KDEVIR at ImageCLEF 2013 Image Annotation Subtask

Ismat Ara Reshma[1], Md Zia Ullah[2], and Masaki Aono[†]

Department of Computer Science and Engineering,
Toyohashi University of Technology,
1-1 Hibarigaoka, Tempaku-Cho, Toyohashi, 441-8580, Aichi, Japan,
{reshma[1],arif[2]}@kde.cs.tut.ac.jp, aono@tut.jp[†]

Abstract. The explosive growth of image data on the web leads to the research and development of visual information retrieval systems. However, these visual contents do not allow user to query images using semantic meanings. To resolve this problem, automatically annotating images with a list of semantic concepts is an essential and beneficial task. In this paper, we describe our approach for annotating images with controlled semantic concepts, which is a scalable concept image annotation subtask in the Photo Annotation and Retrieval task of the ImageCLEF 2013. We label training images with semantic concepts. After that, given a test image, the most k similar images are retrieved from the training image set. And finally, we extract and aggregate the concepts of the k matched training images, and choose the top n concepts as annotation. In our proposed method, the textual concepts of the training images are weighted by introducing BM25. Then, we utilizes some combination of visual features vectors, which are constructed from global descriptor such as color histogram, gist as well as local descriptor including SIFT and some variations of SIFT. The visual feature vectors are used to measure the similarity between two images by employing cosine similarity or inverse distance similarity ($IDsim$) that we introduce here. For a given test image, we find the k-nearest neighbors (kNN) from the training image set based on the image similarity values. Furthermore, we aggregate the concepts of the kNN images, and choose top n concepts as annotation. We evaluate our methods by estimating F-measure and mean average precision (MAP). The result turns out that our system achieves the average performance in this subtask.

Keywords: Visual feature, Bag-of-Visual-Words, textual feature, image annotation, classification.

1 Introduction

To enable the user for searching images using semantic meaning, automatically annotating images with some concepts or keywords using machine learning technique in scalable and efficient method is to be performed. In this paper, we describe our method in scalable concept image annotation subtask [1] of the

Photo Annotation and Retrieval task in ImageCLEF 2013 [2]. Detail information on this subtask, the training, development and test set, the concepts and the evaluation measures can be found in the overview paper [1] of ImageCLEF 2013. In this subtask, the objective is to develop systems that can easily change or scale the list of concepts used for image annotation. In other words, the list of concepts can also be considered to be an input to the system. Thus the system when given an input image and a list of concepts, its job is to give a score to each of the concepts in the list and decide how many of them assign as annotations.

In our participation to the ImageCLEF 2013, we develop a system named KDEVIR to automatically annotate images with semantic concepts. We divide our approach into two steps: preprocessing and main processing. In the preprocessing step, we conduct filtering the textual features of training images, and match them with a list of controlled semantic concepts. And then, the concepts of the training images are weighted and used to label training images. After that, we measure all-pair similarity of visual feature vectors between test set and training set, and choose the k most similar images from the training set as matched images satisfying a threshold value. In main processing step, given a test image, our system retrieves the k-nearest neighbor(kNN) [3] from the matched images that was produced in the preprocessing step. And then, we aggregate all the labelled concepts of the k matched images, and measure their candidate weights. After that, we ranked the concepts based on their candidate weight, and choose the top n concepts as annotation. Our system produces good result 25 percent correct result over all test images.

The rest of the paper is organized as follows. Section 2 describes the general terminology to comprehend the essence of the paper while our system architecture is articulated in Section 3. We describes performance evaluation in Section 4, and Section 5 includes conclusion and some future direction of our works.

2 General Terminology

This section introduces some basic definitions of terminology to familiarize the readers with the notions used throughout the paper. It includes the definitions of *BM25*, Cosine similarity, and our proposed *IDsim* methods to comprehend the essence of our paper.

2.1 Okapi BM25

The Okapi best matching 25 (*BM25*) [4] approach is based on the probabilistic retrieval framework developed in the 1970s and 1980s by [5] (1981). The *BM25* formula is used for measuring the similarity between a user query Q and a document d. It is used to rank a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity). It is not a single function, but actually a whole family of scoring functions, with slightly different components and parameters. One of the most prominent instantiations of the

function is as follows. Given a query Q, containing keywords $\{q_1, q_2..., q_n\}$, the *BM25* score of a document d for the query Q is defined as follows:

$$weight(Q,d) = \sum_{i=1}^{n} \frac{TF_{q_i,d} \cdot (k_1 + 1)}{k_1 \cdot ((1-b) + (b \cdot \frac{|d|}{avgl})) + TF_{q_i,d}} \times \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

where $TF_{q_i,d}$ is the q_i's term frequency in the document d, N is the total number of documents in the collection, $\frac{|d|}{avgl}$ is the ratio of the length of document d to the average document length, and $n(q_i)$ is number of documents where the term q_i appears. k_1 and b are free parameters, usually chosen, in absence of an advanced optimization, as $k_1 \in [1.2, 2.0]$ and $b = 0.75$[1].

2.2 Cosine Similarity

Cosine similarity metric is frequently used when trying to determine similarity between two documents. In this metric, the features (or words, in the case of the documents) is used as a vector to find the normalized dot product of the two documents. By determining the cosine similarity, the user is effectively trying to find cosine of the angle between the two objects. The cosine similarity is described as follows:

$$CosSim(x,y) = \frac{x \cdot y}{||x|| * ||y||} \tag{1}$$

The similarity values depends on the features vectors. In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1, since the term frequencies (tf-idf weights) cannot be negative. The angle between two term frequency vectors cannot be greater than 90.

2.3 IDsim

Cosine similarity metric is only sensitive to vector direction, but does not consider vector length. However, to find out vector similarity, we have to consider not only vector directions but also their vector lengths. To solve these problem, we proposed a new similarity method named inverse distance similarity (*IDsim*). If U and V are two vectors, then IDSim is defined as follows:

$$IDsim = \frac{\sum U_i * V_i}{\sqrt{\sum U_i^2} * \sqrt{\sum V_i^2} * (\log_{10}(\sqrt{\sum (U_i - V_i)^2} + 1) + 1)} \tag{2}$$

The similarity values depends on the features vectors. The IDsim similarity of two documents will range from 0 to 1.

3 System Architecture

In this section, we describes our method for annotating images with a list of semantic concepts. We divide our method into two steps: preprocessing and main processing. Our whole system is depicted in figure 1.

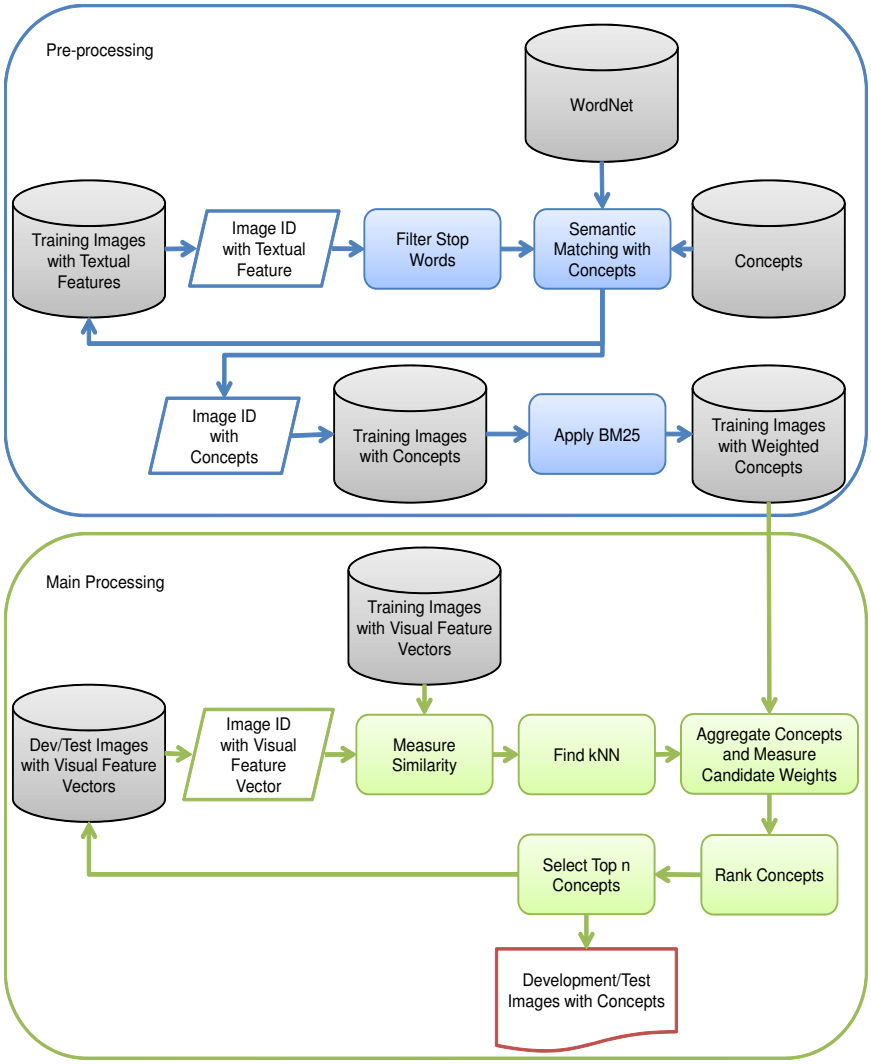

Fig. 1. System Architecture

3.1 Preprocessing

In the step, we conduct filtering of the textual features of training images, and then, matching features semantically with a list of controlled concepts. And finally, the concepts of the training images are weighted and used to label training images.

Textual Features Organizer of ImageCLEF 2013 provided to each participant textual features of training images. Textual features of the training images were collected from the web pages where the images resides. Textual features are a list word-score pairs for each training image, where the scores were derived taking into account 1) the term frequency (TF), 2) the document object model (DOM) attributes, and 3) the word distance to the image[1]. In order to ease the main processing, and reduce memory and time complexity, our system applies multiple filtering on the textual features. Because, the textual features contains some stop words, misspelled words, sometimes words from different languages than English, and some word with no semantic relation with the controlled semantic concepts. We filter out the textual features by stop words, and then, we filter out the non-English words. After that, we semantically match the feature with the list of controlled concepts. In this regard, we extend the concepts list by finding their synonyms from Wordnet 3.0 [6]. Now, we examine If the current word feature is exactly match with concept or the synonyms of the concept, and then, we consider this feature as a semantic concept. If the current word feature does not exactly match with concept list, then we apply stemming the word feature using Lucene[2] stemmer [7], and again reexamine the word feature whether it is sense of the concept or not. If it does not exactly match with the concept, we just discard it from the feature list. After matching the all the textual features of the training images, we apply bm25 [5] to estimate the weights of annotated concepts of the train textual features. Thus, our system figures out the weighted concepts list for each training image.

Visual Features Organizer provided to each participant a list of visual features vectors of training images. Visual features vectors of the training images have been computed: GIST, Color Histograms, SIFT, C-SIFT, RGB-SIFT and OPPONENT-SIFT. For the *-SIFT descriptors a bag-of-words representation is provided. Furthermore, Organizers also provided the corresponding visual feature vectors of the development or test image sets.

3.2 Main Processing

In this section, we describes the steps for annotated images with concepts. Given a set of development/test images, we select a test/development image's feature vector, and measure similarity of the image with all the training images using cosine similarity or IDsim. And then, we choose the k-nearest neighbors of images from the training set as matched images satisfying a threshold value. After that,

we aggregate all the concepts of the k matched train images, and measure the candidate weights of the concepts. And finally, we ranked the concepts based on their candidate weight, and choose the top n concepts as annotation.

Finding image similarity We apply content based image retrieval (CBIR) approach to find out similar images using equation 1 or 2. In order to find similar images of each development/test image, our system compare each image with all training images using similarity metric 1 or 2. If computed similarity exceeds a predefined threshold value which is determined empirically, then the system keeps track of those similar images with their similarity values. Finally, among all similar images of each development/test images, the system keeps track of the k nearest neighbors. We examine all combination of visual features and, empirically find out the best matching images, which we used in final runs.

Concepts retrieval In this steps, we aggregate all the concepts of train images from the weighted training image concept features as the concepts of corresponding development/test images. During aggregation, we measure the candidate weights of the concepts. We measure the candidate weight of concept by multiplying its own bm25 weight by the amount of similarity of its training image with current development/test image pair. Thus, we find out some candidate weighted concepts for each development/test image. And then, we rank the weighted concepts and choose the top n concepts. However, we empirically select top most n concepts from the ranked list of concepts list as the annotation of the corresponding development/test image.

4 Experiments and Evaluation

4.1 Runs and results

The official results of our runs are illustrated in Table 1. During experiment, we noticed that with single visual features, for example, C-SIFT produces best result. When we combine two or more features, result increases gradually. For example, the MF-sample of run 5 is 24.6 percent, which increases at run 3, 4 by adding one more feature SIFT and RGB-SIFT respectively. And the increment continued at run 1 by adding one more features Color histogram. During experiment, we also tried with TF-IDF instead of BM25, however BM25 produces better result than TF-IDF; that is why finally we did not use TF-IDF.

5 Conclusion

In this task, we filtered the textual features of the training images, and matched them with the concepts list to extract concepts for the train images, finally, estimated the weights of the concepts for every training images. After that, we conducted all-pair similarity measure between the test image visual feature

Run	Visual Features	Similarity Metric	MF-samples (%)		MF-concepts (%)		MAP-samples (%)	
			Development	Test	Development	Test	Development	Test
Run 1	C-SIFT, Opponent-SIFT, RGB-SIFT, Color histogram	IDSim	**25.3**	**22.2**	**21.1**	**18.0**	28.7	26.1
Run 2	Color histogram, GIST	Cosine similarity	25.0	20.7	19.2	14.8	26.4	23.5
Run 3	C-SIFT, Opponent-SIFT, SIFT	IDSim	24.8	21.1	18.7	15.9	28.6	24.8
Run 4	C-SIFT, Opponent-SIFT, RGB-SIFT	IDSim	24.7	20.5	18.5	15.4	**29.2**	**26.4**
Run 5	C-SIFT, Opponent-SIFT	IDSim	24.6	20.2	18.5	15.1	29.0	25.6
Run 6	Opponent-SIFT, SIFT	IDSim	24.5	20.8	18.4	15.7	28.3	24.3

Fig. 2. Official results of our runs

vectors with the training images visual feature vectors by introducing a similarity metric named IDsim satisfying a threshold. And then, we selected the kNN images from matched trained images. After that, we aggregated all concepts from the kNN images and measured the candidate weights. And finally, the aggregated concepts are ranked, and the top n concepts are selected as annotation for a test image. Our Result at ImageCLEF 2013 was at middle position. We will improve our system by implementing efficient semantic matching of the features including hyponym. We will also try to introduce efficient machine learning technique to develop scalable image annotation system.

6 Web sites

[1]Scalable Concept Image Annotation, http://imageclef.org/2013/photo/annotation
[2]Lucene Search Engine, http://lucene.apache.org

References

1. Mauricio Villegas, Roberto Paredes, B.T.: Overview of the imageclef 2013 scalable concept image annotation subtask. In: CLEF 2013 working notes. (2013)
2. Caputo, B., Muller, H., Thomee, B., Villegas, M., Paredes, R., Zellhofer, D., H. Goeau, A.J., Bonnet, P., Gomez, J.M., Varea, I.G., Cazorla, M.: Imageclef 2013: the vision, the data and the open challenges. In: Proc CLEF, LNCS (2013)
3. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is nearest neighbor meaningful? In: Database TheoryICDT99. Springer (1999) 217–235
4. Sparck Jones, K., Walker, S., Robertson, S.: A probabilistic model of information retrieval: development and comparative experiments:: Part 2. Information Processing & Management **36**(6) (2000) 809–840
5. Robertson, S., Walker, S., Beaulieu, M.: Okapi at trec-7: automatic ad hoc, filtering, vlc and interactive track. Nist Special Publication SP (1999) 253–264
6. Fellbaum, C.: WordNet. Springer (2010)
7. Hatcher, E., Gospodnetic, O., McCandless, M.: Lucene in action, edn (2010)