Automatic Multimedia Cross-modal Correlation Discovery*

Jia-Yu Pan Hyung-Jeong Yang† Christos Faloutsos Pinar Duygulu‡
Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213, U.S.A.
{jypan, hjyang, christos}@cs.cmu.edu, duygulu@cs.bilkent.edu.tr

ABSTRACT

Given an image (or video clip, or audio song), how do we automatically assign keywords to it? The general problem is to find correlations across the media in a collection of multimedia objects like video clips, with colors, and/or motion, and/or audio, and/or text scripts. We propose a novel, graph-based approach, "MMG", to discover such cross-modal correlations.

Our "MMG" method requires no tuning, no clustering, no user-determined constants; it can be applied to *any* multimedia collection, as long as we have a similarity function for each medium; and it scales linearly with the database size. We report auto-captioning experiments on the "standard" Corel image database of 680 MB, where it outperforms domain specific, fine-tuned methods by up to 10 percentage points in captioning accuracy (50% relative improvement).

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Design, Experimentation

*This material is based upon work supported by the National Science Foundation under Grants No. IIS-0121641, IIS-0083148, IIS-0113089, IIS-0209107, IIS-0205224, INT-0318547, SENSOR-0329549, EF-0331657, IIS-0326322, and by the Pennsylvania Infrastructure Technology Alliance (PITA) Grant No. 22-901-0001. Additional funding was provided by donations from Intel, and by a gift from Northrop-Grumman Corporation.

†Supported by the Post-doctoral Fellowship Program of Korea Science and Engineering Foundation (KOSEF)

‡Pinar Duygulu's current address: Department of Computer Engineering, Bilkent University, Ankara, Turkey, 06800

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

Keywords

Cross-modal correlation, automatic image captioning, graph-based model

1. INTRODUCTION AND RELATED WORK

Given a collection of multimedia objects, we want to find correlations across media. The driving application is auto-captioning, where the problem is defined as follows:

PROBLEM 1 (AUTO-CAPTIONING). *Given a set $\mathcal{S}$ of color images, each with caption words; and given one more, uncaptioned image I, find the best t (say, t=5) caption words to assign to it.*

However, the method we propose is general, and can be applied to video clips (with text scripts, audio, motion); on audio songs, with text lyrics, and so on.

PROBLEM 2 (INFORMAL-GENERAL). Given n *multimedia objects, each consisting of* m *attributes (traditional numerical attributes, or multimedia ones such as text, video, audio, time-sequence, etc).* Find *correlations across the media (eg., correlated keywords with image blobs/regions; video motion with audio features).*

For example, we want to answer questions of the form *"which keywords show up, for images with blue top"* or *"which songs are usually in the background of fast-moving video clips"*.

We assume that domain experts have provided us with similarity functions for all the involved media. The similarity function does not need to be perfect and is sufficient to our needs if it could approximately identify the neighboring objects of an object.

There are multiple research papers, attacking parts of the problem. For example, to associate words with images for *automatic image captioning*, people have proposed methods based on classifiers [15] or information retrieval techniques (relevance model [10] and user feedback [26]), as well as building association models (translation model [7]; hierarchical model [2, 3, 4]; multi-resolution model [13]; co-occurrence model [16]). Video databases [25] spark efforts to associate script words with faces [20], and visual/auditory characteristics with video genres (news or commercial) [18]. Similarly, there are successful efforts [24]. to associate songs with their genres (like jazz, classical, etc.). Previous correlation discovery attempts such as LSI [19] and SDD [12] mostly consider categorical attributes. In this work, we proposed a general methods which consider both categorical and numerical attributes, as well as set-valued attributes.

We would like to find an unifying method, with the following specifications:

- it should be domain independent
- it should spot correlations in any of the above scenarios, with missing values, feature vectors, set-valued attributes, and all combinations thereof.
- it should scale up for large collections of objects, both with respect to training, as well as for responses.

In the following, we define the problem and describe our proposed method in Section 2 and 3, respectively. Section 4 gives experimental results on real data. We discuss our observations in Section 5. Section 6 gives the conclusions.

2. PROBLEM DEFINITION

We proposed a novel approach for cross-media correlations, and we use image captioning as an illustration. Table 1 shows the terminology we used in the paper. The problem is more formally defined as follows:

PROBLEM 3 (FORMAL). Given *a set* $\mathcal{S}$ *of* n *multimedia objects* $\mathcal{S}=\{O_1, O_2, \ldots, O_n\}$ *, each with* m *multimedia attributes,* find *patterns/correlations among the objects and attributes.*

Symbol	Description
	Objects
O_i	the i-th training object
O_q	the query object
A_i	the i-th attribute of an object
$V(O_i)$	the vertex of G_{MMG} for object O_i.
I_i	the i-th training image
I_q	the query image
	Matrix/vector
$\mathbf{A}$	the (column-normalized) adjacency matrix
$\vec{\mathbf{v_q}}$	the restart vector of the query object (all zeros, except a single '1')
$\vec{\mathbf{u_q}}$	the steady state probability vector for the $\vec{\mathbf{v_q}}$ restart vector

Table 1: Summary of symbols used in the paper

We need to elaborate on the attributes: In traditional RDBMSs, attributes must be *atomic* (i.e., taking single values, like "ISBN", or "video duration"). However, in our case, they can be *set-valued*, like a set of caption words, or, even missing altogether. Take problem 1 (auto-captioning) for example, $\mathcal{S}$ is a collection of captioned images, and we want to guess the (missing) caption terms of a new, uncaptioned image. One attribute of an image is the "caption", which is set-valued (a set of words).

We propose to gear our method towards set-valued attributes, because they include atomic attributes as a special case; and they also smoothly handle the case of missing values (null set). Thus, we only talk about set-valued attributes from now on.

DEFINITION 1. *The* domain D_i *of (set-valued) attribute* i *is the collection of atomic values that attribute* i *can choose from. The values of domain* D_i *will be referred to as the* domain tokens *of* D_i.

ASSUMPTION 1. *For each domain* D_i $(i = 1, \ldots, m)$*, we are given a similarity function* $s_i(*, *)$ *which assigns a score to each pair of domain tokens.*

A domain can consist of categorical values, numerical values, or numerical vectors. For problem 1 (auto-captioning), we have objects of m=2 attributes. The first, "caption", has as domain a set of categorical values (English terms); the second, "image regions", is a set of p-dimensional numerical feature vectors (p=30, as we describe next). The similarity function among the caption tokens could be 1 if the two tokens are identical, and 0 otherwise; the similarity function for "regions" could be, say a function of the Euclidean distance between feature vectors. Let's elaborate on image captioning, before we present the main idea.

2.1 Case Study: Automatic Image Captioning

In the driving example of auto-captioning, the objects of interest are images. Each image has a set of regions extracted from the image content, and some of them also have a caption. See Figure 1 for the 3 sample images, their captions and their regions.

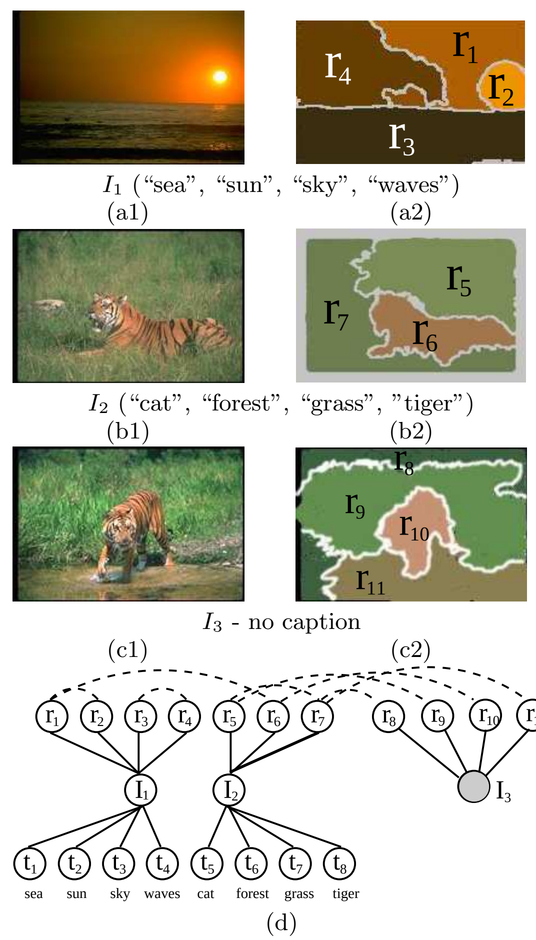


Figure 1: Three sample images, two of them annotated ((a1),(b1)), and their regions ((a2),(b2),(c2)); and their "MMG"graph (d). (Figures look best in color.)

For fair comparison, we use the same data used in the previous work[2, 7]. Image regions are extracted by a standard segmentation algorithm[22] (see Figure 1(d,e,f)). Each region is then mapped into a 30-dim feature vector. The p=30 features extracted from each region are the mean and standard deviation of RGB values, average responses to various texture filters, its position in the entire image layout, and shape descriptors (e.g., major orientation, or bounding region to real region area ratio)[7, 8]. Note that the exact feature extraction details are *orthogonal* to our approach - all our "MMG"method needs is a black box that will map each color image into a set of zero or more feature vectors.

Thus, we turned specific auto-captioning problem (Problem 1) into the general form (Problem 2) as follows: Given n image objects, each with $m = 2$ set-valued attributes: "regions", a set of numerical feature vectors; "captions", a set of categorical terms, find correlations across modalities (regions and words). For example, *which caption terms are more likely, when the image has a blue, sky-like blob.*

The question is what to do next, to capture cross-media correlations. Should we use clustering on feature vectors, or should we use some classification method, as it has been suggested before? And, if yes, how many cluster centers should we shoot for? Or, if we choose classification, which classifier should we use? Next we show how to handle, and actually, bypass, all these issues, for *any* multimedia setting.

3. PROPOSED METHOD - MIXED MEDIA GRAPH ("MMG")

The main idea is to represent all the objects, as well as their attributes (domain tokens) as nodes in a *graph*. For multimedia objects with m attributes, we obtain an $(m+1)$-layer graph G_{MMG}. There are m types of nodes (one for each attribute) and one more type of nodes for the objects. Next we describe (a) how to generate this G_{MMG} graph and (b) how to estimate cross-modal correlations using G_{MMG}.

Graph construction. See Figure 1 for an example. We will denote as $V(O)$ the vertex of object O, and as $V(a_i)$ the vertex of the attribute value $A = a_i$. We put an edge between the node of an object and the nodes of its attributes.

There is only one subtle point: For numerical and vector attributes, we need a way to reflect the similarity between two attribute token values. Our approach is to add an edge if and only if the two token values are close enough. For example, the orange "tiger" region r_6 and the orange sky region r_1 have feature vectors that are closed in Euclidean distance, and therefore, $V(r_1)$ and $V(r_6)$ are connected by an edge.

We need to decide on a threshold for the "closeness". There are many ways, but we decided to make the threshold adaptive: for each feature-vector, choose its k nearest neighbors, and add the corresponding edges. We discuss the choice of k later, as well as the sensitivity of our results to k. Computing the nearest neighbors is straight-forward, because we already have the similarity function $s_i(*, *)$ for any domain D_i (Assumption 1).

In summary, we have two types of links in our "MMG"graph: the *nearest neighbor links* (*NN-links*), between the nodes of two similar domain tokens; and the *object-attribute-value links* (*OAV-links*), between an object node and an attribute value node.

Example 1. *Consider the image set* $\mathcal{S}=\mathcal{I}=\{I_1, I_2, I_3\}$ *(Figure 1). The graph corresponds to this data set has three types of nodes: one for the image objects* i_j*'s* $(j = 1, 2, 3)$*; one for the regions* r_j*'s* $(j = 1, \ldots, 11)$*, and one for the terms* $\{t_1, \ldots, t_8\}$*=*$\{$*sea, sun, sky, waves, cat, forest, grass, tiger*$\}$*. Figure 1(g) shows the resulting "MMG"graph. Solid arcs indicate the object-attribute-value (OAV-links) relationships; dashed arcs indicate nearest-neighbor (NN-links) relationships.*

In this example, we consider only k=1 nearest neighbor, to avoid cluttering the diagram. Note that nearest neighbor relationship is not symmetric, and we treat the edges as un-directional, which make some nodes have degree greater than 1 (for example, node $V(r_1)$: r_2's nearest neighbor is r_1, but r_1's nearest neighbor is r_6).

To solve the auto-captioning problem (Problem 1), we need to develop a method to find good caption words for the uncaptioned image (e.g., image I_3). This means that we need to estimate the affinity of each term to the uncaptioned image (i.e., the affinity of nodes $t_1, \ldots, t_8$ to node i_3). We discuss this next.

Correlation discovery by random walk. We propose to turn the multimedia problem into a graph problem. Thus, we can tap the sizable literature of graph algorithms, and use off-the-shelf methods for assigning importance to vertices in a graph, as well as determining how related is an un-captioned image (represented by node, say "A" in the graph), to the term "tiger" (represented, say, by node "B" in the graph).

We have many choices: electricity based approaches [17, 6], random walks (PageRank, topic-sensitive PageRank) [5, 9], hubs and authorities [11], and elastic springs [14]. In this work, we propose to use *random walk with restart* ("RWR") for estimating the affinity of node "B" with respect to node "A". But, again, the specific choice of method is orthogonal to our framework.

The "random walk with restarts" operates as follows: to compute the affinity of node "B" for node "A", consider a random walker that starts from node "A". The random walker chooses randomly among the available edges every time, except that, before he makes a choice, with probability c, he goes back to node "A" (restart). Let $u_A(B)$ denote the steady-state probability that our random walker will find himself at node "B". Then, $u_A(B)$ is what we want, the affinity of "B" with respect to "A".

Definition 2 (Affinity). *The importance of node B with respect to node A is the steady-state probability* $u_A(B)$ *of random walk with restarts, as defined above.*

For example, to solve the auto-captioning problem for image I_3 of Figure 1, we can estimate the steady-state probabilities $u_{i_3}(*)$ for all nodes of the graph G_{MMG}, we can keep only the nodes that correspond to terms, and we can report the top few (say, 5), as caption words for I_3.

Algorithms. For the general problem, the algorithm is as follows: First, build the "MMG" graph G_{MMG}. When the user asks for the affinity of node "B" to node "A", estimate the steady-state probability $u_A(B)$ defined above.

The computation of the steady-state probabilities is very interesting and important. We use matrix notation, for compactness. Let O_q be the query object (e.g., image I_3 of Figure 1). Suppose that we want to find the most related terms to O_q. We do an RWR from node $q = V(O_q)$, and compute the steady state probability vector $\vec{\mathbf{u_q}}=(u_q(1), \ldots, u_q(N))$, where N is the number of nodes in the graph G_{MMG}.

The estimation of vector $\vec{\mathbf{u_q}}$ can be implemented efficiently by matrix multiplication. Let $\mathbf{A}$ be the adjacency matrix of the graph G_{MMG}, and let it be column-normalized. Let $\vec{\mathbf{v_q}}$ be a column vector with all its N elements zero, except for the entry that corresponds to node q; set this entry to 1. We call $\vec{\mathbf{v_q}}$ the "restart vector". Now we can formalize the definition of the "affinity" of a node (Definition 2).

Definition 3 (Steady-state vector). *Let c be the probability of restarting the random walk from node q Then, the N-by-1 steady state probability vector, $\vec{\mathbf{u_q}}$, (or simply, steady-state vector) satisfies the equation:*

$$\vec{\mathbf{u}}_q = (1-c)\mathbf{A}\vec{\mathbf{u}}_q + c\vec{\mathbf{v}}_q. \qquad (1)$$

The pseudo code of finding cross-modal correlations is shown in Figure 2. Let E be the number of edges in the graph built from the data set. The computational cost per iteration (step 4.1) is O(E), for there are $2E$ non-zero elements in the matrix $\mathbf{A}$ which are involved in the matrix multiplication. If we set a constant maximum number of iterations to be executed before convergence, the overall cost for Algorithm-CCD is O(E), linear to the data set size. Building the graph G_{MMG} is a one time cost, and can be done efficiently using a good nearest-neighbor index (e.g., R+-tree [21]) over the objects.

Given a G_{MMG} graph and an object O_q.

1. Let $\vec{\mathbf{v_q}}$=0, for all its N entries, except a '1' for the q-th entry.
2. Normalize the adjacency matrix of G_{MMG}, $\mathbf{A}$, by column. That is, make each column sum to 1.
3. Initialize $\vec{\mathbf{u_q}}=\vec{\mathbf{v_q}}$.
4. while($\vec{\mathbf{u_q}}$ has not converged)
   4.1 $\vec{\mathbf{u_q}}$ = (1-c)$\mathbf{A}\vec{\mathbf{u_q}}$ + c$\vec{\mathbf{v_q}}$ (*)

Figure 2: Algorithm-CCD: Cross-modal correlation discovery

4. EXPERIMENTAL RESULTS

In this section, we show experimental results to address the following questions:

- Quality: How does the proposed "MMG" method perform on captioning test images?
- Parameter defaults: How to choose good default values for the k and c parameters?
- Generality: How well does "MMG" capture other cross-media correlations: for example, how well does it solve the reverse problem (given a term like, "sky", find the regions that are likely to correspond to it). Similarly, how well does "MMG" capture same-media correlations (say, term-term, or region-region correlations)

In our experiment, we use 10 image data sets from Corel, which is also used in previous works [2, 7]. On the average, each set has 5200 captioned images (each with about 4 captioned terms), 1740 test images, 160 terms for captioning. One "MMG" graph is constructed for each data set, each of which has about 55,500 nodes and 180,000 edges.

Quality. For each test image, we compute the captioning accuracy as the percentage of terms which are correctly predicted. For a test image which has m correct caption terms, "MMG" will predict also m terms. If p terms are correctly predicted, then the captioning accuracy for this test image is defined as $\frac{p}{m}$. This metric is also used in previous works [2, 7].

Figure 3 shows the average captioning accuracy for the 10 data sets. We compare our results with the results reported in [7] which builds a statistical translation model using expectation-maximization (EM) to capture correlation. We refer to the method as the "EM" approach. On the average, on the same data used in [7], "MMG" achieves captioning accuracy improvement of 12.9 percentage points, which corresponds to a relative improvement of 58%. The parameters are set at k=3 and c=0.8, but as shown later, the performance is insensitive to specific settings.

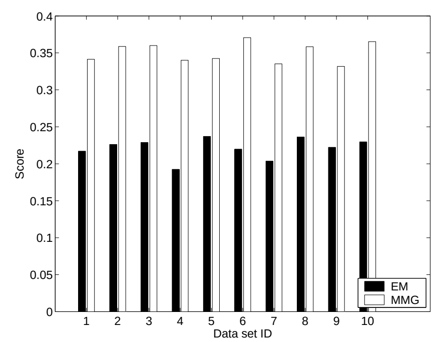


Figure 3: (EM and "MMG") The parameters for "MMG" are $c = 0.8$, and $K = 3$.

We also compare the captioning accuracy with even more recent machine vision methods [2]: the Hierarchical Aspect Models method ("HAM"), and the Latent Dirichlet Allocation model ("LDA"). Figure 4 compares the best average captioning accuracy reported by the two methods (HAM and LDA), with those of the proposed "MMG" method. The same data sets are used for fair comparison. Although both HAM and LDA improve on the EM method, they both lose to our generic "MMG" approach (35%, versus 29% and 25%). We do not compare per data set accuracy due to the lack of such measurements from [2], but we note that "MMG" gives significantly lower performance variance over the 10 data sets, by roughly an order of magnitude: 0.002 versus 0.02 and 0.03.

Parameter defaults. We experiment to find out how would different values of the parameters c (restart probability) and k (nearest neighbor size) affect the captioning accuracy. Figure 5 shows how the captioning accuracy of "MMG" varies with different parameter settings: (a) fixed k=3, vary c; (b) fixed c=0.9, vary k. The plateaus shown in the both plots indicate that the accuracy of the proposed "MMG" is insensitive to the specific setting of k and c. We show only the

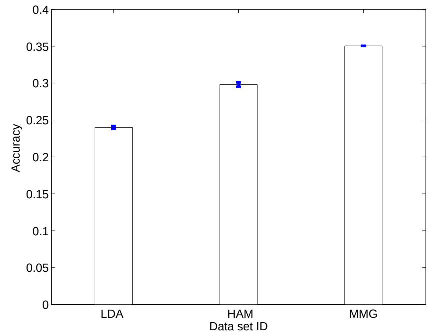
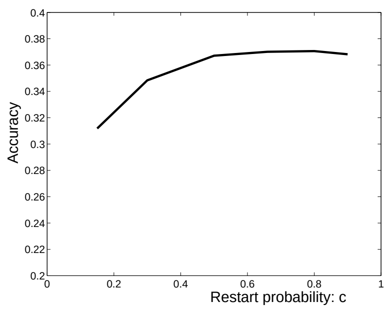
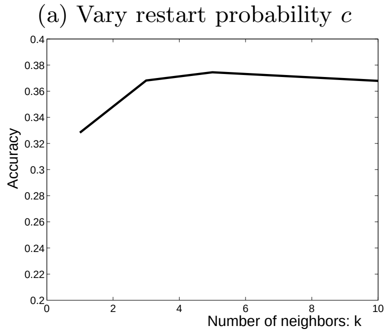


Figure 4: Comparing "MMG" with LDA and HAM on the 10 Corel data sets. LDA: (μ, σ^2)=(0.24,0.002); HAM: (μ, σ^2)=(0.298,0.003); MMG: (mean, variance)=(0.3503, 0.0002). μ: accuracy mean. σ^2: accuracy variance.



(a) Vary restart probability c



(b) Vary nearest neighbor size k

Figure 5: Plateaus in the plots show that the captioning accuracy is insensitive to values of c and k. Data set "006". c: restart probability, k: nearest neighbor size. (a) Fix k=3, vary c. (b) Fix c=0.9, vary k.

result on one data set "006", the results on other data sets are similar.

Generality. "MMG" works on objects of any types. We design a experiment of finding similar caption terms using "MMG". We use the same graph G_{MMG} constructed for automatic image captioning. To find the similar terms of a caption term t, we do "RWR", restarting from the node $V(t)$. Table 2 shows the similar terms found for some of the caption terms. In the table, each row shows the caption term in question at the first column, followed by the top 5 similar terms found by "MMG" (sorted by similarity degree).

Notice that the retrieved terms make a lot of sense: for example, the string 'branch' in the caption is strongly related to forest- and bird- related concepts ("birds", "owl", "night"), and so on. Notice again that we did nothing special: no tf/idf, no normalization, no other domain-specific analysis - we just treated these terms as nodes in our "MMG", like everything else.

Term	1	2	3	4	5
branch	birds	night	owl	nest	hawk
bridge	water	arch	sky	stone	boats
cactus	saguaro	desert	sky	grass	sunset
car	tracks	street	buildings	turn	prototype
f-16	plane	jet	sky	runway	water
market	people	street	food	closeup	buildings
pillars	stone	temple	people	sculpture	ruins
reefs	fish	water	ocean	coral	sea

Table 2: Similar terms of selected caption terms

5. DISCUSSION

We are shooting for a method that requires no parameter tuning. Thus, here we discuss how to choose defaults for both our parameters, the number of neighbors k, and the restart probability c.

Number of neighbors k. In hindsight, the results of Figure 5 make sense: with only k=1 neighbor per region, the collection of regions is disconnected, missing important connections and thus leading to poor captioning performance. On the other extreme, with a high value of k, everybody is directly connected to everybody else. The region nodes form almost a clique, which does not distinguish clearly between really close neighbors, and just neighbors.

For a medium number of neighbors k, our NN-links apparently capture the neighbors they should. Small deviations from that value, make little difference, probably because the extra neighbors we add, are at least as good as the previous ones.

Restart probability c. For web graphs, the recommended value for c is typically c=0.15 [23]. Surprisingly, our experiments show that good quality is achieved for c=0.8 or 0.9. We conjecture that what determines a good value for the "restart probability" is the *diameter* of the graph. For the web graph, the diameter is approximately d=19 [1] which implies that the probability $p_{periphery}$ for the random walker to reach a node in the periphery is roughly $(1-0.15)^{19}$=0.045. If we demand the same $p_{periphery}$ for the three-layer graph (diameter is roughly 3) for captioning, then we have

$$(1-0.15)^{19} = (1-c)^3 \quad (2)$$

$$\Rightarrow c = 0.65 \quad (3)$$

which is much closer to our empirical observations. Of course, the problem requires more careful analysis - but we are the first to show that c=0.15 is not always optimal for random walks with restarts.

6. CONCLUSIONS

We started from the image auto-captioning problem, and we developed "MMG", a general method that can spot correlations across media. The proposed graph-based model can be applied to diverse multimedia data to find multi-modal correlations. Moreover, to the best of our knowledge, this is the first effort in multimedia databases, that proposes such a graph-based approach to find patterns and correlations across media. The method has the following desirable characteristics:

- It is domain independent - the $s_i(*, *)$ similarity functions completely isolate our "MMG" method from the domain.
- It does not require tedious parameter tuning (in contrast to linear/polynomial/kernel SVMs, k-means clustering, etc.). We give good default values for the only 2 parameters k and c, and show empirically that the performance is not too sensitive to them.
- Specifically applied for image auto-captioning, it provides excellent results on real image data sets (680 MBytes), outperforming finely tuned, domain-specific methods (up to 10% absolute, 50% relative, accuracy improvement).
- It is fast, scaling up well with the database size.

Future work could further exploit the promising connection between multimedia databases and graph algorithms, that we propose here: imputation of missing values, outlier detection and any other data mining task that require the discovery of correlations as its first step.

7. REFERENCES

[1] A. Albert, H. Jeong, and A.-L. Barabasi. Diameter of the world wide web. *Nature*, 401:130–131, 1999.

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

[3] K. Barnard, P. Duygulu, and D. A. Forsyth. Clustering art. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 434–441, 2001.

[4] K. Barnard and D. A. Forsyth. Learning the semantics of words and pictures. In *Int. Conf. on Computer Vision*, pages 408–15, 2001.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.

[6] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Kluwer.

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

[8] C. Faloutsos. *Searching Multimedia Databases by Content*. Kluwer, 1996.

[9] T. H. Haveliwala. Topic-sensitive PageRank. In *WWW2002*, May 7-11 2002.

[10] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *26th Annual International ACM SIGIR Conference*, July 28-August 1, 2003, Toronto, Canada.

[11] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[12] T. G. Kolda and D. P. O'Leary. A semidiscrete matrix decomposition for latent semantic indexing information retrieval. *ACM Transactions on Information Systems*, 16(4):322–346, 1998.

[13] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling app roach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):14, 2003.

[14] L. Lovasz. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:353–398, 1996.

[15] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *The Fifteenth International Conference on Machine Learning*, 1998.

[16] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[17] C. R. Palmer and C. Faloutsos. Electricity based external similarity of categorical attributes. In *PAKDD 2003*, May 2003.

[18] J.-Y. Pan and C. Faloutsos. VideoCube: a novel tool for video mining and classification. In *Proceedings of the Fifth International Conference on Asian Digital Libraries (ICADL 2002)*, 2002.

[19] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *PODS 98*, 1998.

[20] S. Satoh, Y. Nakamura, and T. Kanade. Name-it: Naming and detecting faces in news videos. *IEEE Multimedia*, 6(1), January-March 1999.

[21] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R+-tree: A dynamic index for multi-dimensional objects. In *12th International Conf. on VLDB*, pages 507–518, Sept. 1987.

[22] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[23] Taher Haveliwala, S. Kamvar and G. Jeh. An analytical comparison of approaches to personalizing PageRank. Technical report, Stanford University, 2003.

[24] G. Tzanetakis and P. Cook. MARSYAS: A framework for audio analysis. *Organized Sound*, 4(3), 2000.

[25] H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann. Lessons learned from the creation and deployment of a terabyte digital video library. *IEEE Computer*, 32(2):66–73, February 1999.

[26] L. Wenyin, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski, and B. Field. Semi-automatic image annotation. In *INTERACT2001, 8th IFIP TC.13 Conference on Human-Computer Interaction*, Tokyo, Japan July 9-13, 2001.