The Randomized Approximating Graph Algorithm for Image Annotation Refinement Problem

Yohan Jin
University of Texas Dallas
Multimedia Systems Lab.
Richardson, Texas 75083-0688, USA
yohan@utdallas.edu

Kibum Jin
Soongsil University Computer Institute
Dongjak Gu, Sangdo-Dong 511
Seoul, Korea
ckb@ssuci.ac.kr

Latifur Khan
University of Texas Dallas
Data Mining Lab.
Richardson, Texas 75083-0688, USA
lkhan@utdallas.edu

B.Prabhakaran
University of Texas Dallas
Multimedia Systems Lab.
Richardson, Texas 75083-0688, USA
praba@utdallas.edu

Abstract

Recently, images on the Web and personal computers are prevalent around the human's life. To retrieve effectively those images, there are many AIA (Automatic Image Annotation) algorithms. However, it still suffers from low-level accuracy since it couldn't overcome the semantic-gap between low-level features ('color','texture' and 'shape') and high-level semantic meanings (e.g., 'sky','beach'). Namely, AIA techniques annotates images with many noisy keywords. Refinement process has been appeared in these days and it tries to remove noisy keywords by using Knowledge-base and boosting candidate keywords. Because of limitless of candidate keywords and the incorrectness of web-image textual descriptions, this is the time we need to have deterministic polynomial time algorithm. We show that finding optimal solution for removing noisy keywords in the graph is NP-Complete problem and propose new methodology for KBIAR (Knowledge Based Image Annotation Refinement) using the randomized approximation graph algorithm as the general deterministic polynomial time algorithm.

1. Introduction

With the development of digital media and web-technologies, there has been appeared great number of content-based image retrieval (CBIR) researches in last few years such as Co-Occurrence Model [7], Translation Model [12], CRM(cross-media relevance model)[14] and so on. However, for visual similarity, CBIR rely on the low-level features (color histograms, textures, shapes and so on), which leaves a semantic gap between low-level visual features and semantic meaning of images. From this limit, CBIR research still far from reasonable accuracy level for commercial use (There are so many noisy keywords has been annotated along with correct ones). Actually, human understand images based on each person's knowledge beyond image itself. To improve the image annotation performance through imitating the way of human's image understanding, Yohan et al.[1] proposed the first approach for Knowledge-based Image Annotation Refinement (KBIAR) method. Among annotated keywords of each image, It refined image annotation results with removing noisy keywords and proposed semantic distances between annotated keywords (so called, "candidate keywords") for figuring out irrelevant candidate keywords. WordNet, a mirror of world-knowledge, has been used for getting semantic distances between candidate keywords.

Inspired by the Yohan et al's idea, there has been several approaches appeared for refining automatic image annotation problem using the relationship between annotated keywords as 'candidate' keywords by using semantic knowledge, so called KBIAR(Knowledge-Based Image Annotation Refinement) approaches as follows; [3] proposed adaptive graphical model for refining process using fusing visual content feature and keyword correlation and [2] done image annotation refinement by re-ranking the annotations using Random Walk with Restarts algorithm. [4] showed an approach for finding optimal subset annotation keywords of an image by using greedy heuristic solution. There are approaches [8][10] which apply refining methodology into

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

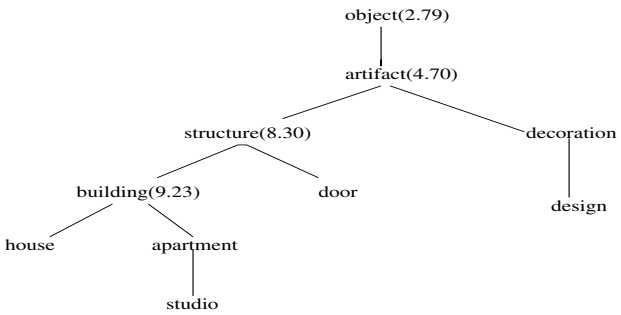


Figure 1. An Example of Information Content in the WordNet

Web-image annotation refinement since there are so many noisy textual description surrounding web-images.

In this paper, we propose a new way of bridging the gap between KBIAR problem and the graph approximation problem. This approach has two important main impacts. First, many previous approaches just use heuristic thresholds for deciding un-related words during re-ranking process. Different from heuristic optimal method [4], which can't show any guaranteed performance, the graph approximation algorithm (especially, weighted max-cut in this paper) can deterministically decide noisy nodes (keywords) as one set with having at least 0.8587 ratio performance to the optimal solution [17].

Second, for the problem of computation complexity, this randomized approximation algorithm can decide irrelevant set in the graph within polynomial time. We show the process of getting semantic similarity values in Section 2. In section 3, from this semantic measure and annotated keywords, we demonstrate that we can build the undirected weight graph and the way of applying the randomized approximation to this weighted max-cut in KBIAR problem. We show that this approach achieves very good accuracy enhancement in section 4 and make conclusion and mention about the future works in section 5.

2. Semantic Similarity

2.1 WordNet

WordNet[16] is the knowledge-base which is the mirror of world knowledge. Each words in WordNet has hierarchical relations, thus we can use this tree-structured knowledge-base as the source for computing semantic distance between two keywords. Semantic similarity between two concepts will be calculated in the following way; the first concept is IC (Information Content) which calculates the amount of information that a word has. For example, in Fig.1, 'building' has less information than 'house' and 'building' (9.23) has more detail information than 'structure' (8.30). (-smaller value of IC means that the word has less information). Formally, we can get IC value with using following simple equations;

$$freq(c) = \sum_{c_i \in c} count(c_i) \tag{1}$$

Here, '$\in$' stands for the "subsuming" relationship, to get the frequency of word c, it summations all the count of each word c_i, which is subsumed by word c. So, the IC of concept c can be computed by taking negative logarithm of relative probability ($Prob(c) = \frac{freq(c)}{N}$).

$$IC(Concept\ c) = -logProb(Concept\ c) \tag{2}$$

There are mainly three different approaches for computing semantic distances from the WordNet and yohan et al.[1] showed that JNC [15] measure's performance is the best among other similar measures. Let us consider an example of comparison between $\alpha(\text{"}studio\text{"} \leftrightarrow \text{"}apartment\text{"})$ and $\beta(\text{"}studio\text{"} \leftrightarrow \text{"}house\text{"})$. If we only consider the information content of subsuming word ("building") - so called, lcs (lowest common subsumer), then the semantic distance between α and β is the same. Thus, JNC measure uses the IC values of two words along with the IC value of lcs. The similarity measure between α and β is different since the IC value of "house" and "apartment" is not the same. The formula of similarity value $\lambda(c1, c2)$ between two words(concepts) like this;

$$\lambda(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(lcs(c_1, c_2))} \tag{3}$$

2.2 Co-Occurrence

We use the Apriori algorithm [11] for finding co-occurrence probability, which is based on the idea of level-wise search. The level–wise search is an iterative approach in which, (*m+1*)–itemsets are explored based on the previous *m*–itemsets. At first, the 1-itemset (L_1) is found, then each *i*–itemsets (L_i) is used to find *i+1*–itemsets (L_{i+1}) until no more frequent *k*–itemsets can be found. In our paper, we choose frequent sets until 2-itemset since we only consider a pair occurrence. So, when we compute the co-occurrence($\mu(w_i, w_j)$ between w_i, w_j by dividing the frequency $\Psi(w_i \cap w_j)$ in L_2 set by the frequency $\Psi(w_i)$ in L_1 set.

$$\begin{aligned} \mu(w_i, w_j) &= P(w_i \rightarrow w_j) \\ &= P(w_j | w_i) \\ &= \frac{\Psi(w_i \cap w_j) \in L_2}{\Psi(w_i) \in L_1} \end{aligned}$$

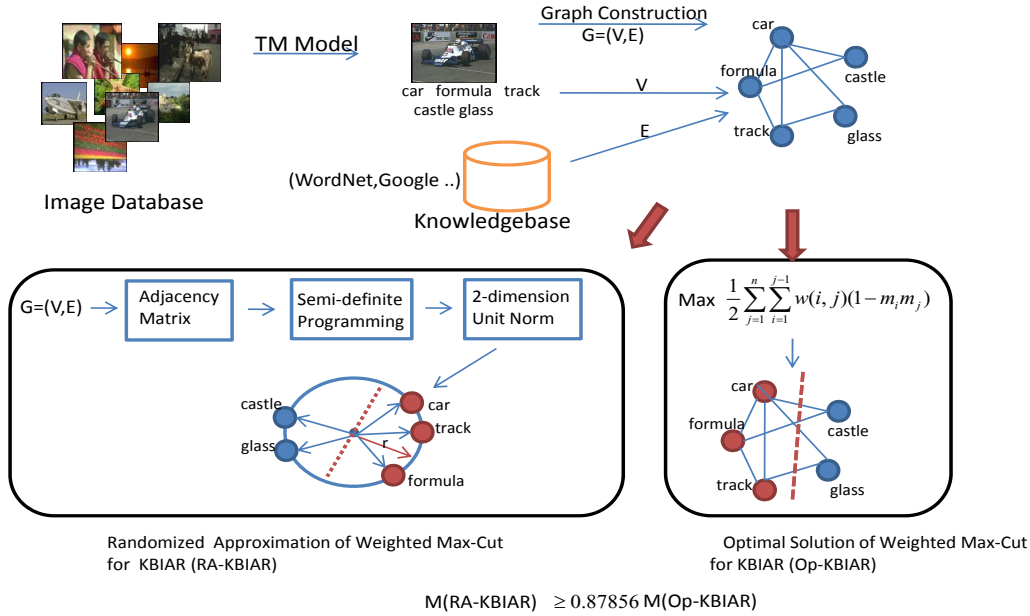


Figure 2. Framework for KBIAR (Knowledgebase Image Annotation Refinement) Through Randomized Approximation of Weighted Maximum Cut Problem

3. Approximated Graph Algorithm in KBIAR Problem

3.1 Reduction From KBIAR to Weighted Max-Cut

KBIAR Problem
Input: Annotated Set $A_k = \{ak_1, ak_2, ..., ak_n\}$, $SDM_{n\times n}$
Output: $max\{s1, s2\}$, where $s1, s2 \subset A_k$ and $s1 \cap s2 = \emptyset$

WMC Problem
Input: a graph G=(V,E) and weight function w(i,j)
Output: Partition V1,V2 of V

We can transform each keyword ak_i to a vertex v_i of Graph G by one to one mapping ($f : ak_i \rightarrow v_i$). For each edge $E(i, j)$ between v_i and v_j, there is an corresponding element in the semantic distance matrix (SDM). Through a mapping function ($g : SDM(i, j) \rightarrow w(i, j)$), we can get the weight function of Graph G. Thus, we can reduce an instance of KBIAR problem into an instance of WMC problem in polynomial time. It makes possible to solve weighted max-cut problem for getting optimal solution of KBIAR problem. MAX-CUT is one of the Karp's [9] original problem. Usually, we can see the polynomial-time reduction from NotAllEqual-3SAT to MAX-CUT problem for showing MAX-CUT's NP-Completeness. (thus, transitivity is follows; MAX-3SAT$\leq_p$NotAllEqual-3SAT$\leq_p$MAX-CUT). Thus, getting the optimal solution with the Weighted MAX-CUT (which is reduced from KBIAR) is NP-Completeness problem.

3.2 Optimal solution of WMC Problem

Let us represent WMC problem as integer quadratic problem (WMC-IQP);

Maximize

$$\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{j-1} w(i,j)(1 - m_i \times m_j) \tag{4}$$

Subject to $m_i \in \{-1, 1\}$, $1 \leq i \leq n$, where $n = |V|$

m_i is the membership binary values, in that, if adjacent vertices m_a, m_b are belong to different set $V1, V2$ respectively by the current cut, then the membership values for each m_a, m_b will be different (-1,1). So, 1- $m_a \times m_b$ value is 2, if m_a, m_b are in same set, then 1- $m_a \times m_b$ value is 0, then

its weight value doesn't count as the total weight of a cut instance. The instance that makes the maximum total weight of cut would be optimal partition V1,V2 of the graph. If we find the maximum cut in non-deterministic way, then we can guess an assignment of each vertex's set and compute the optimal value of the above IQP (Integer Quadratic Problem). To do this thoroughly (namely, check with every possible combinations), we need exponential amount of time (2^n). To find max-cut in polynomial time, we need an approximation scheme for MAX-CUT problem.

3.3 Randomized Approximation WMC (Weighted Max-Cut) in KBIAR

Our work is based on Goeman's randomized 0.87856 approximation scheme for finding maximum-cut that is constructed with each image's candidate annotation keywords and semantic similarity between those words. Goeman et al.[17] showed the way of relaxing from integer quadratic problem to semi-definite programming by increasing the dimensions of membership value m_i from 1 to n dimensions and constructing a matrix M such that $M_{i,j}$ is corresponding to each inner product $m_i \bullet m_j$. To make a problem more tractable with graphs of real-values weights, it associates matrix M with Laplacian matrix $L \Leftarrow Diag(W \bullet e)$. Thus, we can start to run the randomized algorithm with weight adjacency matrix (W) of graph G.

Algorithm3.1 Randomized Approximation Algorithm for WMC-KBIAR Problem

Input: An image k with candidate annotation keywords
$A_k = \{ak_1, ak_2, ..., ak_n\}$, $\mu^{n\times n}$,$\lambda^{n\times n}$
Begin:
Build a weighted adjacency matrix,
$W(i,j) = \lambda(i,j) \times \mu(i,j)$
Using Laplacian matrix $L \Leftarrow Diag(W \bullet e)$,
where e is a vector of all ones, and $M \leftarrow \frac{1}{4}m^T Lm$,
Construct Semi-Definite(SD) programming;
Maximize $tr(LM)$
Subject to $diag(M) = \frac{1}{4}e, M \geq 0$
Run the SDP and get the optimal solution M^*
decompose M^* into 2× n matrix B, ($M^* = B^T B$)
Hyper-plane Separation
for i=1 to n
$\phi = b_i \bullet r$, b_i is a column vector of B
if $\phi \geq 0$, then $b_i \in S1$, otherwise, $b_i \in S2$
end
Major Annotation Set Decision
If $\sum_{i,j\in S1} W(i,j) \geq \sum_{i,j\in S2} W(i,j)$,
then return S1, Otherwise, return S2
Output:
A refined majority keywords set = $max\{S1, S2\}$
among partitioned two set S1, S2

building palace people
crystal anemone reef

Figure 3. Annotation Results with correct and noisy candidates keywords

3.4 Example

Now, we explain how the algorithm 3.1 works by applying an example to this randomized approximation algorithm (see Fig.3). This image has been annotated with 6 keywords. As we can see, there are relative candidate keywords ('building','palace' and 'people') and un-related ones ('crystal','anemone' and 'reef').

In Fig.4, If we utilize the optimal solution using integer quadratic programming (see Eq.4), then, we have to check the total sum of weights with each assignment (m_i=-1 or 1). Namely, one of assignments can be 'building(-1)','palace(1)','people(1)','crystal(-1)','anemone(1)','reef(1)'. Other than this example, we have to compute additional $2^6 - 1$ all possible combination of assignments. But there is no limit of the number of description keywords in an image, so that is why we need to prepare approximating methodology to deal with general cases, which can be finished in polynomial time. Through relaxations, we can transform the WMC-KBIAR problem from integer-quadratic to semi-definite programming. In other words, we can solve the problem with using fixed edge-values instead of assigning membership integer(m_i) variables. For the edge weight values between two keywords (vertices), we already get those values from semantic similarity measure and co-occurrence.

At this moment, we can observe that this randomized max-cut approximation algorithm is quite compatible with our KBIAR (Knowledge-Based Image Annotation Refinement). We can directly use this randomized semi-definite algorithm for solving our WMC-KBIAR problem (see Fig.4). For the initial adjacency matrix construction, we multiply the co-occurrence value $\mu(i,j)$ and semantic similarity value $\lambda(i,j)$ between two nodes of keywords in the graph.

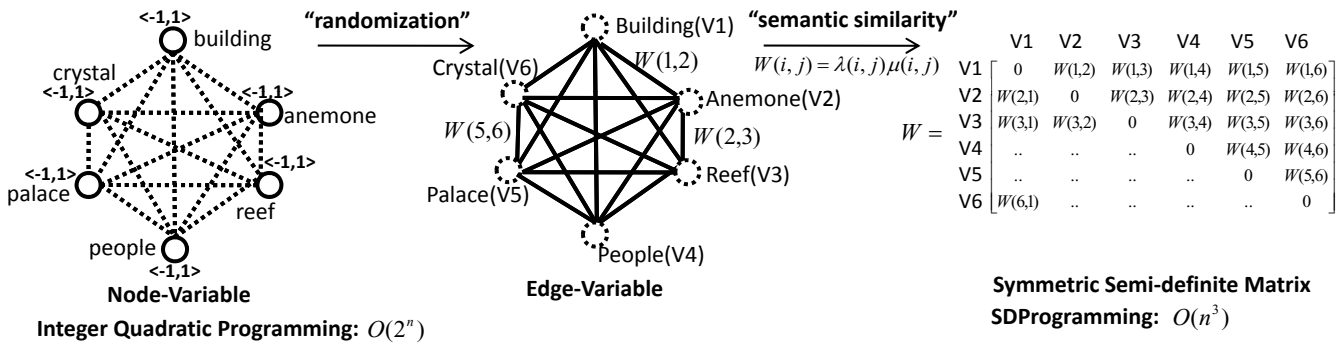


Figure 4. Randomization of WMC-KBIAR Problem through relaxations

$$W = \begin{bmatrix} 0 & 0.8326 & 0.4863 & 0.8700 & 0.9998 & 0.8962 \\ 0.8326 & 0 & 0.7667 & 0.9003 & 0.9998 & 0.9165 \\ 0.4863 & 0.7667 & 0 & 0.8336 & 0.9998 & 0.8743 \\ 0.8700 & 0.9003 & 0.8336 & 0 & 0.9998 & 0.8959 \\ 0.9998 & 0.9998 & 0.9998 & 0.9998 & 0 & 0.9998 \\ 0.8962 & 0.9165 & 0.8743 & 0.8959 & 0.9998 & 0 \end{bmatrix}$$

$$L \Leftarrow diag(W \bullet e)$$
$$M \leftarrow \frac{1}{4} m^T L m$$

Optimal solution to SDP

$$M^* = \begin{bmatrix} 1 & 0.9976 & 0.9999 & -0.8586 & -0.9757 & -0.9806 \\ 0.9976 & 1 & 0.9967 & -0.8918 & -0.9583 & -0.9648 \\ 0.9999 & 0.9967 & 1 & -0.8524 & -0.9783 & -0.9829 \\ -0.8586 & -0.8918 & -0.8524 & 1 & 0.7256 & 0.7415 \\ -0.9757 & -0.9583 & -0.9783 & 0.7256 & 1 & 0.9997 \\ -0.9806 & -0.9648 & -0.9829 & 0.7415 & 0.9997 & 1 \end{bmatrix}$$

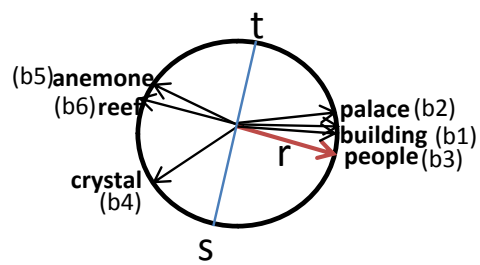
$$M^* = B^T B$$

Incomplete Cholesky Decomposition

$$B = \begin{bmatrix} 1 & 0.9976 & 0.9999 & -0.8586 & -0.9757 & -0.9806 \\ 0 & 0.0686 & -0.0121 & -0.5124 & 0.2189 & 0.1959 \end{bmatrix}$$

Figure 5. Optimal solution of SD Programming and decompose to 2-dimensional matrix

$$B = \begin{bmatrix} 1 & 0.9976 & 0.9999 & -0.8586 & -0.9757 & -0.9806 \\ 0 & 0.0686 & -0.0121 & -0.5124 & 0.2189 & 0.1959 \end{bmatrix}$$



$$S1 = \{b1, b2, b3\} \quad \forall i, b_i \in S1, b_i \bullet r \geq 0$$
$$S2 = \{b4, b5, b6\} \quad \forall j, b_j \in S2, b_j \bullet r < 0$$

Figure 6. 2-dimensional Mapping into the Random-hyper plane for the Cut decision

The current matrix W, which is a symmetric matrix since the semantic distance between W(1,2)=(V1-building,V2-crystal) and W(2,1)=(V2-crystal,V1-building) are the same. After associating the Laplacian matrix $L \Leftarrow Diag(W \bullet e)$,$M \leftarrow \frac{1}{4} m^T L m$, we have a semi-definite matrix M and get the optimal solution M^* by running SD Programming (see Fig.5). Since a semi-definite matrix M can be decomposed into $Y^T Y$ [19], we can decompose the optimal result M^* into a row-reduced matrix B ($M^* = B^T B$, where B is the size of $m \times n$) using Incomplete Cholesky Decomposition. Usually, m is much smaller than n ($m \ll n$) and this algorithm chooses m=2 (see Algorithm3.1 , Fig.5).

Each column vector, b_i is corresponding to each node in the graph G. Now, these 2-dimensional values of each node can be mapped into 2-dimensional unit-norm space[17] since every b_i is unit-norm ($\| b_i \| = 1$). This 2D random-hyper unit-norm space can be used for making a maxi-

mum cut decision within each undirected weighted graph G. Here, the algorithm randomly selects a 2-dimensional unit-norm vector r (Fig.6) from uniform distribution and computes inner product with each b_i vectors.

Finally, if the inner product value ($b_i \bullet r$) is ≥ 0, then the current node (keyword) is assigned to S1 set, otherwise, S2 set (see Fig.6). With this example, building, palace and people has been classified to S1 set and crystal, anemone and reef to S2 set. To find the majority set (Actually, max-cut doesn't tell us which set is a major set, but it just divides into two sets), it computes summation of weighted adjacency matrix, which is from computing semantic similarity and chooses the maximum set as the majority set -If $\sum_{i,j \in S1} W(i,j) \geq \sum_{i,j \in S2} W(i,j)$, then return S1. (here, we have 'building','palace' and 'people' as the 'refined annotation' for this example image -Fig.3). Goemans et al [17] proved that the randomized approximation of weighted max-cut has 0.87856 ratio with the optimal solution (Eq.4), since we already show that the KBIAR problem can be reformulated into weight max-cut problem, the WMC-KBIAR algorithm (algorithm3.1) also has same performance ratio.

4. Implementation & Performance Analysis

We use the Corel Dataset[13] for image database, which has 5,000 images. 4,500 images are used for training set and remaining 500 images for test set. For the semantic similarity measure, we used the WordNet library[16] for implementing JNC measure algorithm. For implementation of randomized approximation of weight-max-cut algorithm, we utilized the SDTP3 Matlab software[18] for computing semi-definite programming part in the whole approximation algorithm. To see the effect of refinement in the various environments, namely, we change the ratio of noisy keywords using synthetic annotation data. Our refinement methodology using randomized approximation of weighted max-cut can improve the accuracy in any ratio of noisy keywords, in that; it enhanced the accuracies to 54.39%, 74.39% and 82.01% from 42.01%, 60.02% and 72.34% accurately annotated set respectively (see Fig.8). As this refinement algorithm can remove noisy keywords with different situation (every web-page include right description about image on that page along with some irrelevant description words), it means that we can also apply this methodology for disambiguating irrelevant keywords description in the web-image description refinement process [6], which is very crucial process for web-image annotation problem[5]. In Fig.9, we can see that our proposed max-cut based refinement algorithm can improve the refinement accuracy and outperform other approaches. In terms of precision accuracy, it achieves 35%, which is the best among the original TM precision accuracy (20%), yohan et al [1] 's TMHD method (30%) and Liu et al [5]'s AGAnn method (24%). The increasing of precision accuracy means that user will see more correct images within retrieval results since precision accuracy is from dividing the number of correctly annotated images by the number of retrieved images. Although recall value of any refinement algorithms is worse than original TM result, the F-measure value (which is the combination of precision and recall values) of (TM+max-cut)-27.12% is also better than any other methods (TM-25.45%, AGAnn-26.26%). If we give 2 times more importance to the precision value when we combine precision and recall values, then F-0.5 measure value enhancement from TM-23.33% to (TM-max-cut)-29.34% is greater than before.

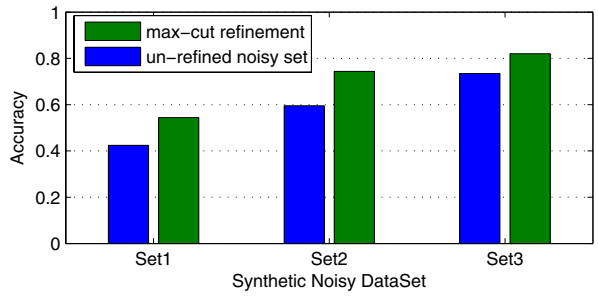


Figure 8. Accuracy enhancement through weighted max-cut refinement with synthetic noisy dataset

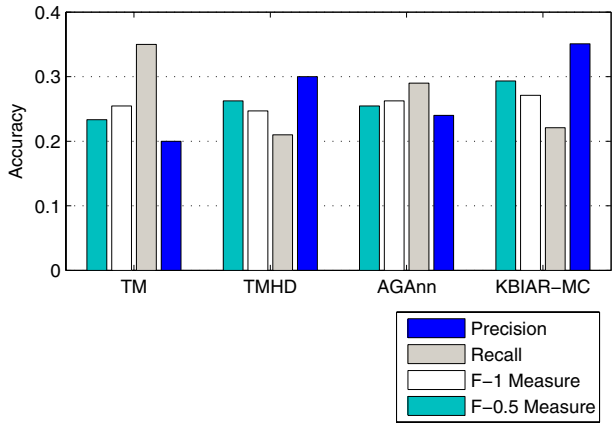


Figure 9. Refinement Performance Comparison between several Approaches

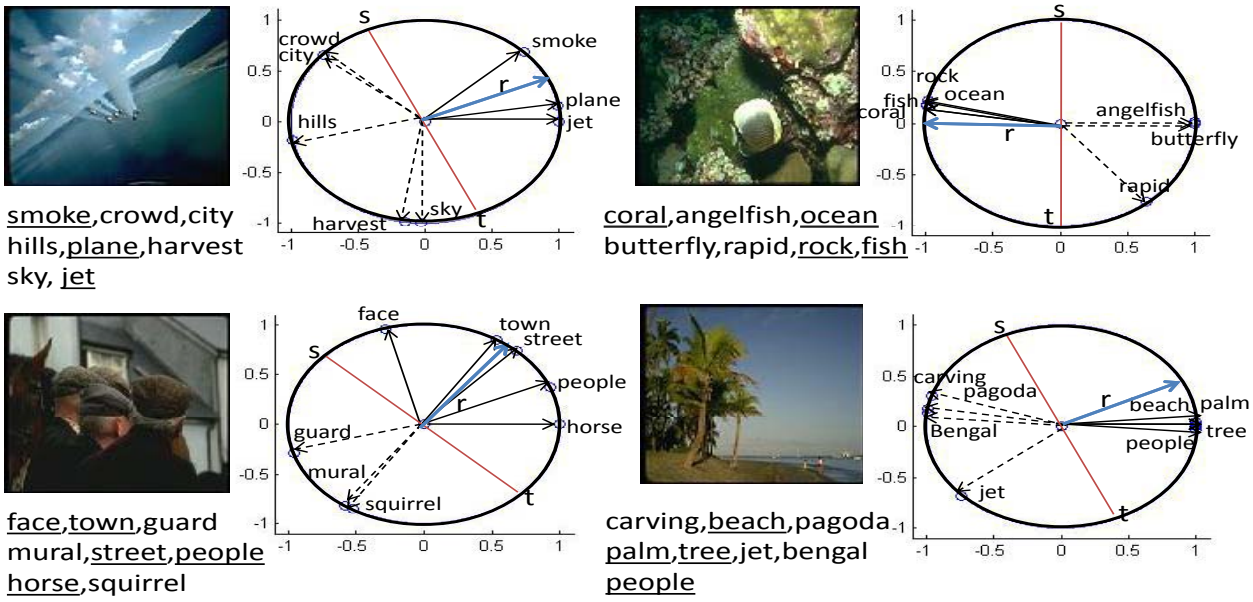


Figure 7. Result Examples of Random Space Separation from Max-Cut Refinement algorithm

5. Conclusions & Future Work

From the problem analysis and proposal of applying randomized weighted-max cut algorithm to image annotation refinement problem, we can see that this novel approach can get the benefit of using the approximation algorithm of weighted max-cut in terms of two following aspects; First, to combine expert knowledge for good understanding about uncertain dataset (usually, it may be "multi-media" dataset), we need a framework which would be suitable for mixing the different sources. In this paper, one of the sources is the TM model, which is kind of machine-learning algorithm. Another one is the semantic distance between result keywords of TM model. After mapping keywords and semantic distance as vertex and edges respectively, the constructed graph can represent local conceptual relations between several keywords, so weighted max-cut algorithm can exclude one set of un-related nodes as noisy keywords set decisively. Thus, this method can increase its performance as other resources' performances improve, namely, the accuracy of weighted max-cut will increase according to the better accuracy of initial keyword translation algorithm (for vertex mapping) and semantic similarity algorithm (for edge-value mapping).

Second, the approximation algorithm about the weighted max-cut is quite required. If we find an optimal max-cut by guessing an integer assignment (-1,1) of each keyword corresponding to vertex, then the required number of times to decide the maximal cut is 2^n, whereas the complexity of randomized approximation version of weighted max-cut is O(n^3) . When the keywords about an image are only several ones, then the time-complexity is not so different, however, if we want to include more description since the knowledge source is more than a single one, then the size of keywords in an image increases. For the web-image annotation problem, usually, the keywords surrounding a web-image could be more than 100. In this case, the computing time about this single image refinement is greater than ($2^{100} \simeq 10^{30}$), which is actually intractable computation time. Consequently, the multimedia content analysis area needs to adapt new methodology since there is rapid change in multimedia content itself and its surrounding knowledge-base. This paper show the very good possibility for adapting approximation graph algorithm to combining multiple knowledge-evidence sources for increasing the accuracy of image-annotation and keep the computational time within polynomial bound in the complex knowledge source environment.

References

[1] Yohan, Jin,L. Khan, L. Wang, and Awad, M, , "Image Annotations By Combining Multiple Evidence and WordNet", *In Proceedings the 13th annual ACM international conference on Multimedia (MM05')*, 2005, Singapore, 706 - 715.

[2] Changhu Wang, Feng Jing, Lei Zhang, Hong-Jiang Zhang, "Image Annotation Refinment using Random Walk with Restarts", *Proceedings of the 14th annual ACM international conference on Multimedia, MM*

06', October 23-27, Santa Barbara, California, USA. 647-650, 2006.

[3] Jing Liu, Mingjing Li, Wei-Yang Ma, Qingshan Liu, Hanqing Lu, "An Adaptive Graph Model for Automatic Image Annotation", *Proceedings of the 8th ACM international workshop on Multimedia information retrieval (MIR 06'), Santa Barbara, California, USA, October 26-27, 61-70, 2006*,

[4] Xiangdong Zhou, Mei Wang, Qi Zhang, Junqi Zhang, Baile Shi, "Automatic Image Annotation By An Iterative Approach: Incorporating Keyword Correlations And Region Matching", *Proceedings of the 6th ACM international conference on Image and video retrieval, (CIVR 07')*, July 9-11, Amsterdam, The Nethelands , 425-432, 2007.

[5] Xiaoguang Rui and Mingjing Li and Zhiwei Li and Wei-Ying Ma and Nenghai Yu "Bipartite graph reinforcement model for web image annotation", *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, 585–594,Augsburg, Germany.

[6] Xin-Jing Wang, Lei Zhang, Feng Jing, Wei-Ying Ma, "AnnoSearch: Image Auto-Annotation by Search", *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,1483-1490, 2006.

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

[8] Xiaoquang Rui,Nenghai Yu, Taifend Wang and Mingjing Li, "A Search-Based Web Image Annotation Method", *Proceedings of Multimedia and Expo, 2007 IEEE International Conference on 2007, July 2-5, 655-658*

[9] R.Karp, ""Reducibility among combinatorial problems", *Plenum Press, pages 85-103, 1972*.

[10] Bin Wang,Zhiwei Li and Mingjing Li, "Automatic Refinement of Keyword Annotations for Web Image Search", *Int'l Proceedings of "ACM Multimedia Modeling Conference (MMM) 2007, January 9-12, 259-268, Singapore*

[11] Agrawal, R., and Srikant, R. , "Fast Algorithms for Mining Association Rules", *In Proceedings of the 20th International Conference Very Large Data Bases, (VLDB)*, Santiago, Chile, September 1994,

[12] Duygulu, P., Barnard, K., De Freitas, N., Forsyth, D, "Object recognition as machine translation: learning a lexicon for a fixed image vocabulary", *In Proceedings of the Seventh European Conference on Computer Vision (ECCV) Part IV*, Copenhagen, Denmark, May 28-31, 2002, 97–112.

[13] Corel DataSet, "http://kobus.ca/research/data/eccv_2002/index.html", 2002.

[14] Jeon, J., Lavrenko, V., and Manmatha, R., 2003. Automatic Image Annotation and Retrieval using Cross–Media Relevance Models. In *Proceedings of the 26th Annual International ACM SIGIR Conference*, Toronto, Canada, 2003, 119–126.

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

[16] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., AND Miller, K, "WordNet: An on-line lexical database", *International Journal of Lexicography, 3(4)*, 1990, 235-244.

[17] Goemans, M. X., and Williamson, D. P., "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming", *Journal of ACM, 42, 1115-1145*

[18] K. C. Toh, M. J. Todd, and R. H. Tutuncu, "SDPT3– a MATLAB software package for semidefinite programming", *Technical Report TR1177, Cornell University, December 1996*.

[19] Vijay V.Vazirani, "Approximation Algorithms", "Springer, 2001".