and used parameters ($e.g., max\_iter = 300$ *and* $tol = 0.0001$) for maximum number of iterations for a single run and relative tolerance with regards to inertia to declare convergence. The K-means algorithm aims to choose centroids that minimise the inertia, or within-cluster sum of squared criterion, which can be recognized as a measure of how internally coherent clusters are. After generating the clusters, the top terms per cluster are ranked and used for later use of caption prediction and concept detection of an unknown query (test) image. It is assumed that caption (concepts) that belong to a given cluster will be more similar in terms than belonging to a separate cluster.

4 Detecting Concepts and Predicting Captions of Test Images

After performing all the above processes (e.g., visual feature extraction, CBIR search, text feature extraction, and clustering), our final task is to detect concepts and predict captions of unknown images (without associated captions and UMLS concepts) in the test set. For this step, we at first find the closest matching image in the training (validation) set for each test image by applying the similar visual feature extraction and similarity search process described in Section 2. After finding the associated captions (concepts) of closet matching images, we next generate the text feature vectors accordingly and find the closet cluster labels by applying a minimum distance matching function to previously generated "centroid" or "center" in the training set as described in the previous section. Based on the cluster labels, we look-up for the top terms (keywords) and return those as assumed image captions or concepts. Below, all the steps of the above process is described algorithmically:

Algorithm 1 Concept Detection/Caption Prediction Process

```
Initially, resize and extract color, texture, and edge-related visual features $F$ (local color descriptor, HOG, and LBP) for images in the training set.
Extract text feature from captions (concepts) in the training set to generate the vocabulary list $T$ and document-term matrix $\mathbf{M}$
Apply K-mean clustering in matrix $\mathbf{M}$ to generate different number of clusters $(n_clusters)$.
for $j \in S$ images in the Test Set do
    Resize and extract visual features for test image $I_j$.
    Extract text feature vector from associated caption (concepts) $D_j$ by using vocabulary list $T$.
    Find the closet cluster label by applying a minimum distance matching function between feature vectors of caption (concept) and cluster centroids.
    Return and print the top (K) terms (keywords) from the best matching centroids.
end for
Finally, generate the result file (run) for test images with image name and associated caption (concepts).
```

5 Submitted Runs and Results

This section provides descriptions of our submitted runs and analysis of the result. We performed, feature (visual and textual) extraction and clustering in both the training set of around 164K images and validation set of 10K images with associated captions and UMLS concepts. We submitted the following four runs for the concept detection task:

1. **_DET_Morgan_result_concept_from_CBIR.csv** : This is our baseline run for the concept detection task. In this run, each test image is automatically acted as a query image to our CBIR system to find the closet matching image in the training set and use the associated concepts as the concepts for the test image.

2. **_DET_Morgan_result_concept_from_train_Kmean_top20.csv** : In this run, each test image is automatically acted as a query image to our CBIR system to find the closet matching image in the training set and use the associated concepts to form a text feature vector. This vector matches to the closet cluster centroids out of 50 clusters previously generated by K-means in the training set and returns the top (20) terms (concepts) of that particular centroids.

3. **_DET_Morgan_result_concept_from_val_Kmean50_top15.csv** : In this run, each test image is automatically acted as a query image to our CBIR system to find the closet matching image in the validation set and use the associated concepts to form a text feature vector. This vector matches to the closet cluster

centroids out of 50 clusters previously generated by K-means in the training set and returns the top (15) terms (concepts) of that particular centroids.

4. **_DET_Morgan_result_concept_from_train_Kmean300_top15.csv** : In this run, each test image is automatically acted as a query image to our CBIR system to find the closet matching image in the training set and use the associated concepts to form a text feature vector. This vector matches to the closet cluster centroids out of 300 clusters previously generated by K-means in the training set and returns the top (15) terms (concepts) of that particular centroids.

For concept detection task, evaluation is conducted in terms of average (mean) F1 scores between system predicted and ground truth concepts in the test set [2], which was generated based on the UMLS Full Release 2016AB.

Table 1. Results of the four Submitted Runs for the Concept Detection Task

Run ID	Run Type	F1 Score
1494048330426_DET_Morgan_result_concept_from_CBIR.csv	Auto	0.0273
1494048615677_DET_Morgan_result_concept_from_train_Kmean_top20.csv	Auto	0.0434
1494049613114_DET_Morgan_result_concept_from_val_Kmean50_top15.csv	Auto	0.0461
1494060724020_DET_Morgan_result_concept_from_train_Kmean300_top15.csv	Auto	0.0498

Our last run in the Table 1 ranked fourth group wise with a mean F1 score, 0.0498 when no external resources were used.

For the Caption Prediction task, we tried to submit few run based on following the same process described for the Concept Detection task. However, there were some problems in our runs (result files) and we received errors while the files were parsed by the evaluation tool as provided by the CLEF organizer. We are currently trying to fix the problem and will evaluate and analyze our results at a later time.

6 Conclusions

This article describes the cross-modal strategies of the CS Morgan group for the concept detection and caption prediction tasks of the ImageCLEFcaption track. Instead of directly performing image understanding, our cross-modal approach tries to link test images with images in the training set based on visual similarity at first and from there further text processing and clustering are performed to detect concepts and predict captions from groups(clusters) where images with similar concepts/captions reside. Our results indicate that clustering with large number (300) of centroids is better in terms of mean F1 score. However, content-based approaches to image retrieval are not yet advanced enough to achieve the precision of text-based approaches, and we are currently working towards directly mapping image region to concepts aided by a ground-truth training set of image patches.

Acknowledgment

This research is supported by the NSF HBCU-UP Research Initiation Award No. 1601044. We would like to thank the ImageCLEF [1] organizers for making the database available for the experiments.

References

1. Ionescu, B., Müller, H., Villegas, M., Arenas, H., Boato, G., et al. : Overview of ImageCLEF 2017: Information extraction from images, Title: Experimental IR Meets Multilinguality, Multimodality, and Interaction 8th International Conference of the CLEF Association, CLEF 2017, Proc. of LNCS. **10456** (2017)
2. Eickhoff, C., Schwall, I., Seco de Herrera, A. Garca and Müller, H.: Overview of ImageCLEFcaption 2017 - the Image Caption Prediction and Concept Extraction Tasks to Understand Biomedical Images. CLEF working notes. CEUR, (2017)
3. Simpson, M.S., You, D., Rahman, M.M., Xue, Z., Demner-Fushman, D., Antani, S.K. and Thoma, G.R.: Literature-based biomedical image classification and retrieval. Comput Med Imaging Graph. **39** (2014) 3–13
4. Demner-Fushman, D., Antani, S.K., Simpson, M.S. and Rahman, M.M. : Combining Text and Visual Features for Biomedical Information Retrieval and Ontologies. September 2010 Technical Report to the LHNCBC Board of Scientific Counselors. (2010)
5. Mori, Y., Takahashi, H., Oka, R.:Image-to-word transformation based on dividing and vector quantizing images with words. In Proc. MISRM99 first international workshop on multimedia intelligent storage and retrieval management. (1999)
6. Yang, Y., Zhuang, Y., Wu, F. and Pan, Y.:Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval. IEEE Trans Multimed. **10 (3)** (2008) 437-446
7. Jeon, J., Lavrenko, V. and Manmatha, R. :Automatic image annotation and retrieval using cross-media relevance models. In: Proceeding SIGIR 03 Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval. (2003) 119126
8. Rasiwasia, N., Costa Pereira, J., Coviello, E., Doyle, G., Lanckriet, G.R.G., Levy, R. and Vasconcelos, N. : A newapproach to cross-modalmultimedia retrieval. In: Proceedings of the international conference on multimedia. (2010) 251260
9. Ojala, T., Pietikinen, M. and Menp, T. : Multiresolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. **24 (7)** (2002) 971-987
10. Dalal, N and Triggs, B. : Histograms of oriented gradients for human detection. In Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) **1**(2005) 886893
11. Yates, R.B. and Neto, B.R.: Modern Information Retrieval. Addison Wesley, Reading (1999)
12. Dubes, R.C. and Jain, A.K. : Algorithms for Clustering Data. Prentice Hall, (1988)