Semantic Knowledge Building for Image Database by Analyzing Web Page Contents

Yung-Kwang Lai, Song Liu, Liang-Tien Chia and Syin Chan
Center for Multimedia and Network Technology
School of Computer Engineering
Nanyang Technological University, Singapore 639798
{S8035225E, pg03988006, asltchia, asschan}@ntu.edu.sg

Abstract

In this paper, we present a method of semantic knowledge building for image database by extracting semantic meanings from Web page contents. The novelty of our method is that it is able to effectively extract media with a high degree of relevancy to a specific topic by incorporating word similarity and ontologies. The method is implemented in our Web image crawler and analysis system (WICAS). The system downloads Web pages and media automatically and further analyzes the semantic meanings of page contents to build up semantic knowledge for media entities. Subsequently, our system accepts high-level query terms and returns relevant media efficiently. Our experiment results show that with this new method of high-level content abstraction, media retrieval accuracy can be improved tremendously over traditional methods.

1. Introduction

With the rapid growth in the amount of Web content, there is an increasing demand for effective and efficient media retrieval on the internet. Currently there are no effective and accurate methods for media retrieval using high-level contents based on a user's single query. For example, one of the critical problems in CBIR is the mismatch between what such CBIR systems can provide and the needs of users. This problem is known as the semantic gap. The majority of image users look for images with specific semantic contents such as types of object, places or people. While current research work has been able to narrow the semantic gap, CBIR have been unable to incorporate it effectively. To overcome the problem, we propose a new method to build up semantic knowledge for image database by analyzing Web page contents.

The large amount of Web page contents provide us rich semantic information which describes Web images. Such information provides not only content but also context descriptions for semantic meanings of the images. Therefore, semantic Web image retrieval can be achieved through searching semantic knowledge built from Web page contents. Since the descriptions for the images in our semantic knowledge are semantically meaningful terms, users can directly retrieve the images using high-level query terms and avoid the semantic gap encountered in traditional methods. To extract that information accurately, we propose and implement various methods in our Web image crawler and analysis system (WICAS) and further build up semantic knowledge for image database.

In WICAS, a Web crawler fetches pages and images from the internet based on a user-supplied URL. These Web pages will undergo an analysis that consists of three main parts: dictionary generation, feature vector generation using term frequency and inverse document frequency (TF/IDF), and word similarity measurement. The system then processes the user's query, which comprises of one or more keywords. Based on these keywords and the earlier analysis, the system will retrieve and display the most relevant media.

In the rest of the paper, Section 2 introduces the system structure in detail. Section 2.1 describes the web image crawler and database design. Section 2.2 explains how the dictionary is generated, discusses the construction of feature vectors using TF/IDF, and describes web content analysis based on similarity matrix and path distance generated by using WordNet [3]. Section 2.3 explains how the images are retrieved. Experimental results are provided in Section 3. Lastly, the conclusion and future work are presented in Section 4.

2. System Structure

Figure 1 shows the overall system structure of WICAS. WICAS consists of three main components: the Web image crawler, the analysis system, and the user query interface. The image crawler is responsible for fetching and storing Web page contents into the WICAS database. These contents will go through the analysis and the results are stored in the database. When the user queries a certain topic, images will be retrieved from the database based on the keywords entered by the user and the final results will be displayed.

0-7803-9332-5/05/$20.00 ©2005 IEEE

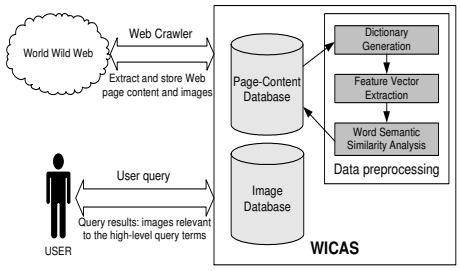

Figure 1: Overall system structure for WICAS.

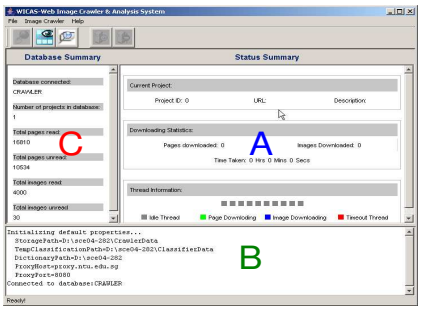

Figure 2: Downloading interface of WICAS. A) project information and download status, B) system status and C) database summary.

2.1 Web Image Crawler

By entering single Web links in individual projects, a user initiates the Web image crawler to crawl the website from the single link, as well as expand and branch to other websites. Web pages and images are downloaded and stored locally so that the web page contents can be analyzed later. Images in JPEG or GIF formats are downloaded only if their sizes are between 200×200 pixels to 4000×4000 pixels, or some other user-defined range. This helps to exclude images of menu bars, icons, buttons and advertisements. In addition, images from advertisements can be filtered out effectively by checking the number of times they are downloaded. Images that occur more than 2 times from every page are very likely to be repetitive and would be excluded. Figure 2 shows the download interface of WICAS.

With the base URL entered by the user, WICAS will crawl and store relevant hyperlinks of images and pages from various sites into the database. With the Web pages downloaded, we store into our database the relevant data comprising image sources, relations between pages, title, meta-keywords, meta-descriptions, anchor texts of links, text surrounding image links, as well as plain text in the Web pages. These data extractions assume that the Web pages conform to the World Wide Web Consortium (W3C) standards such as having meta-contents and using standard HTML codes.

2.2 Web Page Analysis

Web page analysis consists of 3 main parts: dictionary generation, page vector representation, and word similarity measurement, as seen in the next few sub-sections.

2.2.1 Dictionary Generation

While there are no specific standards on how Web images are labelled, most Web pages use anchor links and place descriptive text immediately together with the image. By capturing this information, we effectively draw out the keywords associated with the images. Also, a HTML parser is used to filter off irrelevant HTML codes and capture only plain text in the document.

The strings of words extracted are parsed using WordNet which can capture the various forms of words such as nouns, verbs, adverbs and adjectives. WordNet is a lexical reference system which organizes English nouns, verbs, adjectives and adverbs into synonym sets, each representing one underlying lexical concept. As we are concerned with retrieving images of entities such as places, objects and people, only words in the noun forms are retained to construct the dictionary. To limit the dictionary size, words that are only found in a few Web pages are excluded.

2.2.2 Page Vector Representation

To effectively represent the text information in each Web page, we make use of feature vectors based on Rocchio algorithm TF/IDF [1]. Given a Web page, firstly, we use the page-title, meta-keywords, meta-descriptions, anchor texts of links, surrounding text of images, and plain text in the page to build up the page data. Subsequently we calculate the TF/IDF for each word in the page data using equation:

$$tfidf(t_k, d_j) = \#(t_k, d_j) \cdot \log \frac{|D_s|}{\#D_s(t_k)} \qquad (1)$$

where $\#(t_k, d_j)$ denotes that the number of times word t_k occurs in the page d_j, $|D_s|$ is the cardinality of the set D_s of all pages under one project, and $\#D_s(t_k)$ denotes the number of pages in which the word t_k occurs. Finally, we convert the page data into a feature vector. The vector length is the size of the dictionary and its components are the TF/IDF values of each word. The feature vectors of all the Web pages are constructed in the same way.

2.2.3 Word Similarity Measurement

In this section, we present two methods that are capable of improving high-level word-context abstraction. In the first method, we take pairs of words from the dictionary and compute their similarity values using the WordNet API. These similarity values are then used to construct a word semantic similarity matrix (WSSM) M which is a square matrix of the same dimension as the dictionary. In our experiments we name this matrix WSSM1.

To obtain the semantic similarity values between a pair of words, we first implemented the method provided by

JavaSimLib toolbox [4] which is an extension project using WordNet. The words are compared and a similarity value, which is a double-precision value that lies in the interval $[0, 1]$, will be returned by the toolbox that measures how similar the two words are. For example, 'Car' and 'Automobile' would have the similarity value 1.0 as each clearly defines the same object as the other. Likewise, words 'Coast' and 'Shore' have a similarity value close to 1.0 since they share very similar meanings. On the other hand, 'Coast' and 'Forest' are different in meanings and would have a similarity value close to 0. Since we wish to identify words that have similar semantic meanings, we filter out those similarity values that are lower than a threshold α. In this way, the query terms entered by the user can be expanded to include other semantically similar words in the dictionary. For example, if a user queries the topic on 'Monkey', the following words like 'Chimp', 'Chimpanzees', 'Primate', 'Gorilla', 'Ape', 'Gibbon', 'Orangutan', 'Orang', would also be considered relevant.

One shortcoming of the above approach (WSSM1) is that for given words t_k and t_j the similarity $S_v(t_k, t_j) = S_v(t_j, t_k)$, which means the order of influence is not considered in the semantic similarity. For example, S_v('monkey','animal') $= S_v$('animal','monkey'). In this case, for the query keyword 'animal', 'monkey' will be a correct inference, but 'monkey'→'animal' would be an over-expanded inference and it can lead to a lot of irrelevant returns in the query results. This prompts us to propose an alternative method.

In the second method, we construct the matrix M in a slightly different way. We name this matrix WSSM2 in our experiments. Using WordNet, we can obtain a different kind of information that also measures the semantic similarity between words. That information contains a common parent index I and the depths of different nodes or words D_p, which share the same common parent and have different levels. Based on this information, the semantic relationships between two words can be summarized into these four types: 1) *Synonyms*: If words t_k and t_j are synonyms, then $I(t_k, t_j) = I(t_j, t_k) = 0$ and $D_p(t_k, t_j) = D_p(t_j, t_k) = 0$; 2) *Child-parent relationship*: If word t_k is the child of t_j, then $[I(t_k, t_j) = D_p(t_k, t_j)] > 0$, and the value of D_p or I shows how different the semantic meanings of (t_k, t_j) are; 3) *Parent-child relationship*: If word t_k is the parent of t_j, then $I(t_k, t_j) = 0$, $D_p(t_k, t_j) > 0$ and the value of D_p shows how different the semantic meanings of (t_k, t_j) are; and 4) *Sibling relationship*: If word t_k is the sibling of t_j, then $I(t_k, t_j) > 0$, $D_p(t_k, t_j) > 0$ and the combination of D_p and I shows how different the semantic meanings of (t_k, t_j) are. For each type of relation, the combined values of I and D_p are quantized into a value in the interval $[0, 1]$. This value is obtained for every pair of words in the dictionary and the values are used to construct the matrix WSSM2.

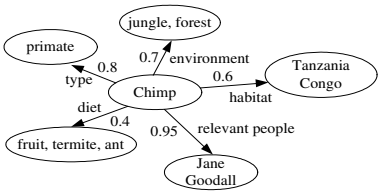

Figure 3: Example ontology for keyword 'chimp', the arrows denote relationship and ellipses denotes concepts. The values on the arrows indicate the weights of the properties defined by a user.

To complement WSSM and make good use of available domain knowledge, an ontology is also integrated in our Web content analysis. It is used as a formalized representation of a particular knowledge in a domain taken from certain perspective or concept. By deploying word ontologies, words that are related to the query topic, but do not share the same semantic meanings, can be easily obtained and used for analysis. Ontologies are readily available on the World Wide Web for various purposes. In Figure 3, which shows an example of our ontology, we have defined common properties that relate to a Chimp such as the type, environment, habitat, diet and people relevant to research on Chimps. Weights are user defined depending on the importance of the properties.

2.3 Distance Measure

As described in Section 2.2.2, each Web page is represented as a TF/IDF vector. The column page vectors of a collection of Web pages are put together to form a matrix of order $L \times |Ds|$, called the page data matrix. The query, which contains one or more words, is also represented by a query vector that has the same length as the page vector. The query vector will consist of one or more non-zero elements corresponding to keywords that are found in the dictionary. Therefore, the query process is one that searches for the page vectors that are geometrically close to the query vector. In the current implementation, we use the cosine distance between the query vector and page vector to measure their similarity,

$$\cos\theta_k = \frac{a_k^T q}{||a_k||_2||q||_2} = \frac{\sum_{j=1}^{L} a_{kj} q_j}{\sqrt{\sum_{j=1}^{L} {a_{kj}}^2}\sqrt{\sum_{j=1}^{L} {q_j}^2}} \quad (2)$$

for $k = 1 \cdots |Ds|$, where a_k is a column vector from the page data matrix, q is the query vector and L is the length of the feature vector. Thus, pages whose page vector returns a value from equation (2) that exceeds a certain threshold are deemed relevant. Alternatively, the values of $\cos\theta_k$ can be sorted to return the top n results. In our experiments, we name this method the original matrix (OM).

A well-documented problem of the above method is that only pages containing the exact query terms will be returned as results. To solve this problem in WICAS, we implemented the various word semantic similarity measurements (WSSM1, WSSM2, Ontology) as described in the previous sub-section. For a given dictionary $Dict$, each element

keyword (ground-truth)	OM		WSSM1		WSSM2		OM-Onto		WSSM1-Onto	
	C/F	C/T	C/F	C/T	C/F	C/T	C/F	C/T	C/F	C/T
chimp (49)	21/6	18/20	39/54	21/30	38/35	25/30	21/5	18/20	41/26	27/30
tiger (61)	17/67	12/20	43/79	25/30	48/56	25/30	17/24	15/20	52/53	27/30
bear (63)	7/15	7/20	43/79	12/30	48/65	16/30	7/14	7/20	44/43	22/30
volcano (52)	3/21	3/20	39/70	20/30	38/55	21/30	3/11	3/20	42/39	26/30
red-indian (59)	19/69	13/20	39/80	23/30	40/62	21/30	18/24	15/20	43/39	24/30

Table 1: Results for image retrieval based on given keywords. We carried out the experiment based on five methods including simple $\cos\theta_k$ method (OM), together with WSSM generated using similarity values (WSSM1), together with WSSM generated from path distance (WSSM2), together with relevant ontology (OM-Onto), and lastly ontology together with WSSM1 (WSSM1-Onto). C/F gives the number of correct images vs. false images, C/T gives the number of correct images in the top T results.

M_{ij} in M represents the semantic similarity of words $Dict_i$ and $Dict_j$. Therefore, we modify equation 2 to:

$$\cos\theta_k = \frac{a_k^T M q}{||a_k||_2||q||_2} = \frac{\sum_{i=1}^{L}\sum_{j=1}^{L} a_{kj} M_{ij} q_j}{\sqrt{\sum_{j=1}^{L} {a_{kj}}^2}\sqrt{\sum_{j=1}^{L} {q_j}^2}} \quad (3)$$

With this modified equation, together with the original matrix equation, different tests are carried out and the results are presented in the next section.

3. Experimental Results

In this section, we show the experimental results for image retrieval using keywords that represent high level concepts. The keywords used in the tests include 'chimp', 'tiger', 'bear', 'volcano' and 'red-indian'. We crawled a total of 16,810 Web pages and 4000 images from 3 websites. From these Web pages, 9,226 words were selected to form the dictionary.

Table 1 shows the experimental results of different image retrieval methods using five high-level keywords. In the table, "ground-truth" is the total number of possible pages which contain images with topics relevant to the user's query. C/F means the number of correct images versus false images, and C/T represents the number of correct images in the top T images. Take for example the word 'chimp', the ground truth is 49. Using OM, out of 27 search entries, 21 of them are correctly retrieved with 6 falsely identified. Of the top 20, 18 are correctly retrieved. High precisions of 78% based on C/F and 90% based on C/T are obtained. Using WSSM1, the precision drops to 42% based on C/F but that for the top 30 still maintains high at 70%. Similarly for WSSM2, the precision is 52% based on C/F but the 80% based on C/T. The results for OM-Onto with precision of 81% based on C/F and 90% based on C/T remains impressive. WSSM1-Onto yields precision of 61% based on C/F and 90% based on C/T.

Both WSSM methods can greatly improve the efficiency of retrieval in some cases. For example, those pages that contain images of 'volcano' seldom have this keyword, but may contain related words such as 'Fuji', 'Krakatau' and 'eruption'. By comparing WSSM1 and WSSM2, we see that the latter can effectively reduce the number of false results by 20%. Relevant domain information is also helpful in reducing the number of false results, in addition to increasing the correct ones. This can be seen in the results of OM-Onto and WSSM1-Onto. For OM-Onto, the query search is based on the keyword first followed by expansion using ontology. In WSSM1-Onto, WSSM1 is applied first followed by ontology. Overall, our system performance achieves 52.6% average precision and 84.0% average accuracy for top 30 results in retrieving images based on high-level abstract concepts.

4. Conclusion

We have described a new method to build semantic knowledge for image database by analyzing Web page contents and implemented it in WICAS. The experimental results of image retrieval demonstrate that by using generated dictionaries, word similarity and ontologies, WICAS provides accurate semantic descriptions for Web images and is able to achieve high accuracies for Web image retrieval. In our future work, improvements can be made to the dictionary generation in which more relevant words can be extracted and sorted based on Web pages' structures. Moreover, useful low-level image features will also be considered in the next version of WICAS [2]. By combining the strengths of both high and low-level content abstraction, an even greater accuracy in media retrievals can be accomplished.

References

[1] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151, 1997.

[2] S. Liu, L.T. Chia, and S. Chan. Ontology for nature-scene image retrieval. In *Proceedings OTM confederated international conferences*, October 2004.

[3] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography 3*, pages 235–244, 1990.

[4] N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of ECAI'2004, the 16th European Conference on Artificial Intelligence*, 2004.