Chapter 5

Automatic annotation for image retrieval

An efficient search on multimedia documents like images, music, videos *etc.*, needs an organized orientation. This orientation is achieved through indexing, ranking and finally retrieval. Image retrieval encompasses browsing, searching and retrieving images from a large database of digital images. With the advent of social networks and e-commerce and the data explosion over the past few decades, image search has received a renewed interest from the computing community. For image search using text, annotations or captions are the only feasible way to aid an efficient search wherever associated image description is not available. However, in real world images are rarely found to be annotated. There are two main challenges relating to

automatic image annotation: the first one is concerned with the seed images annotated manually by humans which are generally used a starting point for automating image annotation; the second is how to automate annotation to a larger and varied group of images beyond the examples used as a starting point in the process.

Thus, finding a correct annotation for images from a pre-defined vocabulary is a necessity and lays a foundation for image searching using text. However, given the humongous size and the vagaries associated with it, manual image annotation of online and offline image databases is neither a reliable nor a viable method. Lately many tools and techniques have been devised to facilitate crowdsourcing to annotate images under a controlled setting (*e.g.* Amazon Mechanical Turk). But, these processes too have certain limitations[1]. This fact establishes the need for automatic image annotation algorithms that are scalable and efficient. Automatic image annotation using only visual features is not a trivial task as it involves cross-media semantic linkage. To establish such a link between an image and a natural language (such as English) by inferring semantic concepts from images without human intervention is a big challenge. Bridging this gap by learning on some loosely labeled data is an open problem for the current machine intelligence and statistical learning research community [29].

[1] http://www.yudkin.com/crowdsourcing.htm

5.0.1 Motivation

Probabilistic approaches for automatic image annotation are fairly common where the association of words with images are quantified with a probability count [51]. Co-occurrence based approaches are equally popular. Generally, in such approaches, images are clustered into regions, called *blob*, and then word to blob co-occurrence is counted and the word-blob pair with the highest count is used to annotate the images. Jeon *et al.* [45] use document based expansion to generate words for each test image by associating probabilities from the joint distribution of blobs and words. This model annotates each image with a vector of probabilities for all the words present in the vocabulary. However, in these blob to word based approaches, there is a chance of integrity loss as the blobs are confined to object levels only. For example, let us consider the image in Figure 5.1.

FIGURE 5.1: Describing a *concept*

If we consider each discrete object as a blob, discarding the background noise, we can isolate objects like 'train', 'tracks', 'platform', 'people' *etc.* using co-occurrence

based approaches. But these are individual words, which a general user is not expected to input while searching for a concept like 'railway station'. While as a collection these words corroborate the sense of a 'railway station', semantically they are vastly different from each other (such as 'people' and 'train'). In this scenario, keyword based search using a collection of (object) words can (incorrectly) return images of trains running through the countryside while a phrase like 'railway station' is far more specific and hard to confuse. It should be noted while 'railway station' could be considered to be a holonym of objects like 'train', 'tracks' *etc.*, it does not account for terms like 'people'. So, there is a marked distinction between what we define as a *concept* and what qualifies as a meronym-holonym relation. To avoid or rather minimize the loss of information either at the object level or the collective information that an image conveys, we come up with a novel concept-based approach. Linan Feng and Bir Bhanu [29] have used the co-occurrence pattern of concepts to find the semantic relation between images and words. To this end, they have hypothesized that multiple concepts that frequently co-occur across images form patterns could provide contextual cues for individual concept inference.

Also in the work by Villegas and Paredes [90], a simple k-NN based automatic image annotation is presented. The authors rely only on the automatically gathered Web data (ImageCLEF 2012 Scalable Web Image Annotation dataset) for the same. The authors use different distance metrics among which Jensen-Shannon divergence provides the best possible efficiency. Since the work done in this paper is on similar lines to ours we use the same dataset and consider this paper as one of our baselines.

5.1 Methodology

In this section, we describe our image annotation model in details which comprises of three steps. Each of these constituent steps is discussed below in details.

5.1.1 Graph Construction

Graph or network structure based solutions have grown to be quite popular to understand or to analyze the complex nature of the systems or objects. One such application includes a graph-based ranking for keyphrase extraction from text documents in which a graph is built from the input documents and its nodes are ranked according to their importance [62]. Each node of the graph corresponds to a candidate keyphrase from the document and an edge connects two related candidates. We borrow this notion of representing *relatedness* between two words (or concepts) from the graph-ranking method, but build the graph differently altogether. We construct the concept graph, $\mathcal{C} =< V,\ E >$, with all concepts and associated words present in the training dataset. In this graph, concepts are the nodes (V). We initially assume that the graph is fully connected and the co-occurrence count or mutual information score between any two nodes are the edge weights (W). The intuition of concept graph arises from the fact that for any two images, words or image-word pair to

co-occur, there must be an information overlap. We use two different measures to assign the edge weights of the graph. The first one is *co-occurrence count* and the other is *mutual information*.

We compute the number of co-occurrences of any two words or concepts (nodes). It should be mentioned that each concept or word is associated with an image in the dataset as explained later in Section 6.2. Suppose there are $\mathcal{I}$ images in the training dataset. Now, let us consider that a word w_i occurs n_1 times in the dataset, and another word w_j occurs n_2 times, and these words coincide on exactly, say, p number of images. In such a scenario, the total number of arrangements in which the two words coincide in p images will be given by Equation 5.1.

$$\binom{I}{p, n_1 - p, n_2 - p} \tag{5.1}$$

Hence, the probability (or in other words, the co-occurrence count C_m) that two words that appear in n_1 and n_2 images each and are randomly and independently distributed among I images coincide in exactly m of them can be given by Equation 5.2:

$$C_m = \frac{(n_1)_m (n_2)_m (I - n_1)_{n_2 - m}}{(I)_{n_2 - m} (I - n_2 + m)_n (m)_m} \tag{5.2}$$

where $i_j = i(j-1)...(i-j+1)$ for any $i \geq j$.

Now, let us focus on our second edge-weight measure *i.e.* mutual information. Let us consider that $p(w_1)$ and $p(w_2)$ are the probability that words w_1 and w_2 are associated with an image i respectively. Also, $p(w_1w_2)$ is the joint probability of image i being associated with both w_1 and w_2 concurrently. Here, the chance of image i being associated with the word w_1 and the chance of i being associated with the word w_2 are independent. Now, from the definition of mutual information, it compares the probability of observing any two random variables together (the joint probability) with the probabilities of observing the both variable independently. So, in our case, the mutual information $MI(w_1w_2)$ between the two words or nodes in the concept graph will be as shown in Equation 5.3.

$$MI(w_1w_2) \propto log\frac{p(w_1w_2)}{p(w_1)p(w_2)} \tag{5.3}$$

If there is a genuine association between the two words w_1 and w_2 the joint probability $p(w_1w_2)$ will be much larger than the chance $p(w_1)p(w_2)$. This implies $MI(w_1w_2) \gg 0$. If there is no valid relationship between the two words w_1 and w_2, *i.e.* they are not concurrently associated with any image, then $p(w_1w_2) \approx p(w_1)p(w_2)$. Now, we compute mutual information, $MI(w_i, w_j)$, between any two words w_i and w_j for any image as shown in the Equation 5.4:

$$MI(w_i, w_j) = p(w_i, w_j) log \frac{p(w_i w_j)}{p(w_i)p(w_j)} \tag{5.4}$$

where, $p(w_i, w_j)$ is the joint probability between any two words w_i and w_j.

5.1.2 Graph Clustering

Non-uniform data usually contains an underlying structure arising out of its heterogeneity. The process of identifying this structure in terms of grouping the data elements is called clustering. The resulting groups are called clusters. Graph clustering is the task of grouping the vertices of the graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters [82]. It is a fairly common phenomenon that likelihood of pairs of nodes to be connected is greater if they are both members of the same community (cluster). Hence, community detection becomes a useful approach to find the similar nodes from a complex graph. Taking cue from this aspect, we use community detection to find the pattern of the concepts and to find the similarity between words and concepts from our constructed graph stated in the above subsection 5.1.1.

Among the various community detection techniques that are available in the existing literature, we select *Infomap* [75]. Infomap basically is a multilevel network clustering algorithm based on the Map Equation. The community structure is represented

through a two-level nomenclature based on Huffman coding: one to distinguish communities in the network and the other to distinguish nodes in a community. Using this nomenclature, the problem of finding the best partition is reduced to minimizing the information quantity required to represent some random walk in the network. With a partition containing few inter-community links, the walker (or the traversing agent) will be forced to stay longer inside communities. Thus, only the second level is needed to describe its path, leading to a compact representation. The criterion is optimized using simulated annealing. Many community detection algorithms, including the popular modularity maximization approach, do not consider how the network structure is formed. However, Infomap captures how the network structure influences over the behavior of nodes. The inherent map equation helps to highlight and to simplify the network structure. Using Infomap, we obtain clusters of concepts related to each other. Consequently, the corresponding images are clustered as well. This exercise reduces the complexity as well the search space for the incoming test image in which it is to be embedded. All set of node pairs along with their associated edge weights are fed to the Infomap as input and it returns a mapped set of images with their associated clusters to which they belong. The clusters are denoted by $\mathcal{K}$.

5.1.3 Ranking and extraction of annotations

Next step is to identify the possible words (concepts) for annotation *i.e. candidate annotations*. For a word to be a candidate annotation, it has to abide by all the rules as stated below:

i. The size of each candidate annotation set *i.e.* the number of words present in the annotation, must be at most five. We confine to this limited size because, in general, a larger annotation will unnecessarily lengthen the search query [8].

ii. Each cluster of the concept graph is identified with a centroid. For each test image, the Euclidean distance from the centroid image (and its associated concepts) is computed. Whichever cluster has the minimum distance bears the closest resemblance to the test image and hence the test image is assigned to it. Now, to find the candidate annotation, we start our walk from the centroid concept (word) until the test image is reached.

iii. The following candidate words must have a positively weighted edge with the previous word (node).

iv. If a node has more than one outgoing edges, the algorithm selects the edge with the highest value, and if there is a tie between two outgoing edges, we randomly select one of them. Hence, we follow a greedy approach.

The search for the words is terminated when the maximum pre-defined number (size of candidate annotation set) of words are selected. This process is repeated so that each node in the cluster is visited at least once.

Once all the candidate annotation have been extracted from the cluster in which the test image belongs, we rank them in descending order on the basis of *informativeness*. Informativeness of an annotation is measured by summing over all the edge weights of words present in the annotation. After ranking, we select the best among

the set of annotations.

The complete process can be formally represented as Algorithm 4. Here, steps 1 through 18 summarize concept graph formation (Section 5.1.1). We have only indicated the mutual information measure here which can be replaced with co-occurrence count accordingly. The procedure TraverseAndExtract() corresponds to the steps

Algorithm 4: Probabilistic Model for Image Annotation

Data: Training concept set D, training image set I, a test image $\hat{i}$
Result: Annotation a

1 $V \leftarrow D$;
2 **for** $\forall w_i \in V$ **do**
3 $\quad p(w_i) \leftarrow ProbOccur(w_i)$;
4 **end**
5 **for** $\forall w_i, w_j \in V$ **do**
6 $\quad e_{i,j} \leftarrow 1$;
7 **end**
8 $l \leftarrow 1$;
9 **for** $l \leq |V|$ **do**
10 $\quad$ **for** $\forall w_i, w_j \in V$ **do**
11 $\quad\quad$ **if** $p(w_i, w_j) > 0$ **then**
12 $\quad\quad\quad MI(i,j) \leftarrow p(w_i, w_j) \log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}$;
13 $\quad\quad\quad e_{i,j} \leftarrow MI(i,j)$ ▷ Replace with co-occurence count otherwise;
14 $\quad\quad$ **end**
15 $\quad\quad$ **else**
16 $\quad\quad\quad e(i,j) \leftarrow 0$;
17 $\quad\quad$ **end**
18 $\quad$ **end**
19 **end**
20 $\mathcal{C} =< V, E >$ ▷ $e_{i,j} \in E$ and $i \neq j$;
21 $\mathcal{K} \leftarrow$ Infomap($\mathcal{C}$) ▷ Generate clusters;
22 $a_{cand} \leftarrow$ TraverseAndExtract($\hat{i}$);
23 $a \leftarrow$ RankDescend(a_{cand})[1];

mentioned in Section 5.1.3. Finally, the candidate with the highest score (informativeness) is adjudged to be the test image's annotation.

5.2 Experimental Setup

- **Dataset**: The experiments are carried out on ImageCLEF 2012 Scalable Image annotation task WEBUPV dataset[2]. There are 250,000 images for training in the dataset, which includes various visual feature types, and textual features obtained from the websites in which the images appeared. The URLs of the images were obtained by querying popular image search engines (namely Google, Bing and Yahoo) when searching for words in the English dictionary. Textual features are extracted from four different sources–
  - Raw text extracted near the image with the image position marked.
  - For each image a list of word-score pairs. 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.
  - Triplets of word, search engine and rank, of how the images were found.
  - The URLs of the images as referenced in the corresponding webpages.

  Different extracted visual features are also available in the dataset such as 576-dimensional color histograms, GIST, SIFT, C-SIFT, RGB-SIFT, OPPONENT-SIFT *etc.* [91]. The text in the dataset needed to be parsed to obtain the keywords associated with the images. A mapping between image-id to set of keywords associated is also done to ease the computation. We test on 1,000 images which are available in the test set. These test images belong to 95

[2]http://risenet.prhlt.upv.es/webupv-datasets/

different concepts and the judgment for each image is also available.

- **Tools and Hardware used**: As stated earlier, for identifying clusters from the concept graph using community detection we use *Infomap* [10] implementation[3]. For rest of the implementation, we used C++ and Python. All our experiments were carried out an Intel(R) Core(TM) i7-4770 processor with clock speed of 3.40 GHz running 64-bit Linux operating system.

- **Metric**: For evaluation, we used mean F-measure (MF) since it is a standard metric and all our baseline systems have been evaluated using the same.

5.3 Results and Analysis

In this section, we discuss the evaluation of our experiments and analyze them. Before we talk about our model's performance in terms of figures, let us take a look at a theoretical comparison of the baselines against our proposed model (Table 5.1).

We can observe that each of the approaches mentioned has its share of merits and demerits. Among these, our proposed system which is based on a purely graph-based approach performs superlatively both in terms of complexity and effectiveness. From Table 5.2, we can see that both our probability-based approaches surpass all

[3]http://www.mapequation.org/

Method / Parameters	**Best @ ImageCLEF2012 Subtask2**	**JS+ZSUM**	**Our proposed model**
Characteristics	(1) Supervised learning (2) Multiclass classification using PAAL (3) Multiple labels are attached to one sample	(1) Unsupervised learning (2) k-NN based classification 3. Estimates Concept-Word Probabilities using WordNet	(1) Graph-based (2) Community detection using Graph clustering (3) Mutual Information and Co-occurrence count used to capture informativeness
Advantage(s)	(1) Efficiency achieved by averaging all pairwise loss between relevant and irrelevant labels.	(1) Use of dictionary definitions for estimating word-concept probabilities	(1) Insightful approach (2) Better clustering (3) Significantly better performance (4) Captures probability-weighted information for annotation (5) Incorporates relatedness between concepts
Disadvantage(s)	(1) Classifier needs to be trained on different type descriptors	(1) Needs external language resources	(1) Co-occurrence based approach fails for identical images

TABLE 5.1: Theoretical comparison of various models

the baselines, although the performance of the co-occurrence based approach is comparable to that of the best system at ImageCLEF2012. This, in turn, corroborates

Method	MF_1 in %
ImageCLEF2012 random baseline [91]	5.5
ImageCLEF2012 co-occurrence baseline [91]	17.1
ImageCLEF2012 best submission [88]	25.4
JS+ZSUM [90]	25.1
Co-occurrence Count based model	25.9
Mutual Information based model	$\mathbf{37.4}^{p}$

TABLE 5.2: Annotation Results Comparison

that mutual information (MI) effectively embodies the *concept* of an image while retaining the ability to retrieve images with keyword based search as well. The MI-based approach surpasses the best baseline by 48.4% improvement in terms of mean F-measure and hence can be touted as the best system. Also, the performance of the MI-based method is statistically significant against all the other listed methods (validated by two tailed variable mean t-test as denoted by a superscript p in Table 5.2). While the exceptional performance of Mutual Information based approach is due to the ability to capture informativeness of an annotation, the relatively subdued performance of co-occurrence based approach could be attributed to the presence of a few near-duplicate images. Such images create a hindrance since co-occurrence based approach tends to work well with novel images. The judgment file for the test set contained a few images which had only single worded annotation while some images had up to ten words in their annotation. This is also one of the reasons why we limited our candidate annotation length to an five, the average length. In future,

we would like to perform an empirical study the effect of length of annotation on retrieval performance.

5.4 Discussion

Automatic image annotation has attracted serious attention from researchers in the past few decades owing to its varied applications especially in Image Retrieval. Existing systems and tools focus on discrete objects present in the image and annotate them separately. However, they fail to capture the essence of a scene, termed as *concept*, depicted in an image. To counter this, we propose a probabilistic model with two different measures namely co-occurrence count and mutual information which supersedes the performance of the baselines comprehensively. To this end, we perform a greedy walk on the clusters of concepts extracted from the concept graph and finally rank them on their informativeness to obtain the best possible annotation for an image. The clusters are identified using an established community detection algorithm, *Infomap*. The work done in this paper is a subtask of a bigger project where we aim to propose a novel image retrieval system that is both scalable and efficient. Hence, as part of future work, we would like test our proposed system's efficacy and efficiency in the context of multimodal image retrieval and compare it against state-of-the-art systems. Also, we would like to explore other image annotation datasets and analyze our model's performance.