AUTOMATIC DE-NOISING FOR IMAGE ANNOTATION USING LATENT SEMANTIC ANALYSIS

Abbass.A.Khorsheed¹, Hussein Chible¹, Giorgio Giacinto², Ammar J. Fattah³

¹Doctoral School of science and Technology, Lebanese University, Hadtah, Beirut, Lebanon
²Department of Electrical and Electronic Engineering, University of Cagliari, Italy
³CEO Science Gate Online Research Center.

ABSTRACT

Images are important material accessed through the internet by a huge number of applications such as medical, social, mining applications. The biggest challenge facing the usage of those billion of images is the retrieving challenge. Two approaches are available to retrieve images over the internet: first one is by using textual matching between user query and image annotation, and second one is by using image contents.

This paper introduces a novel approach to remove redundant words used to annotate images; this is done by using Latent Semantic Analysis (LSA) to build the semantic space that combines queries and annotations, and then use Singular Value Decomposition (SVD) to determine variance produced by annotation words. As a last step, words with less variance are omitted.

Keywords: Image Annotation, LSA, SVD, Automatic De-Noising, Semantic Space, Singular Values

1- INTRODUCTION

From the inspection of popular image search engines such as Google, Bing and Baidu, the retrieval paradigm employed by these search engines is still based on the keywords composing the query; this query is formulated by users to initiate image search process. Users use natural language words to describe requested image, or other multimedia contents, and the responsibility of a search engine is to scan databases for a proper match. The most crucial element is the search scenario is the indexing of images, or other multimedia contents, where natural language is demanded to achieve the labeling of available images with textual description; this process is called image annotation [1,2].

Content-based image retrieval, the problem of searching large image repositories according to their content, has been the subject of a significant amount of computer vision research in the recent past. While early retrieval architectures were based on the query-by-example paradigm, which
formulates image retrieval as the search for the best database match to a user-provided query image, it was quickly realized that the design of fully functional retrieval systems would require support for semantic queries. These are systems where the database of images are annotated with semantic keywords, enabling the user to specify the query through a natural language description of the visual concepts of interest. This realization, combined with the cost of manual image labeling, generated significant interest in the problem of automatically extracting semantic descriptors from images [1,2,3].

Images are annotated using different methodologies, some are manually; this when clients comment on certain images and automatically such as mining the textual text in internet pages that hold that image. Crucial challenge in image annotation is the redundant words that increase false results such as the irrelevant images returned by Google search engine [3].

The earliest efforts in the area were directed to the reliable extraction of specific semantics, e.g. differentiating indoor from outdoor scenes, cities from landscapes, and detecting trees, horses, or buildings, among others. These efforts posed the problem of semantics extraction as one of supervised learning: a set of training images with and without the concept of interest was collected and a binary classifier trained to detect the concept of interest. The classifier was then applied to all database of images which were, in this way, annotated with respect to the presence or absence of the concept [2,3].

More recently, there has been an effort to solve the problem in its full generality, by resorting to unsupervised learning. The basic idea is to introduce a set of latent variables that encode hidden states of the world, where each state defines a joint distribution on the space of semantic keywords and image appearance descriptors (in the form of local features computed over image neighborhoods). After the annotation model is learned, an image is annotated by finding the most likely keywords given the features of the image [1, 2, 3].

2- LATENT SEMANTIC ANALYSIS (LSA)

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the meaning of words. Meaning is estimated using statistical computations applied to a large corpus of text [4].

The corpus embodies a set of mutual constraints that largely determine the semantic similarity of words and sets of words. These constraints can be solved using linear algebra methods, in particular, singular value decomposition [4].

LSA has been shown to reflect human knowledge in a variety of ways. For example, LSA measures correlate highly with humans’ scores on standard vocabulary and subject matter tests; it mimics human word sorting and category judgments; it simulates word-word and passage-word lexical priming data; and it accurately estimates passage coherence [4, 5].

The core processing in LSA is to decompose A using SVD (Singular Value Decomposition); SVD has designed to reduce a dataset containing a large number of values to a dataset containing significantly fewer values, but which still contains a large fraction of the variability present in the original data [3, 4, 5].

\[ A = U \Sigma V^T \]  ------eq.1

Where

1- \( \text{EigenVector}(AA^T) \rightarrow \text{Columns}(U) \)
2- \( \text{EigenVector}(A^T A) \rightarrow \text{Columns}(V) \)
3- \( \text{EigenValue}(A^T A) \text{ OR } \text{EigenValue}(AA^T) \rightarrow \Sigma \)
the first structure is the single pattern that represent the most variance in the data, after all, SVD is an orthogonal analysis for dataset. U is composed of eigenvectors of the variance-covariance matrix of the data, where the first eigenvector points to the direction which holds the most variability produced by all other vectors jointly. U is an orthogonal matrix where all its structures are mutually uncorrelated. Eigen values are representing scalar variance of corresponding eigenvectors; this way total variation exhibited by the data is the sum of all eigenvalues and singular values are the square root of the eigenvalues [4, 6].

3- TEXTUAL IMAGE INDEXING AND RETRIEVAL

In 1970s, the conventional image retrieval system used keyword as descriptors to index an image however the content of an image is much richer than what any set of keywords can express [2].

Text-based image retrieval techniques employ text to describe the content of the image which often causes ambiguity and inadequacy in performing an image database search and query processing. This problem is due to the difficulty in specifying exact terms and phrases in describing the content of images as the content of an image is much richer than what any set of keywords can express. Since the textual annotations are based on language, variations in annotation will pose challenges to image retrieval [2, 5].

4- HYPOTHESIS

Hypothesis 1: Latent Semantic Analysis (LSA) reduces the redundant annotation of an image by truncating less variant key words of the annotation.

Hypothesis 2: variation in variance-covariance natural language semantic space is analogues to visual semantic space.

5- THE PROPOSED IMAGE INDEXING AND RETRIEVAL

In this proposal images are represented by concepts it hold. Image concept is the projection of human interpretation to the visual structures within an image, hence:

\[ I = \sum_{i=1}^{N} C_i \cdot \vec{v}_i \quad \text{--- eq.2} \]

Where \( I \) is any image and \( C_i \) is the \( i^{th} \) concept recognized with that image

\[ q = \sum_{i=1}^{K} w_i \cdot \vec{u}_i \quad \text{---eq.3} \]

Where \( q \) is the query entered by the user, \( w_i \) is the \( i^{th} \) word within the query and \( \vec{u}_i \) is the semantic unit vector. Semantic meaning for image’s concept should correlate human’s interpretation for that concept; hence, eq.3 is a prerequisite

\[ \vec{v}_i \cdot \vec{w}_i = 1 \quad \text{--- eq.4} \]

The semantic space is represented by a \( A_{m \times n} \) matrix and this matrix is decomposed into its principal components as the following equation:

\[ A = U\Sigma V^T = \sum_{i=1}^{N} \sigma_i u_i v_i^T \quad \text{---eq.5} \]
Where \( \sigma_i \) is the \( i^{th} \) singular value of the matrix, \( \sigma_1 \) and \( \nu_1^T \) are the most effective direction

Block similarity is measured by the following formula:

\[
\theta = \cos^{-1} \frac{\nu_i \nu_j}{|\nu_i| |\nu_j|} \quad \text{--- (5)}
\]

Where

\[
\text{Min} (\theta) < \text{Threshold} \rightarrow \nu_i \equiv \nu_j \quad \text{--- (6)}
\]

The priorities of using specific word to index and retrieve certain image is corresponding to the singular values calculated by the SVD algorithm, this way words with less singular values can be omitted from the annotation.

\( \Sigma \) matrix can be used as a noise filter where queries are treated as vectors within the semantic space and those who are on the same direction toward the most singular value; those queries would composed of the most effected words.

**Example:**

To demonstrate the effect of the proposal hypothesis, real queries have been posted through Google search engine and textual annotations for some of the return images have been extracted. The extracted annotations and posted queries have been used to build the semantic space required by LSA, after that SVD algorithm has been applied to find out what direction holds the maximum variation, as the following presents:

S1: instead-of-mowing-grass-the-plains-man-wins-car
S2: Oregon_state_police_investigating_fatal_car_crash_west_of_valley
S3: pb_man_lying_on_grass
S4: free_ems_mini_plant_cut_hair_man_grass_doll
S5: vin_diesel_actor_man_car_wheel_serious_bald
S6: two_people_car_race_arrested_grass
Q1: car_man_grass
Q2: car_crash_race

LSA is applied to the annotations and the query to construct the semantic space matrix as it is presented in figure (3):

<table>
<thead>
<tr>
<th>I</th>
<th>Query</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>Q1</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Man</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Grass</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Crash</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Race</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 3:** Semantic Space of LSA based on word repetition in Annotation
The analysis steps are shown below:

\[
\begin{align*}
U &= \begin{pmatrix}
-0.5891 & 0.4065 & -0.0000 & 0.1621 & 0.6793 & 0.0000 & 0.0000 & 0.0000 \\
-0.6135 & -0.5192 & 0.0000 & 0.4895 & -0.3382 & 0.0000 & 0.0000 & 0.0000 \\
-0.4837 & 0.4274 & -0.0000 & -0.5317 & -0.5483 & 0.0000 & 0.0000 & 0.0000 \\
-0.1459 & -0.4373 & 0.7071 & -0.4751 & 0.2485 & 0.0000 & 0.0000 & 0.0000 \\
-0.1459 & -0.4373 & -0.7071 & -0.4751 & 0.2485 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \\
\end{pmatrix}
\end{align*}
\]

\[
\Sigma = \begin{pmatrix}
3.3777 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 2.3183 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.9691 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.5271 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
\end{pmatrix}
\]

\[
V = \begin{pmatrix}
-0.4993 & 0.1357 & 0.0000 & 0.1237 & -0.3931 & -0.7500 & 0.0000 & 0.0000 \\
-0.2248 & -0.4126 & 0.7071 & 0.0149 & -0.1702 & 0.1667 & 0.3830 & 0.2748 \\
-0.3176 & 0.3597 & -0.0000 & -0.3814 & 0.2486 & 0.0833 & 0.6038 & -0.4371 \\
-0.3176 & 0.3597 & -0.0000 & -0.3814 & 0.2486 & 0.0833 & -0.2208 & 0.7119 \\
-0.3561 & -0.0486 & 0.0000 & 0.6724 & 0.6471 & 0.0000 & 0.0000 & -0.0000 \\
-0.2248 & -0.4126 & -0.7071 & 0.0149 & -0.1702 & 0.1667 & 0.3830 & 0.2748 \\
-0.4993 & 0.1357 & 0.0000 & 0.1237 & -0.3931 & 0.5833 & -0.3830 & -0.2748 \\
-0.2681 & -0.6013 & -0.0000 & -0.4753 & 0.3012 & -0.1667 & -0.3830 & -0.2748 \\
\end{pmatrix}
\]
Images indexing and retrieval, due to the above analysis, are described by the following weighted vector:

\[
\text{WeightedAnnotationVector} = 3.3777 \text{ Man} + 2.3183 \text{ Car} + 1.0 \text{ Grass} + 0.9691 \text{ Crash} + 0.5271 \text{ Race}
\]

From the above vector, ‘Race’ can be omitted from the annotation of the processed group of images.

6- CONCLUSION

LSA can be used efficiently to filter annotation concepts (i.e., natural language words) where semantic similarities among annotations attached to certain images and the set of queries posted to search engines, is an effective approach to determine and omit redundant words.

The accuracy of the results is corresponding to the distribution pattern of the natural language words over the query and the annotation at the same time, where semantic similarities among annotations and queries vectors should span the semantic space of a group of images that are to be de-noised.

The most obstacle facing this approach is the intensive calculations required by the LSA when new image added to the group of images which have been de-noised against redundant words.

7- REFERENCES