Image Annotations By Combining Multiple Evidence & WordNet

Yohan Jin
Department of Computer Science
University of Texas at Dallas
Richardson, Texas
75083-0688, USA
yohan@student.utdallas.edu

Latifur Khan
Department of Computer Science
University of Texas at Dallas
Richardson, Texas
75083-0688, USA
lkhan@utdallas.edu

Lei Wang
Department of Computer Science
University of Texas at Dallas
Richardson, Texas
75083-0688, USA
leiwang@utdallas.edu

Mamoun Awad
Department of Computer Science
University of Texas at Dallas
Richardson, Texas
75083-0688, USA
maa013600@utdallas.edu

ABSTRACT

The development of technology generates huge amounts of non-textual information, such as images. An efficient image annotation and retrieval system is highly desired. Clustering algorithms make it possible to represent visual features of images with finite symbols. Based on this, many statistical models, which analyze correspondence between visual features and words and discover hidden semantics, have been published. These models improve the annotation and retrieval of large image databases. However, current state of the art including our previous work produces too many irrelevant keywords for images during annotation. In this paper, we propose a novel approach that augments the classical model with generic knowledge-based, WordNet. Our novel approach strives to prune irrelevant keywords by the usage of WordNet. To identify irrelevant keywords, we investigate various semantic similarity measures between keywords and finally fuse outcomes of all these measures together to make a final decision using Dempster-Shafer evidence combination. We have implemented various models to link visual tokens with keywords based on knowledge-based, WordNet and evaluated performance using precision, and recall using benchmark dataset. The results show that by augmenting knowledge-based with classical model we can improve annotation accuracy by removing irrelevant keywords.

Categories and Subject Descriptors:H.3.3 [Information Systems]: INFORMATION STORAGE AND RETRIEVAL

General Terms: Management,Image Annotation, WordNet,Semantic-Similarity, Dempster-Shafer Rule, Corel Dataset.

MM'05, November 6–11, 2005, Singapore.

1. INTRODUCTION

Images are a major source of content on the Internet. The development of technology such as digital cameras and mobile telephones equipped with such devices generates huge amounts of non-textual information, such as images. We need to find the images that have objects or to find keywords that best describe its content [5] with given an unseen image. Hence, these techniques raise the possibility of several interesting applications such as automated image annotation, browsing support, and auto-illustration. Content-based image retrieval (CBIR) computes relevance based on the visual similarity of low-level image features such as color histograms, textures, shapes and spatial layout etc. However, the problem is that visual similarity is not semantic similarity. There is a gap between low-level visual features and semantic meanings. The so-called semantic gap is a major problem that needs to be solved for most CBIR approaches. For example, a CBIR system may answer a query request for 'red ball' with an image of a 'red rose'. If we undertake the annotation of images with keywords, a typical way to publish an image data repository is to create a keyword-based query interface. Images are retrieved if their descriptions/annotations (i.e., metadata) contain (some combination of the) keywords specified by the user. One approach is to simply rely on humans entirely for annotation, which is labor intensive and subjective. The alternative is to rely on a supervised algorithm to generate metadata of images. For this, we would like to address the linking problem between visual regions and keywords that appear in images. Therefore, given a set of images in which each image is captioned with a set of keywords that describe the image content, researchers have already proposed various algorithms to determine the correlation between keywords and visual tokens/regions. Once we identify a correlation between key-

words and image visual tokens/regions, this association can be used to annotate images that do not have captions. Several statistical models have been proposed in recent years [14, 6, 5, 3, 4, 11, 8] to determine the correspondence between keywords and image visual tokens/regions.

These statistical models can be categorized into a several groups, such as a translation model (TM), a cross media relevance model (CMRM) and a continuous relevance model (CRM)[19]. By analyzing the statistical relations between visual features and keywords, these models can reveal hidden semantics. However, whatever model we employ the current annotation accuracy is quite low due to the existence of too many noisy words. Therefore, it is quite difficult to get a meaningful understanding of images in this manner. Furthermore, it is impossible to distinguish between some keywords such as valley and mountain, garden and tree, cat and tiger, as designations of image content (these keywords are part of the Corel keywords). When a user query is for valley, and the retrieved images include mountains, the user will be satisfied with this result. Hence, our goal is to facilitate the steps which need to be taken to achieve a semantic understanding of images. The semantic meaning of an image will be described by a set of keywords, For example, In Fig. 1, two images include people, however, the context of people in each image is different. The first image (384008) has the keyword-'the people on the beach' and the second has the keyword-'the people in the garden'. Noisy keywords for the first and second images In Fig. 1, are 'desert snow' and 'rock goat' respectively. To remove noisy keywords for an image we will utilize correlations of keywords based on semantic similarity. Intuitively, non-correlated keywords may be treated as noisy, and discarded. For example, the correlation between 'beach and sand' is greater than 'snow and sand' based on semantic similarity given in Knowledge based, WordNet, and 'snow' will be discarded. On the other hand, 'people beach', and 'people garden' are highly correlated. In this paper we discard an annotated keyword from an image which does not correlate with other annotated keywords that appeared in that image. For this, first, we investigate various semantic similarity measures between keywords with the usage of WordNet. Each semantic similarity measure tries to find the distance between keywords using several different approaches (e.g., node-based, edge-based, gloss-based). Next, we fuse these measures using Dempster-Shafer multiple evidence combinations to make a final decision. To the best of our knowledge, this is the first attempt to improve the annotation accuracy by applying the semantic similarity between keywords with the usage of WordNet. We have evaluated the performance of our novel approach with a classical one using precision, and recall using benchmark dataset. The results show that by augmenting knowledge-based with classical model we can improve annotation accuracy.

This paper is organized as follows: Section 2 presents a description of Translation Model. Section 3 explains several semantic similarity measures along with shortcomings, presents motivation behind various measures and presents Dempster-Shafer multiple evidence combination mechanisms to fuse various measures. Section 4 presents experimental setup and results of our approach. Section 5 presents related work. Section 6 presents conclusion and a comment on future work.

384008:beach people sand desert snow

147066:people flower garden rock goat

Figure 1: An Example of Annotations with having noisy and correct keywords

2. TRANSLATION MODEL (TM)

TM is a way to automate image annotation by addressing the following problems. Although here we consider TM, however, even if we consider CMRM and CRM, our semantic similarity measure will be applicable.

With regard to TM & CMRM models, each image will be represented by a set of keywords and visual tokens. It is possible that more than one image can share the same visual token. Since the keyword of the similarity of visual tokens is ill defined compared to keywords, visual tokens will be clustered together and a finite set of visual tokens will be generated. Thus, visual tokens will be classified into groups (blob tokens) by clustering the feature space for all of the regions in the data set. Each visual token will be assigned to the label of the cluster that it belongs to (i.e., blob-token). The premise is that if some visual tokens are the same they will belong to the same cluster. Hence, to address the correspondence problem, we need to address the following issues:

1 Segment images into meaningful visual segments/tokens.

2 Clustering visual segments to generate blob-token.

3 Determine correlation between associated keywords and visual blob-tokens.

2.1 Segmentation

With regard to the first problem, we rely on normalized cut that segment images into a number of visual tokens [18]. Each visual token will be represented by a vector of colors, textures, shapes etc. Therefore, visual token means a segmented region or object, and it will be described by a set of low level features like color, texture, and shape. For example, each image segment in Corel is represented by 30 features.

2.2 Clustering to Generate blob-token

We would like to quantize image object representation. For this, we will apply clustering algorithms to group similar visual tokens (i.e., image objects) into a blob token. Thus we generate a fixed set of blob tokens. The problem is that most current image clustering algorithms do not consider the relevant features, but assign the same weight to all low-level features. Yet image data are high dimensional data, and many dimensions are irrelevant. These irrelevant dimensions will hide clusters in noisy data and confuse the clustering algorithms.

The objects in the same cluster are very similar with regard to dominant feature dimensions, but the distance or similarity measures may indicate dissimilarity due to the noisy value in irrelevant dimensions. For example, all segmented 'tiger' visual tokens have the same color; the color features are relevant for all 'tiger' visual tokens. However, shape or position features are not relevant for 'tiger'. For all 'ball' visual tokens, the relevant features are shape as compared to the color feature. Thus, the set of relevant features may be different for different clusters. The relevant features or dominant features are very useful when we measure similarity between two visual tokens (i.e., clusters). Furthermore, The problem could become even worse when the data have different scales in different dimensions [9]. Hence, we normalize data $< x_{i1}, x_{i2}, ...x_{im} >$ into its normal form using mean(μ_j) and variance(σ_j) for j-th low-level feature as $< (x_{i1} - \mu_1)/\sigma_1, (x_{i2} - \mu_2)/\sigma_2, .., (x_{im} - \mu_m)/\sigma_m >$.

2.3 Weighted Feature Selection

Our weighted feature election mechanism is as follows: First, we cluster visual tokens using K-means assuming equal weight. Second, we distribute visual tokens into clusters and update centroids. Third, for each cluster we identify the most important features and discard irrelevant features. Finally, the same process will be repeated until the algorithm converges. In fact at step 3 we apply weighted feature selection to determine the relevancy of a feature. In other words, we determine the weight of features. We represent m features in j-th cluster as $< f_{j1}, f_{j2}, ...f_{jm} >$, and corresponding weights of these features are $< w_{j1}, w_{j2}, ...w_{jm} >$. Let us assume that we have altogether N visual tokens and the dimension of a visual token is m. Then the i-th visual token in the dataset is represented by $< x_{i1}, x_{i2}, ...x_{im} >$. Hence, we need to determine dominating features across a set of visual tokens/cluster on the fly and assign more weight over others. Each feature in a cluster will be assigned a weight according to how relevant the feature is to the cluster. We present a method estimating this relevance based on a histogram analysis (See [20] for more details).

2.4 Link between Keyword and blob-token

To determine a link between keywords and blob-tokens, first we construct a probability table. Let us assume that there are W keywords, B blob-tokens, and N images. Then, the dataset can be represented by a matrix . Where in M matrix, row N corresponds to the number of images and first W column corresponds to W keywords, and next B column corresponds to B blob-tokens. Next, we calculate probability table by implementing various weight calculation strategies. Finally, the relationship between keywords and blob-tokens can be determined by probability table. For example, we assign a keyword w_i to a blob-token b_j if $p(w_i|b_j)$ is the maximum in j-th column of probability table.

- Unweighted Matrix (M1) First, we generate $M_{N\times(W+B)}$ by counting the frequency of keywords and blob-tokens.

$$M_{N\times(W+B)} = [M_{N\times W}|M_{N\times B}] = [M_{W1}|M_{B1}] = M_1. \quad (1)$$

  $M_{W1}[i,j]$ is the frequency of j-th keyword which appeared in i-th image. Similarly, $M_{B1}[i,j]$ is the frequency of j-th blob-token that appeared in i-th image.

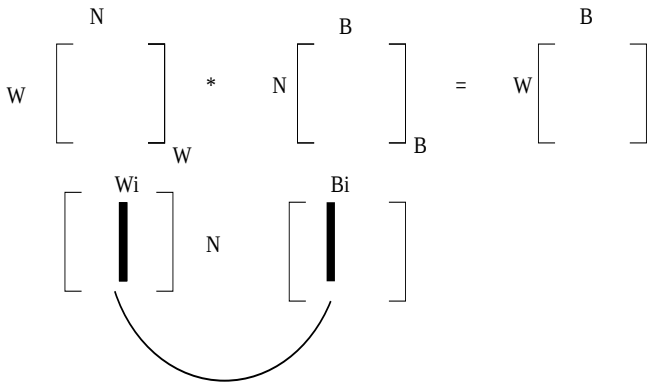

Figure 2: Matrix Multiplication of Words and Blobs

Based on above two matrixes, there are following different models to

calculate probability table.

- Correlation Method (CRM) We use $M_W^T \times M_B$ that gives a matrix with the dimension of W x B and normalize each column to get a probability table T_{corr} based on co-occurrence. $T_{corr}[i,j]$ is an estimate of $p(w_i|b_j)$ which is a conditional probability of keyword w_i given blob b_j.

- Cosine Method (CSM)

  Instead of using $M_W^T \times M_B$, we can apply cosine to calculate the matrix with the dimension of W x B in which the element of ith row and jth column is the cosine between ith column in M_W^T and jth column in M_B. Then, same as CRM, we normalize each column to get a probability table T_{corr}. In fact, correlation method takes into account the following fact: If a keyword appears across a set of images, and a blob also appears in the same set of images, then there is a chance that this blob and keyword are correlated (see Fig. 2).

2.5 Auto-Annotation

To annotate the image automatically, we calculate the distance between the given image object and all centroids of blob-tokens, and represent this image object with the keyword of the closest blob-token. The annotation is generated using keywords assigned to all objects in the image.

3. MEASURING SEMANTIC SIMILARITY

As introduced in Section 1, semantic similarity is very important as a basis for removing noisy keywords and keeping the right keywords. The TM model generates a set of keywords, some relevant and some irrelevant. In order to remove irrelevant keywords, we can measure semantic similarity between various annotated keywords of images. Intuitively, semantically related keywords/concepts will be placed together in a knowledge based as compared to non semantically related keywords. In that case, semantic similarity can help us to determine noisy keywords for an image generated by TM. In Fig. 3, annotated keywords by TM of two images (384008,147066) in Fig. 1. are shown. A set of keywords will provide context/semantic of an image. Note

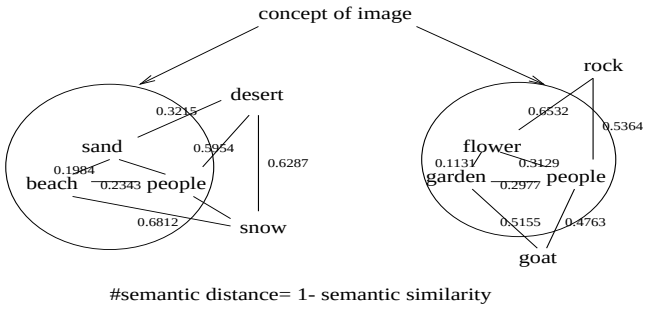

Figure 3: Concept Detection & Noisy words Exclusion within annotation using Semantic Similarity

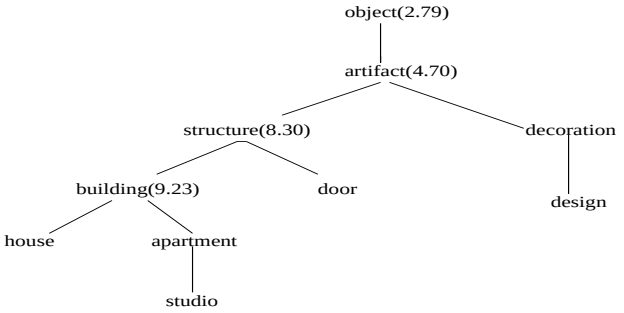

Figure 4: An Example of Information Content in the WordNet

that usually a single keyword is not adequate to represent semantic of an image. For example in Fig. 3, the related keywords in the circle convey some specified concepts ('the people in the beach','the people in the garden') and remove the unrelated keywords that appear outside the circle. Here, the circle of semantic similarity covers relevant concepts of an image. For this, first we will find relevant concepts from annotated keywords in an image. Next, we will measure similarity between these concepts. Finally, some concepts corresponding keywords will be discarded in which total similarity measure of a concept with other concept falls below a certain threshold.

We will use the structure and content of WordNet for measuring semantic similarity between two concepts. Current state of the art can be classified to the three different categories such as: Node-Based Approach[16, 7, 12], Distance-Based Approach[10] and Gloss-Based Approach[2].

In this section, first, we will present various measures to determine semantic similarity between two concepts. Second, we will present drawback of each measure. Finally, we will present hybrid model by fusing these various measures.

3.1 Resnik Measure (RIK)

Resnik et al.[16] introduce first *Information Content (IC)* notion by relying node based approach. More higher value of IC (*Information Content*) means that the concept has specified and detailed information. For example, *cable-television* has more specific information than *television*. RIK first uses Corpus (in our case SemCor) to get the probabilities of each concept and computed how many times the concept appear in the Corpus.

$$freq(c) = \sum_{n \in word(c)} count(n) \tag{2}$$

where word (c) is the set of words subsumed by concept c. Next, the probabilities of each concepts are calculated by the following relative frequency.

$$Prob(c) = \frac{freq(c)}{\tilde{N}} \tag{3}$$

If only one root node is selected, the probability of that node will be 1. This is because root node concept subsumes every concept in WordNet. Second, RIK calculates IC of a concept by taking the negative logarithm of above mentioned probability. Finally, semantic similarity between two concepts will be calculated in the following way. First, RIK determines lowest common subsumer (lcs) between two concepts and then for that lcs concept IC will be determined.

$$IC(concept) = -\log Prob(concept) \tag{4}$$

$$sim(w1, w2) = max_{c1,c2}[sim(c1, c2)] \tag{5}$$

Note that a keyword may be associated with more than one concepts in WordNet. However, the keyword will be associated with a single concept. For example, keyword w1 and w2 are associated with a set of concepts c1 and c2 respectively. Base on that, pair wise similarity between set of concepts c1 and c2 are calculated and keep pair (c1, c2) which yields maximum value. Therefore, word similarity takes into account the maximal information content over all concepts of which both words could be an instance. RIK measure does neither consider the IC value of two concepts/keywords, nor the distance between concepts/keywords in the WordNet. If we consider the similarity between studio and house in Fig. 4, the lcs will be the building and its IC value will be 9.23. However, this value will be the same as the value between house and apartment. This is the weakness of RIK measure.

3.2 Jiang and Conrath Measure (JNC)

Jiang et al.[7] use the same notion of the Information Content and takes into account the distance between selected concepts. In regard to this, JNC combines node-based and edge-base approach. Let us consider the above example. Hence, the two different pair of keywords (*studio* and *house*, *studio* and *apartment*) have the same semantic similarity based on RIK measure. There is no way to discern the semantic similarity between them. However, with regard to semantic similarity between two concepts, JNC uses the IC values of these concepts along with the IC value of lcs of these two concepts. Therefore, the similarity will be different since the IC value of house and apartment are not the same.

$$similarity(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(lcs(c_1, c_2))} \tag{6}$$

3.3 Lin Measure (LIN)

Lin et al.[12] follows the similarity theorem, use the ratio of the commonality and information amounts essential for describing each concept. Commonality between two concepts is the Information Content of lcs. In reality, Lin measure has the close relation of JNC.

$$similarity(c_1, c_2) = \frac{2 * IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (7)$$

3.4 Leacock and Chodorow Measure (LNC)

Leacock et al.[10] measures only between noun concepts by following IS-A relations in the WordNet1.7 hierarchy. LNC computes the shortest number of intermediate nodes from one noun to reach the other noun concept. This is a measurement that human can think intuitively about the semantic distance between two nouns. Unfortunately, WordNet1.7 has a different root node. Therefore, no common ancestor between two keywords can happen. To avoid that, LNC measure introduces the hypothetical root node which can merge multiple-root tree into one-root tree.

$$similarity(c_1, c_2) = max[-\log(ShortestLength(c_1, c_2)/(2*D))] \quad (8)$$

Shortest Length means the shortest path between two concepts. D is the overall depth of WordNet1.7 and a constant value of 16.

3.5 Banerjee and Pedersen Measure(BNP)

Banerjee et al.[2] use the gloss-overlap to compute the similarity. Originally, Gloss-overlaps were first used by [13] to perform word sense disambiguation. The more share their glosses, the more relate two words. BNP not only considers the gloss of target word but also augments with the shared glosses by looking over all relations including hypernym, hyponym, meronym, holonym, troponym. Based on that, BNP measures proliferate their gloss vocabulary. By gathering all glosses between A and B through all relations in WordNet, BNP calculates the similarity between two concepts. If the relations between two concepts are gloss, hyponym, and hypernym,
related-pairs = {(gloss,gloss),(hype,hype),(hypo,hypo), (hype,gloss),(gloss,hype)}

$$similarity(A, B) = \sum_{\alpha \in related-pairs, \beta \in related-pairs} score(\alpha(A) + \beta(B))$$

Here, BNP computes the *score* by counting the number of sharing word and especially if same words appeared consecutively, and assign the score of n^2 where n is the shared consecutive words.

3.6 Comparison of Various Methods

Every measures has some shortcomings. On the one hand, RIK measure cannot differentiate the two keywords which have the same lcs. On the other hand, JNC and LIN address this problem. Their measures give the different similarity value of a pair of keywords having a same ancestor by considering its IC. However, JNC and LIN are sensitive to the Corpus. Based on Corpus, JNC and LIN may end up with different values. Furthermore, LNC measure has additional limitation. For some keywords, SL(ShortestLength)value does not reflect true similarity. For example, furniture will be more closely related with door as compared to sky. However, with LNC, SL for furniture and door and SL for furniture and sky will be 8 in both cases. Due to the structural property of WordNet, it is quite difficult to discriminate between such keywords with LNC. BNP measure relies heavily on shared glosses. If there exists no common word in the augmented glosses by considering every possible relation in WordNet, then this approach will fail to get semantic distance. For example, there is no shared word between glosses of sky and jet , which causes the score between sky and jet is 0.

From the above discussion, it is obvious that we cannot solely rely on a single method. We need to fuse all these measures together to get rid of noisy keywords.

3.7 Applying Semantic Measures for Improving Annotations using Hybrid Measure (TMHD Model)

Here, we propose how we can apply similarity measure to remove unrelated keywords. For this, we rely on the annotated keywords of each image. To remove noisy keywords from each image, we determine correlation between keywords produced by TM model. Intuitively, highly correlated keywords will be kept and non-correlated keywords will be thrown away. For example, annotation for an image by TM model is: *sky, sun, water, people, window, mare, scotland.* Since *scotland* is not correlated with other keywords, it will be treated as noisy keyword. Hence, our strategy will be as follows: First, in an image for each annotated keyword, we determine the similarity score with other annotated keywords appeared in that image based on various methods (JNC, LIN, BNP) discussed in Section 3.1-3.5. Second, we combine these scores for each keyword using Dempster-Shafer Theory. This combined score for each keyword will demonstrate how correlated this keyword with other annotated keywords in that image. Therefore, non correlated keywords will get lower score. Finally, scores of keywords that fall below a certain threshold will be discarded by treating as noisy words. These steps are presented in Fig. 5.

3.8 Dempster-Shafer Evidence Combination

Dempster-Shafer Theory [17] (also known as theory of belief functions) is a mathematical theory of evidence which is considered to be a generalization of the Bayesian theory of subjective probability. Since a belief function rather than a Bayesian probability distribution is the best representation of a chance the Dempster-Shafer theory [1] differs from the Bayesian Theory. A further difference is that probability values are assigned to sets of possibilities rather than single events. Nor does the Dempster-Shafer framework specify priors and conditionals, unlike Bayesian methods which often map unknown priors to random variables The Dempster-Shafer theory is based on two ideas. The first idea is the notion of obtaining degrees of belief for one question based on subjective probabilities for a related question, and Dempster's rule for combining such degree of belief when they are based on independent items of evidence. Since we use independent sources of evidence, namely, JNC and LIN, BNP measure, we are interested in the latter part of the Dempster-Shafer theory, namely, Dempster's rule.

Dempster's Rule is a well known method for aggregating two different bodies of evidence in the same reference set.

λ_i : test images

$\lambda_i = \{\lambda_1, ...\lambda_n\}$

χ_j : annotated keywords of λ_i

$\chi_j = \{\chi_1, ...\chi_m\}$

$SSDT$: Semantic Similarity Distance Table

/*– Detect the unlated keywords in each images –*/

i $\rightarrow$ 1 to Num.images

/*– Compute similarity values for every pairs –*/

j $\rightarrow$ 1 to Num.annotate words

k $\rightarrow$ 1 to Num.annotate words

$similarity_j = \sum_{j \neq k} \text{find\_similarity}(\chi_j, \chi_k)$ in $SSDT$

$sum_i = sum_i + similarity_j$

$\frac{similarity_j}{sum_i} < Threshold \Rightarrow$remove χ_j from ith image

Figure 5: Pseudo Code for removing the noisy keywords

Suppose we want to combine evidence for a hypothesis H. In Semantic Similarity between keywords in each images, H is the assignment of a similarity value between annotated keywords. For example, H is the semantic similarity of 'sky' with other keywords such as 'water', 'mountain', and 'door' in a particular image. H is a member of 2^Θ, i.e., the power set of Θ, where is our frame of discernment. A frame of discernment is an exhaustive set of mutually exclusive elements (hypothesis, propositions). All of the elements in this power-set, including the elements of, are propositions. Given two independent sources of evidence m_1 and m_2, Dempster's Rule combines them in the following frame:

$$m_{1,2}(H) = \frac{\sum_{A,B \subseteq \Theta, A \cap B = C} m_1(A) m_2(B)}{\sum_{A,B \subseteq \Theta, A \cap B \neq \varnothing} m_1(A) m_2(B)} \quad (9)$$

Here A and B are supersets of H, they are not necessarily proper supersets, i.e., they may be equal to H or to the frame of discernment Θ. However, we need to include three different independent sources of evidence m_1, m_2, m_3 are functions (also known as a mass of belief) that assign a coefficient between 0 and 1 to different parts of 2^Θ , so we need another formula to combine three sources as like below,

$$m_{1,2,3}(H) = \frac{\sum_{A,B,C \subseteq \Theta, A \cap B \cap C = H} m_1(A) m_2(B) m_3(C)}{\sum_{A,B,C \subseteq \Theta, A \cap B \cap C \neq \varnothing} m_1(A) m_2(B) m_3(C)} \quad (10)$$

$m_1(A)$is the portion of belief assigned to A by m_1.
$m_{1,2,3}(H)$ is the combined Dempster-Shafer probability for a hypothesis H. To elaborate more about Dempster-Shafer theory, we present the following example.

Example 1

Consider an image that contains three different annotation keywords A, B and C. Each keyword has a semantic distance to other keywords. We are interested in evaluating semantic similarity between the annotated words (i.e., A, B, or C), which will be useful to decide whether each keywords is noisy or not. We may form the following propositions which correspond to proper subsets of Θ :

P_A :The measure will give the similarity dominance for A.

P_B :The measure will give the similarity dominance for B.

P_C :The measure will give the similarity dominance for C.

P_A, P_B :The measure will give the similarity dominance for A or B.

P_B, P_C :The measure will give the similarity dominance for B or C.

P_C, P_A :The measure will give the similarity dominance for C or A.

Each measure would give the the similarity dominance, which is the combined similarity value of a keyword within one image (for this example, A,B,C).With these propositions, 2^Θ would consist of the following:
$2^\Theta = \{\{P_A\}, \{P_B\}, \{P_C\}, \{P_A, P_B\}, \{P_B, P_C\}, \{P_C, P_A\}, \{P_A, P_B, P_C\}, \varnothing\}$
In many applications basic probabilities for every proper subset of Θ may not be available. In these cases a non-zero $m(\Theta)$ accounts for all those subsets for which we have no specific belief. Since we expect the measures (JNC, LIN, BNP) to evaluate semantic dominance about only one keyword at a time (not to calculate the similarity dominance of two different keywords at the same time), we have positive evidence for each keywords only

$$m(\psi) > 0 : \psi \in \{\{P_A\}, \{P_B\}, \{P_C\}\}$$

The uncertainty of the evidence $m(\Theta)$ in this scenario is

$$m(\Theta) = 1 - \sum_{\psi \subset \Theta} m(\psi)$$

In equation (10), the numerator accumulates the evidence which supports a particular hypothesis and the denominator conditions it on the total evidence for those hypotheses supported by all sources.

3.8.1 Using Dempster-Shafer Theory in Removing Noisy Annotation Keywords

We have three sources of evidence: the output of JNC, LIN and BNP, which three different measures already show good performance with the standard data sets. (see the Result Section) Since JNC, LIN, BNP we observed give better result over other method. From now on, we focus on these three methods. If we combine these three different measures into one measure by giving different weights, we need to know the importance of each measure in an image. This may vary from image to image and set of annotations. Furthermore, in one image, JNC would play a main role in discarding noisy keywords; on the other hand,in a another image BNP is very important to remove the noisy keywords there.

Hence, the TMHD model can predict the semantic similarity for a set of keywords in an image by combining Dempster's Rule for three evidences in the following way:

$$m_{JNC,LIN,BNP} = \frac{\sum_{A,B,C \subseteq \Theta, A \cap B \cap C = H} m_{JNC}(A) m_{LIN}(B) m_{BNP}(C)}{\sum_{A,B,C \subseteq \Theta, A \cap B \cap C \neq \varnothing} m_{JNC}(A) m_{LIN}(B) m_{BNP}(C)}$$

In the case of Semantic Similarity Prediction, we can simplify this formulation because we have only belief for singleton classes (i.e., the final prediction is only one keyword) and the body of evidence itself ($m(\Theta)$). This means for any proper subset A of Θ for which we have no specific belief, m(A)=0. For example, based on Example 1 we would have the following terms in the numerator of above formula:

$m_{JNC}(P_B)m_{LIN}(P_B)m_{BNP}(P_B), m_{JNC}(P_B)m_{LIN}(P_B,P_C)$
$m_{BNP}(P_B), m_{JNC}(P_B)m_{LIN}(P_A,P_B)m_{BNP}(P_B)...m_{JNC}(P_B)$
$m_{LIN}(P_\Theta)m_{BNP}(P_B), m_{JNC}(P_A,P_B)m_{LIN}(P_B)$
$m_{BNP}(P_B), ..., m_{JNC}(\Theta)m_{LIN}(P_B)m_{BNP}(P_B)$

Since we have non-zero basic probability assignments for only the singleton subsets of Θ and the Θ itself. This means

$m_{JNC}(P_B)m_{LIN}(P_B)m_{BNP}(P_B) > 0$,
$m_{JNC}(P_B)m_{LIN}(P_B,P_C)m_{BNP}(P_B) = 0$
(since $m_{LIN}(P_B,P_C) = 0$),
$m_{JNC}(P_B)m_{LIN}(P_A,P_B)m_{BNP}(P_B) = 0$
(since $m_{LIN}(P_A,P_B) = 0$),
$m_{JNC}(P_B)m_{LIN}(P_\Theta)m_{BNP}(P_B) > 0$
$m_{JNC}(P_A,P_B)m_{LIN}(P_B)m_{BNP}(P_B) = 0$
(since $m_{JNC}(P_A,P_B) = 0$),
$m_{JNC}(\Theta)m_{LIN}(P_B)m_{BNP}(P_B) > 0$

After eliminating zero terms we get the simplified Dempster's combination rule and we are interested in ranking the hypotheses, we can get further simplified equation where the denominator is independent of any particular hypothesis (i.e., same for all) as follows:

$$m_{JNC,LIN,BEN}(P_B) \propto \sum_{x,y,z \in P_B,\Theta} m_{JNC}(x)m_{LIN}(y)m_{BEN}(z)$$

The $\propto$ is the "is proportional to" relationship. $m_{JNC}(\Theta)$, $m_{LIN}(\Theta)$ and $m_{BNP}(\Theta)$ represent the uncertainty in the bodies of evidence for the $m_{JNC}, m_{LIN}, m_{BNP}$ respectively. For $m_{JNC}(\Theta), m_{LIN}(\Theta)$ and $m_{BNP}(\Theta)$ in the above Equation, we use the following. For each measure, we use the TSD (Total Semantic Distance) values for each measure to compute the uncertainty. Uncertainty is computed based on the TSD of training examples as follows.

$$m_{JNC}(\Theta) = \frac{1}{\ln(e + TSD_{JNC})} \quad (11)$$

TSD_{JNC} is the summation distance of JNC over pairwise keywords within a particular image annotation. For LIN, BNP measures, we will use the similar formula. Here stands for exponential series.

$$TSD_{JNC} = \sum_{i}^{n}\sum_{j}^{n}(1 - JNC_{sim}(i,j))$$
$$n : num.keywords$$

Since we consider the distance which is opposite of semantic similarity from TSD calculation, we subtract JNC similarity value from 1. We can get the TSD values for LIN,BNP as the same way as JNC. $TSD_{JNC} = 2.2087$, $TSD_{LIN} = 2.2875$, $TSD_{BNP} = 5.69211$. If we apply the Equation (11), we can get the uncertainty values of each measures,

$m_{JNC}(\Theta) = 0.29008, m_{LIN}(\Theta) = 0.2956, m_{BNP}(\Theta) = 0.4135$

Table 1: JNC measure values

	Sun	Water	Field	Pillar
Sun	-	0.9691	0.9500	0.6099
Water	0.9691	-	0.9962	0.6893
Filed	0.9500	0.9962	-	0.6722
Pillar	0.6099	0.6893	0.6722	-

Table 2: LIN measure values

	Sun	Water	Field	Pillar
Sun	-	0.7747	0.9902	0.5693
Water	0.7747	-	1.000	0.5805
Filed	0.9902	1.000	-	0.9146
Pillar	0.5963	0.5805	0.9146	-

Example 2
Let us consider a set of annotated keywords of an image by TM. Now, we would like to decide the noisy keywords. The table 1,2,3 has the semantic similarity values between pair of keywords for each Measure. Next, we will determine the semantic dominance of each keyword in the following way.

$m_{JNC}(sun) = 0.3394, m_{LIN}(sun) = 0.1979, m_{BNP}(sun) = 0.0698$
$m_{JNC}(water) = 0.1623, m_{LIN}(water) = 0.2842, m_{BNP}(water) = 0.4007$
$m_{JNC}(field) = 0.3664, m_{LIN}(field) = 0.3642, m_{BNP}(field) = 0.4007$
$m_{JNC}(pillar) = 0.1319, m_{LIN}(pillar) = 0.1537, m_{BNP}(pillar) = 0.1163$

We can get the final combination result from the simplified equation.
$m_{JNC,LIN,BNP}(Sun) = \frac{0.1717}{\sum}$
$m_{JNC,LIN,BNP}(Water) = \frac{0.2578}{\sum}$
$m_{JNC,LIN,BNP}(Field) = \frac{0.4401}{\sum}$
$m_{JNC,LIN,BNP}(Pillar) = \frac{0.0989}{\sum}$

Since the denominator is the same, and we are only interested in the ranking, we can simplify it by the following way.

$m_{JNC,LIN,BNP}(Sun) = 0.177, m_{JNC,LIN,BNP}(Water) = 0.266$

$m_{JNC,LIN,BNP}(Field) = 0.454, m_{JNC,LIN,BNP}(Pillar) = 0.102$

Then, we can remove keywords that below a certain threshold value (for this image, 0.15). Then, Pillar will be treated as noisy keyword and the remaining annotation words are Sun, Water, Field.

Table 3: BNP measure values

	Sun	Water	Field	Pillar
Sun	-	0.4869	0.4998	0.4933
Water	0.4869	-	0.6616	0.3323
Filed	0.4998	0.6616	-	0.4998
Pillar	0.4933	0.3233	0.4998	-

Measure	Num.correct	Num.incorrect	Accuracy
JNC	994	452	67.4%
LIN	855	372	63.6%
LNC	805	562	57.4%
RIK	756	1030	38.7%
BNP	880	700	61.2%

Table 4: With a 50% accuracy test data set

Measure	Num.correct	Num.incorrect	Accuracy
JNC	655	930	58.6%
LIN	778	978	55.6%
LNC	604	990	36.2%
RIK	705	487	40.8%
BNP	650	746	53.4%

Table 5: With a 33% accuracy test data set

4. EXPERIMENT AND RESULTS

The dataset used in this paper is downloaded from [9] which is same as [5].There are 5000 images from 50 Stock Photo CDs in this dataset. Each CD contains 100 images on the same topic. We use 4,500 images as training set and the remaining 500 images as a testing set. The image segmentation algorithm is normalized cut [18]. Each image is represented as a 30 dimensional vector, which corresponds to 30 low-level features. The vocabulary contains 374 different keywords. First, we cluster a total of 42,379 image objects from 4,500 training images into 500 blobs using K-means algorithm and weighted selection method. Second, we apply EM algorithm to annotate keywords for each images automatically. This will be known as the TM model. Finally, we apply hybrid measures (TMHD) to get rid of some noisy annotated keywords. In Fig. 6 we demonstrate the power of the approach, TMHD over our previous approach, TM. For example, one of the example image (108037) includes very unrelated keywords (horses, swimmers) which could make the CBIR system misunderstand the image. After postprocessing, the image not only excludes those noisy keywords, but also keeps 'cat' as annotation. However, 'cat' does not make a difference in understanding the image (108037) semantically. Let us consider the image with identifier 147066. This image has a set of noisy keywords (*beach, coral, crab, nest*). We can see TM generates these noisy keywords and TMHD discards all noisy irrelevant keywords and keep only relevant one. However, if we consider the second image (identifier 17017), TMHD discards irrelevant keywords sky along with relevant keywords, sky and tree. Therefore, this TMHD not only discards irrelevant keywords but also discards occasionally some relevant keywords. Furthermore, it is obvious that if the TM model generates all noisy keywords along with zero relevant keyword for an image, TMHD will not be able to generate correct keywords at all.

4.1 Comparison of Various Measures

Here we would like to demonstrate the power of TMHD over various measures. We report two sets of results based on two accuracy levels (50% and 33%). To make these datasets, initially we select 500 images along with 6 manually annotated correct keywords.

First, we prepare datasets with 50% accuracy in keyword annotation, which means that the ratio of correct and incorrect keywords of an image is 1:1. To get this, we remove three correct keywords from an image and insert three noisy keywords randomly. Similarly, the second dataset has been constructed with 33% accuracy. In Table 4, given the first daatset with 50% accuracy, JNC improves the accuracy to 67.4%. Here, the JNC measure chooses 994 correct keywords out of 1500 keywords and remove 1058 incorrect keywords from 1500 keywords. Furthermore, the JNC,LIN and BNP measures outperform RIK and LNC measures. In Table 5, with dataset 2 (accuracy 33%), accuracy of the JNC, LIN, BNP measures are still greater than 50% even with 67% noisy keywords in images. This demonstrates the power of semantic similarity measures. From these two tables, JNC, LIN and BNP are the best measures regardless of distribution of noisy data. Therefore, in TMHD, we fuse these three(JNC, LIN and BNP) and ignore the other two.

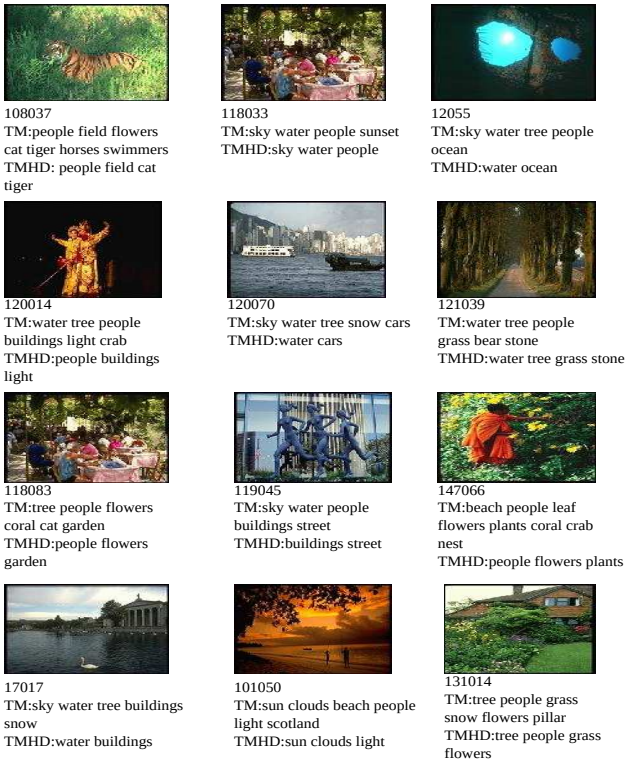

Figure 6: Examples of removing unrelated keywords by Hybrid Measure

4.2 Comparison of TMHD with TM

Here we report results based on most frequently used keywords for TMHD and TM. Recall that TMHD considers hybrid measures. For keyword nest, we observe that precision of TMHD (100%) is substantially higher than precision of TM (12.5%); on the other hand, recall is the same in both cases. This happens due to the removal of only noisy keywords , as no relevant keywords will discarded (i.e., recall is the same). For all these keywords, precision of TMHD has increased as compared to TM to some extent. Note that with the increasing precision recall will be dropped. However, here we observe that, except for the keywords, water, tiger and garden, recall will be the same in both models.

Keywords	TM		TMHD	
	precision	recall	precision	recall
water	0.2482	0.8965	0.5000	0.0431
window	0.1111	0.1250	0.1111	0.1256
plane	0.1428	0.1600	0.1481	0.1600
tiger	0.1428	0.3000	0.5000	0.1000
stone	0.1666	0.3809	0.1702	0.3809
garden	0.0952	0.2000	0.1666	0.1
nest	0.1250	0.1428	1.000	0.1428

Table 6: Performance of Most Frequently Used Keywords for TM and TMHD

Measure	Precision	Recall	VAccuracy
TM	0.2001	0.3501	0.2858
TMHD(JNC+LIN+BNP)	0.3020	0.2116	0.5608
TM+JNC	0.2214	0.1427	0.3718
TM+LIN	0.2007	0.1606	0.3356
TM+LNC	0.2201	0.1408	0.3343
TM+RIK	0.1983	0.1466	0.3340
TM+BNP	0.2230	0.1402	0.3577

Table 7: With a TM data-set, the results of TM and TMHD

On average, precision values of TM and TMHD are 14.21%, and 33.11% respectively. This number demonstrates that TMHD is 56.87% better than TM.

We know that the JNC,LIN and BNP measures generated better results than others. We hybrid these measures(TMHD) by combining them using Dempster-shafer rules. We used the same annotation result of TM. In Fig. 6, we can see the improvement by removing noisy keywords. If an image is not annotated by any relevant keyword, we cannot improve the accuracy since all of the annotation in the image is noisy. To check the accuracy of detecting noisy keywords in an efficient way, we introduce Valid Accuracy (VAccuracy). We define an image as a τ(valid image) if it is annotated with at least one relevant keyword. Note that we do not calculate the accuracy of an image which is associated with all irrelevant keywords. In Table 7, VAccuracy of TMHD is increased substantially as compared to TM only and TM along with LIN, LNC, RIK and BNP. This demonstrates the power of TMHD over individual measures.

$$\lambda = \text{Number of Correct Keywords}$$
$$\chi = \text{Number of InCorrect Keywords}$$
$$\tau = \text{valid image(has at least one correct keyword)}$$

$$VAccuracy = (\lambda \cap \tau)/(\lambda + \chi) \cap \tau \quad (12)$$

5. RELATED WORK

Many statistical models have been published for image retrieval and annotation. Mori et al. [14] use a co-occurrence model, which estimates the correct probability by counting the co-occurrence of words with image objects. Duygulu et al. [5] strived to map keywords to individual image objects. Both treated keywords as one language and blob-tokens as another language, allowing the image annotation problem to be viewed as translation between two languages. Using some classic machine translation models, they annotated a test set of images based on a large number of annotated training images. Based on translation model, Pan et al. [15] propose various methods to discover correlations between image features and keywords. They apply correlation and cosine methods and introduce SVD as well, but the idea is still based on translation model with the assumption that all features are equally important and no knowledge (KB) based has been used. The problem of translation model is that frequent keywords are associated with too many different image segments but infrequent keywords have little chance. To solve this problem, Kang et al. [8] propose two modified translation models for automatic image annotation and achieve better results [8]. Jeon et al. [6] introduce cross-media relevance models (CMRM) where the joint distribution of blobs and words is learned from a training set of annotated images. Unlike translation model, CMRM assumes there is a many to many correlation between keywords and blob tokens rather than one to one. Therefore, CMRM naturally takes into account context information. However, almost all of these proposed models treat all features are equally important and their annotation contains so many noisy keywords. On the other hand, in our case, we apply weighted feature selection and using knowledge based we strive to improve annotation accuracy.

6. CONCLUSIONS AND FUTURE WORKS

The traditional translation model shows limits for matching keywords to the segmented region. In the Corel dataset, the disambiguation between cat and tiger is impossible if we rely on the low-level features (shape, texture, color). To meet the user's request in the CBIR system, the system must detect the semantic understanding of images. One image may have multiple objects and backgrounds. Humans usually read the images semantically through combining the objects in images based on existing knowledge. We used semantic similarity measure using WordNet and removed semantically unrelated keywords, in order for the CBIR system can easily detect the semantic concept of images. Our proposed translation model along with the knowledge based (TMHD model) would get better annotation performance and correspondence accuracy than other traditional translation model. Since traditional translation models annotate too many irrelevant keywords, our model strived to prune irrelevant keywords by exploiting knowledge-based data (here WordNet). During pruning we kept relevant keywords. In our model, we fuse outcomes of all these methods together to make a final decision using Dempster-Shafer Rule. As the result of the test data set, we got more than 50% accuracy after post-processing even if the accuracy of the original annotation is 33%. This is a very meaningful demonstration, which means that the system can overcome the majority of noisy keywords and get the correct semantic understanding of images at the same time.

In future, we would like to extend the work in the following directions. First, we will do more experiments based on different grid analysis, image features, clustering algorithms and statistical models. Next, we would like to extend the work in the video domain.

7. REFERENCES

[1] Y. A. Aslandogan and C. T. Yu. Diogenes: A web search agent for content based indexing of presonal images. *In Proceedings of ACM SIGIR 2000, Athens, Greece*, pages 481–482, July 2000.

[2] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, 2003.

[3] K. Barnard, P. D. N, de Freitas D.and Forsyth D., and B. M.Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[4] D. Blei and M. Jordan. Modeling annotated data. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134, July 2003.

[5] P. Duygulu and K. Barnard. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. *In Seventh European Conference on Computer Vision (ECCV)*, 4:97–112, 2002.

[6] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, July 2003.

[7] J.Jiang and D.Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *In Procedeeings on International Conference on Research in Computational Linguistics*, 1997.

[8] F. Kang, R. Jin, and J. Y. Chai. Regularizing translation models for better automatic image annotation. *CIKM'04*, pages 350–359, 2004.

[9] Kobus. http://www.cs.arizona.edu/people/kobus/research/data. 2002.

[10] C. Leacock. Combining local context and wordnet similarity for word sense identification. *In Christiane Fellbaum, editor, WordNet: A Lexical Reference System and its Application. MIT Press, Cambridge, MA.*, pages 265–283, 1998.

[11] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10), 2003.

[12] D. Lin. Using syntatic dependency as a local context to reslove word sense ambiguity. *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 64–71, 1997.

[13] M.Lesk. Automatic sense disambiguation machine readable dictionaries: How to tell a pine cone from an ice cream cone. *In Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26, 1986.

[14] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. *MISRM'99 Frist International Workshop on Multimedia Intellegent Storage and Retrieval Management*, 1999.

[15] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. *KDD 2004*, pages 653–658, August 2004.

[16] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995.

[17] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Conf. Computer Vision and Pattern Recognition(CVPR)*, pages 731 – 737, June 1997.

[19] R. M. V. Lavrenko and J. Jeon. A model for learning the semantics of pictures. *Proceedings of the 17th Annual Conference on Neural Information Processing Systems*, 2003.

[20] L. Wang and L. Khan. Automatic image annotation and retrieval using weighted feature selection. *To appear in International Journal of Multimedia Tools and Applications by Kluwer Publisher*, 2005.