Giving Meanings to WWW Images

Heng Tao Shen Beng Chin Ooi Kian-Lee Tan

Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543

ABSTRACT

Images are increasingly being embedded in HTML documents on the WWW. Such documents over the WWW essentially provides a rich source of image collection from which users can query. Interestingly, the semantics of these images are typically described by their surrounding text. Unfortunately, most WWW image search engines fail to exploit these image semantics and give rise to poor recall and precision performance. In this paper, we propose a novel image representation model called *Weight ChainNet*. Weight ChainNet is based on *lexical chain* that represents the semantics of an image from its nearby text. A new formula, called *list space model*, for computing semantic similarities is also introduced. To further improve the retrieval effectiveness, we also propose two relevance feedback mechanisms. We conducted an extensive performance study on a collection of 5000 images obtained from documents identified by more than 2000 URLs. Our results show that our models and methods outperform existing technique. Moreover, the relevant feedback mechanisms can lead to significantly better retrieval effectiveness.

Keywords:

WWW, semantic similarity, image retrieval, relevance feedback, image representation.

1. INTRODUCTION

With the increase in Internet bandwidth and CPU processing speed, the use of images in WWW pages has become very prevalent. Images are used to enhance description of content, to capture attention of readers and to reduce the textual content of a page. Images have become an indispensable component of WWW pages today. This pool of WWW images becomes a very rich source from which users can obtain interesting images. However, managing such images to facilitate their retrieval is an interesting research topic that has not received much attention. In particular, to be able to search for relevant images among such a large collection of images calls for novel mechanisms that exploit the semantics of the images.

Traditional image retrieval systems are not adequate to deal with the problem. Text-based systems [1, 2, 3] use keywords or free text description of images supplied by the authors as the basis for retrieval. These systems can be adopted for WWW images since the textual content of the HTML page in which the image is embedded provides the free text description. However, the entirety of the textual content does not represent the semantics of the image adequately for them to be useful in retrieving the images. In other words, while the textual content may contain information that captures the semantics of the embedded image, it also contains other description that are not relevant to the image. These “noises” lead to poor retrieval performance.

On the other hand, content-based image retrieval systems [4, 5, 6, 7, 8, 10] capture the visual content of an image (such as color, texture and shape) as its semantics and use these features as the basis for similarity matching. Unfortunately, retrieval by content is still far from perfect. First, their effectiveness depends on how precise the user specifies the query. Second, they cannot capture the more useful image semantics, like object, event, and relationship. Finally, they do not scale well. More recently, integrated systems that combine the various features (color, texture and shape) has led to better effectiveness. But, they remain unsatisfactory as different features tend to have different degrees of importance for different classes of queries.

In this paper, we adopt a different approach to identify the semantics of an image within a HTML document. This is based on the observation that an image in a Web page is typically s*emantically* related to its surrounding texts, with the exception of *functional* images (such as *new* symbol and *under construction* symbol). These surrounding texts are used to illustrate some particular semantics of the image content, i.e. what objects are in the image, what is happening and where the place is. In particular, in a HTML document, certain components are expected to provide more semantic information than other portion of the text. These include the caption of the image, its title and the title of the document. We propose a novel image representation model called *weight ChainNet*. Weight ChainNet is based on *lexical chain* obtained from an image's nearby text. A new formula, called *list space model*, for computing semantic similarities is also introduced. To further improve the retrieval effectiveness, we also propose two relevance feedback mechanisms. We conducted an extensive performance study on a collection of 5000 images obtained from documents identified by more than 2000 URLs. Our results show that our models and methods outperform existing technique. Moreover, the relevant feedback mechanisms can lead to significantly better retrieval effectiveness.

The rest of this paper is organized as follows. In the next section, we briefly review some related works. In Sections 3 and 4, we present our image semantic representation model and the similarity measure respectively. Section 5 presents the relevance feedback approaches to refine queries for further retrieval. In Section 6, we describe an experimental study and report our findings, and finally, we conclude in Section 7.

2. RELATED WORKS

As discussed in the introduction, traditional text-based and content-based retrieval mechanisms are no longer effective for managing images obtained from the WWW. There have also been several approaches that combine hypertext in WWW pages with information retrieval (IR) engines [10, 11, 12, 13, 14, 16]. These techniques, however, do not apply well to images in WWW for the same reasons.

Recently, [15] extended [16, 18] to work with WWW-based image collections. In [15], an image's content is given by the combined content of the text *nodes*. An image's set of text nodes include textural content (e.g., caption) obtained from the document in which it is embedded, as well as those obtained from its neighboring pages (those pages that are reached by a single hyperlink from the embedded page). This model was further extended to take into account not only the textual content of the immediate neighbors of an image, but also all nodes that can be reached from the image by following at most two hyperlinks (a *two-step link*), thus considering more information about an image node. However, there are no explicit image/query semantics considered. The inner semantic relationship within a text node was lost based on this model. Moreover, while keeping more information is desirable, the approach extracted too much unrelated information. For example, an image's own caption usually describes its content, but its neighboring pages' image captions do not reflect the same content. In addition, the similarity measure did not take into account any semantic structure. Such a similarity measure may not be good enough to show the *real* semantic similarity between an image and a query.

Relevance feedback (RF) is a very important way to improve the accuracy. System refines the query by using feedback information from users to improve subsequent retrieval. The use of relevance feedback using multiple attributes of color has been investigated in [9]. Their results showed significant improvement in retrieval effectiveness by applying RF mechanisms.

3. IMAGE REPRESENTATION MODEL

Two key issues must be addressed in designing an image retrieval system to support WWW images:

- Determine a representation for a WWW image and the query semantics.
- Determine a similarity measure between an image and a query based on their representations.

In this section, we shall address the first issue, and defer the second issue to the next section. Before that, we have identified several desirable properties of a query/image representation:

- *Exactness*. For a representation to be effective, it has to capture the essential image/query semantic meanings.
- *Space efficiency*. The representation should not consume too much storage; otherwise, besides large storage cost, the database structure will not be effective in reducing much I/O cost.
- *Computationally inexpensive similarity matching*. It should be fast to compute the similarity between the representations.
- *Preservation of the similarity between the image/query semantic meanings.*
- *Automatic extraction*. The representation should be automatically extracted, rather than manually generated.
- *Insensitivity to noise, distortion, rotation*. Any noise or distortion should not affect the representation drastically.

3.1 Semantics of an Embedded Image

To understand the relationship between an image embedded in a HTML document and its surrounding text, we conducted a preliminary study on a collection of images obtained from HTML documents. Based on our findings, we have identified four parts of the textual content that are well related to the embedded image. These are

- Image title. Image file title (simply image title) is a single word that basically indicates the main object that the image is concerned with.
- Image ALT (alternate text). The image ALT tag in HTML document is a phrase that usually represents an abstract of the image semantics.
- Image caption. The image caption usually provides the most semantics about an image. It is the image's surrounding text in the HTML document. It can range from one sentence to a paragraph of text that contains many sentences.
- Page title. Since images are used for enhancing the Web page's content, page title is most probably related to the image's semantics. It is usually a short sentence that summarizes the Web page's content.

There are also some other parts which may provide some information about the image, such as other HTML meta data, However, they contain too much unrelated information. We have also excluded the textual content of the whole HTML document as part of the image's semantics for the same reason, i.e., that some information may be completely unrelated to the image content, and indexing the whole HTML document for each image in a very large database is not expected to provide an efficient solution. Therefore, we just use these four parts to represent image content. We note that all these four parts -- image title, image ALT, page title and image caption -- can be automatically extracted from the HTML document based on hypertext structures.

3.2 Weight ChainNet Model

To represent the image semantics more adequately, we propose the Weight ChainNet model that is based on the concept of *lexical chain* [17]. Figure 1 illustrates an example. A lexical chain (LC) is a sequence of semantically related words in a text. Here, we define it as one sentence that carries certain semantics by its words. As an image title is just a single word, we say it's a trivial lexical chain - *Title Lexical Chain (TLC).* The text obtained from the ALT tag is referred to as the *Alt Lexical Chain (ALC).* The page title is represented as a LC too - *Page Lexical Chain (PLC).* Finally, since a caption comprises multiple sentences, we represent it as three types of lexical chains. Type one is called *sentence lexical chain (SLC),* which represents one single sentence in an image caption. In Figure 1, each sentence is shown as one column in the caption component, i.e., each column is a SLC. Type two is called *reconstructed sentence lexical chain (RSLC)*, and it represents one new sentence reconstructed from related sentences. Two sentences are *related* if both share one or more words. One common word in two SLCs splits each SLC into two. Based on the first common word, the second SLC's second half is connected to the first SLC's first half to form a RSLC. In Figure 1, a RSLC exists if there is an arrow from one column to another column. The last type is called *caption lexical chain (CLC)*, which represents the whole image caption. A CLC is formed by connecting SLC one after another. In Figure 1, the connections are made by dotted arrows. To illustrate, the followings are some examples from Figure 1.

SLC (1→ 2→ 3→ 4→ 5),
RSLC (1→ 2→ 8→ 9),
CLC (1→ 2→…→ 13→ 14).

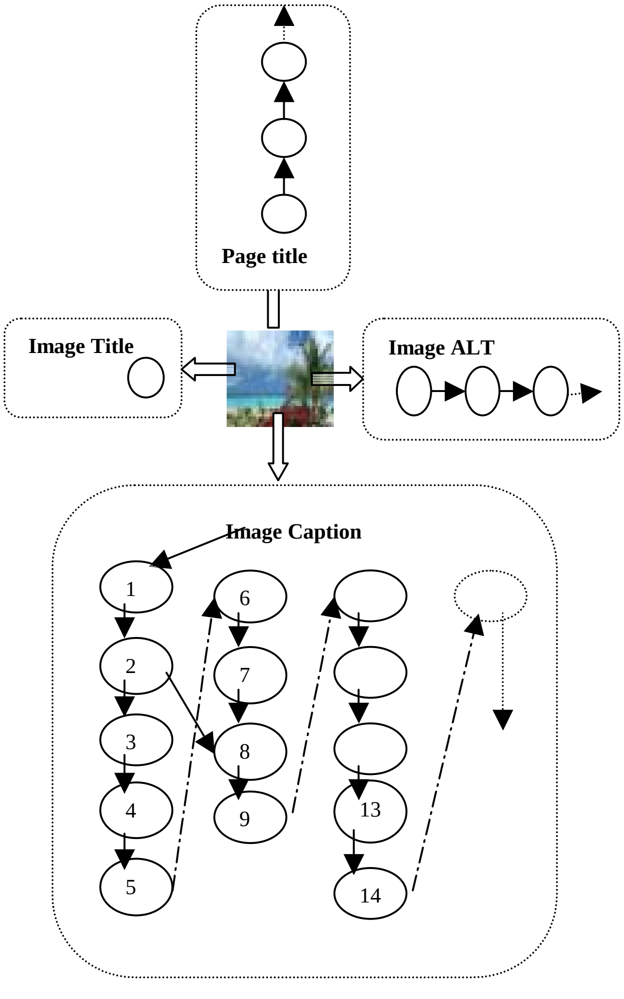


Figure 1: Image Semantic Representation - *Weight ChaiNet*

The ChainNet model is built by these 6 types of lexical chains. Each chain captures a portion of the semantic structure of the image. A TLC indicates the main subject of an image. An ALC provides short description about an image. A PLC shows part of its content. An SLC captures the semantics of a single sentence in the image caption. An RSLC captures related sentences' semantics, and a CLC keeps the image's overall semantics. That's why we call it ChainNet, which is basically made of a chain of LCs.

However, the ChainNet treats each type of LC as of equal importance, Now, simply representing an image in this way without capturing the relative importance of the various components is not expected to lead to good performance. For example, the image title, ALT, page title and image caption play different roles in representing an image's semantics. The reason we have divided the entire image caption into three types of lexical chains is that we want to differentiate the importance of each type of sentences due to their positions and inner relationship within an image caption. The three types of lexical chains in an image caption are not equally important. The importance order from high to low is expected to be like this: SLC > RSLC > CLC. If all the same words in a query appear in an SLC, an RSLC and a CLC respectively, the SLC possesses the most semantic meanings among the three, followed by RSLC and finally CLC. For example, if a query matches an SLC in the first image, only matches an RSLC in the second image, and only matches a CLC in the third image, it's most likely the case that the first image is most relevant to the query, followed by the second image and then the third image, because an SLC is more semantically structured than an RSLC, which is more semantically structured than a CLC.

To capture the relative importance of the various types of LCs, we assign weights to the various LCs such that LCs that are deem to be more representative of the image content are assigned larger weight values. We shall see how these weights come into play in the similarity measure to be discussed in the next section. We note that for the caption, one word in the caption may have up to three different weights with respect to the lexical chains it belongs to. Of course, each word has at least two weights: a SLC weight and a CLC weight. If the word belongs to one RSLC, it will have three weights. One image caption may have several SLCs and several RSLCs, but only one CLC.

The resultant Weight ChainNet model uses a well-structured notion of image's content to capture the semantic relationship between an image and its nearby text. Such a model can be seen as a semantic representation of the content of an image. This model has the properties of exactness, since it captures an image's essential semantic meanings by an image title, ALT, page title and caption. It is space efficient because it does not keep too many words for image representation. The content can be automatically extracted. It is insensitive to noise since all words are stemmed and no stop words exist. Finally, similarity matching is computationally inexpensive using the proposed *list space model* which we shall introduce in the next section.

For a user query, it's usually a free sentence that describes the image content. Naturally, we represent it as a *Query Lexical Chain - QLC.*

4. SEMANTIC MEASURE MODEL

In this section, we will present our similarity measure model between two lexical chains, and between an image and a query respectively.

4.1 Similarity between two Lexical Chains

We have presented the model for representing image/query semantics. To calculate the semantic similarity between a query and an image, we start from determining the similarity between two basic components in an image ChainNet - LC. In our implementation, we store terms of each LC as a list. All the lists belonging to an image are connected to the image root as shown by the ChainNet model (see Figure 1). We propose a *list space formula* to compute the similarity between two LCs as follows:

$$Similarity_{list1,list2} \equiv \frac{\sum_{i=0}^{list1.size()} \sum_{j=0}^{list2.size()} e_i.weight * e_j.weight}{\sqrt{list1.size()} * \sqrt{list2.size()}} * MatchScale$$

where e_i and e_j are matched words in list 1 and list 2 respectively. Two words are *matched* if they are the same word. We note that we have removed stop words and performed stemming from the various LCs.

In the formula, one important parameter is considered: *MatchScale*. Match scale is defined as the closeness of two lists from the view of match order. For example, one LC is " *US president Clinton and wife visited China in 1997*", and the other one is: " *China president Jiang Zemin welcomed Clinton and wife in Tian'an square*". For these two LCs, there are four matching words. For the first LC, the matched words are in order of "*president Clinton wife China"*, and in the other, they are *"china president Clinton wife"*. We treat each one as a child LC of its original LC. Therefore, the orders of matched words in the two original LCs are not the same. Obviously, the closer the matched order of two children LCs are, the closer the semantics of the original two LCs are. Inspired from the formula for the *angle between two nonzero vectors in 2d-space,* we define the match scale as below:

$$MatchScale_{v1,v2} \equiv \frac{v1 \bullet v2}{\| v1 \| * \| v2 \|}$$

where v1 and v2 represent the child LC of the first and second original LCs respectively. The element value is the position in their respective LC. But the *dot product* between two LCs is redefined as the following:

$$v1 \bullet v2 \equiv \sum_{i=1}^{v1.size\,()} v1_i * v2_j$$

Where $v2j$ is the matched word in $v2$ for $v1i$ in $v1$. As mentioned, two words are matched as long as they are the same.

The above measure determines the similarity between two LCs. However, the two LCs may not be semantically related. For example, consider the query "Singapore Map". An image about Singapore Food, say I1, that contains several occurrences of "Singapore" in CLC may result in a high similarity value even though the images are not semantically related. On the other hand, another image about Singapore Map, say I2, contains only one occurrence of "Singapore" in CLC may result in a lower similarity value despite the fact that it is a desired image. To ensure that two LCs are *semantically related*, we need another parameter called: *Match Level*. Match Level is the number of the *distinct* matched words by a LC and a QLC, denoted as: **LCMatchLevel (LC, QLC)**. The *match level threshold* is the minimum match level for a LC to keep its original semantics. We say one LC is semantically related to a QLC, if and only if the LC's match level is equal to or greater than QLC's match level threshold. Therefore, in our semantic measure model, *semantic similarity* for a LC with respect to a QLC is indicated by the *similarity* calculated by list space formula in its *match level*. The match level determines *if* the LC is semantically related to the QLC. And the similarity calculated by list space model shows how *well* it is semantically related to the QLC.

4.2 Similarity between ChainNet and LC

Now it is time to calculate the semantic similarity between an image and a query. From the discussion above, we know that an image is represented by a Weight ChainNet, and a query is in the form of a lexical chain. To calculate their similarity, we use the following formula:

$$imageSimilarity_{image,query} \equiv S(TLC, QLC) + S(ALC, QLC) + S(PLC, QLC)$$
$$+ \sum_{i=1}^{SLC.number} S(SLC_i, QLC) + \sum_{i=1}^{RSLC.number} S(RSLC_i, QLC) + S(CLC, QLC)$$

where S is the similarity between two LCs. The image match level is defined as:

ImageMatchLevel (ChainNet, QLC) =
MAX (TLC.weight * LCMatchLevel(TLC, QLC),
ALC.weight * LCMatchLevel(ALC, QLC),
PLC.weight * LCMatchLevel(PLC, QLC),
SLC.weight * LCMatchLevel(SLC, QLC),
RSLC.weight * LCMatchLevel(RSLC, QLC),
CLC.weight * LCMatchLevel(CLC, QLC))

We say one image is semantically related to a query if and only if its match level is equal to or greater than the query's match level threshold. It has the similarity calculated by the above formula with the query in its match level.

5. RELEVANCE FEEDBACK

Because of the large image collection and the impreciseness of a query, it is important to provide mechanisms to help users in specifying their queries more accurately. One such mechanism is to exploit feedback from users based on resultant images returned from the initial query. By allowing users to indicate the relevant (and irrelevant) images, the original query can be refined to further improve the retrieval effectiveness. For this purpose, we develop two techniques: *semantic accumulation* and *semantic integration and differentiation*.

5.1 Semantic Accumulation

The first method, called *semantic accumulation*, allows the user to pick the most relevant image (from the user's subjective judgement) from the result of previous retrieval as the feedback image. The method accumulates all the previous feedback images' semantics to construct a new query for the next retrieval. The resultant query is represented as a kind of ChainNet called Weight F/Q ChainNet (Feedback/Query ChainNet) since it is constructed by the query and the feedback image's ChainNet. This kind of new query is represented in Figure 2.

Obviously, the combination of every entire ChainNet from each previous feedback images is tedious if the user searches again and again. More seriously, more noise will be added into the new query. Therefore, rather than a whole image ChainNet, we use just one single lexical chain which is most semantically related to the original query in the previous feedback image's ChainNet. This is calculated by the list space model. The steps for this method are:

1. Perform search using the F/Q Weight ChainNet (or Weight ChainNet for first attempt)
2. User selects the current feedback image
3. Construct the feedback image's Weight ChainNet
4. Extract the closest lexical chain to the original query from the feedback image by list space model
5. Use the QLC and the weight ChainNet to construct F/Q ChainNet
6. Use that extraced LC and old QLC to construct new QLC
7. Go to step 1

In this algorithm, the semantic is accumulated by adding one most related LC from every previous ChainNet to QLC to form a new QLC. Therefore, the QLC carries richer and richer semantics as users provide more feedback.

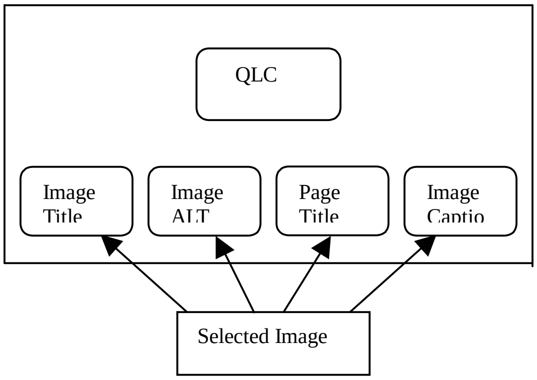


Figure2: F/Q ChainNet in Semantic Accumulation

5.2 Semantic Integration and Differentiation

In the semantic accumulation feedback approach, users can only select one image at a time as the feedback information. To save time and to filter more unrelated images, we introduce another technique: semantic integration and differentiation. In this method, users can select several relevant and irrelevant images simultaneously. By relevant, we mean images that are semantically related to the query as judged by the user and hence should be retrieved. On the other hand irrelevant images are those that the user considers to be unrelated and should not have been retrieved. The system *integrates* the related semantics obtained from the relevant feedback images to construct a new query for the next try. After that, the system combines the semantics from irrelevant images to *differentiate* the irrelevant images from the returned results. The new query is also represented by a F/Q Weight ChainNet as shown in Figure 3.

The steps for this method are:

1. User selects a number of relevant and irrelevant images.
2. Extract the most semantically related LC from each relevant image's ChainNet to form a new F/Q ChainNet with QLC as a new query.
3. Extract the most un-semantically related LC from each irrelevant image's ChainNet to form a ChainNet for bad images
4. Submit the query
5. From each returned image, remove it from results if it's more related to the bad images' ChainNet.
6. Go to step 1

The semantic similarity formula between two ChainNets can be easily extended from the formula for measuring the similarity between an image and a query.

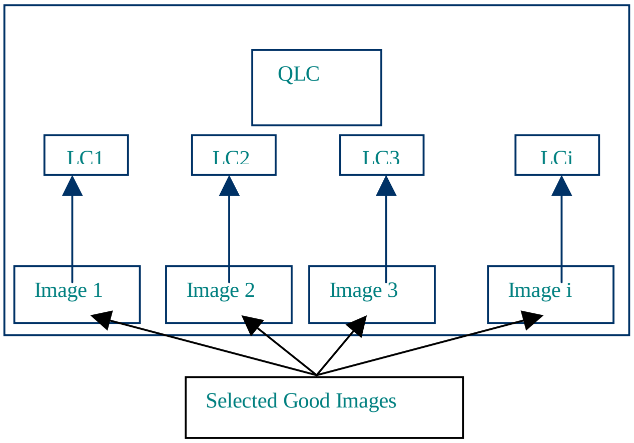


Figure3: F/Q ChainNet in Semantic Integration and Differentitaion

6. EXPERIMENTS

To study the effectiveness of the proposed method, we implemented the proposed model in an image retrieval system and conducted an extensive performance study. This section reports our study and findings.

6.1 Experimental Setup

For purpose of testing the model, we "centralized" the image collection (instead of simply extracting the image at runtime from the various Web sites/pages in the form of a search engine). This is achieved through the design of a Web crawler that automatically searches the WWW for documents with embedded images. The crawler also extracts the image title, image ALT, page URL, page title and image caption from the HTML documents as the images' semantic content. In total, we collected 5232 images from over 2000 different URLs. These images are general, random and diverse enough for us to test on any query. We used 12 text descriptions, as shown in the following table, as our queries for our experiments.

Table 1. Test queries.

Query	Query Description
Q1	Singapore map
Q2	Travel in Spain
Q3	Valentine flower
Q4	Island in the sky
Q5	California beach girl
Q6	England football league
Q7	Green lizard on a red leaf
Q8	Husband is kissing his wife
Q9	National University of Singapore
Q10	Hollywood superstar Jennifer Lopez
Q11	Elephant in the beautiful national park
Q12	Celebrations for new millennium of 2000

Given that we have over 5000 images, it is not practical to scan all images to obtain the relevant images for each query. To determine the set of relevant images for the queries, we adopt the following realistic approach. For each query, we expand the query terms to include terms that are related. This is done using the WordNet [19]. For example, the term girl may be expanded to include the term woman. Each term is then used as a query to extract the list of images whose semantics (or rather the LCs) contain that term. The union of the results from each term form a candidate set of relevant images. We then manually examine the candidate set to eliminate those that are not semantically related to the query to get final set of relevant images.

6.2 Tuning the Weight ChainNet Model

Tuning the Weights

Weight ChainNet model calls for some tuning to be performed. As mentioned earlier, there are 6 types of LCs and different LC types may have different significance in identifying the image semantics. In the first experiment, we evaluate the performance of each type of LCs exclusively to study their different impact on retrieval effectiveness. Figure 4 shows the results.

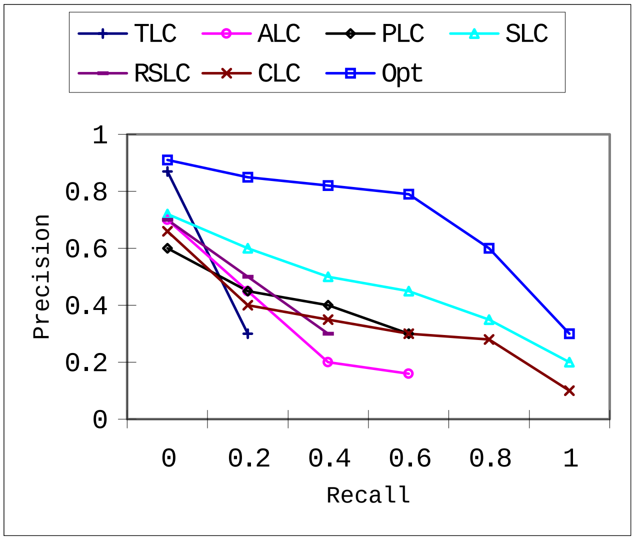


Figure 4: Utility by each Type LC alone to Represent Image

From Figure 4, we can see that for TLC, it cannot achieve >20% recall, although it has high precision. This is due to the lack of information in TLC. For similar reasons, PLC and ALC are not very effective also. For RSLC, since quite a number of images do not have RSLC, it cannot achieve high recall. Only CLC and SLC can result in high recall, but the precision is not satisfactory. TLC, ALC, PLC and RSLC can be used to improve the precision a lot. On the other hand, SLC and CLC can improve the recall. From this result, we have a rough picture of the relative importance of each type of LCs. Clearly, SLC is the most important, followed by TLC, RSLC, ALC or PLC, and finally, it is CLC.

To determine the weights to be assigned to the proposed model that combines all the LCs, we tested different weight combinations from values in 0, 0.2, 0.4, 0.5, 0.6, 0.8 and 1 for each type of LCs. However, we narrowed the search space based on the result from Figure 4 by adopting some simple heuristics. For example, since SLC is the most important, we fixed its weight at 1.0. Moreover, for a LC that is more important, the weights assigned to the other less important LCs cannot be more than its weight. In total, we tested 22 combinations and obtained the following weight assignment for the various LCs: TLC (0.8), ALC(0.6), PLC(0.6), SLC(1), RSLC(0.5), CLC(0.2). In this experiment, we have fixed the scale parameter *coef* of the match level to be 0.6 (see Tuning the Match Level). We shall refer to this scheme as OPT. We also presented the result of OPT in Figure 4. As shown, we can get more than 80% precision with recall of 60%.

Though this experiment is meant to tune the proposed method, we note that it is also a comparative study among the different schemes. Clearly, the results show that using a single LC exclusively cannot provide the best performance, even though such an approach is clearly simple. Moreover, it shows that proper combinations of the various LCs can lead to very effective retrieval results. We also note that the exclusive CLC scheme can be viewed as a form of traditional text-based system without any semantic structure involved. Thus, we expect OPT to outperform existing schemes too.

Tuning the Match Level

There is another parameter that we have to tune, the match level. Recall that the match level is the number of common terms shared by two lexical chains. It determines whether two LCs are semantically related, and then derives if two images are semantically related. In our evaluation system, only those semantically related images are returned.

One single word cannot reflect the semantic meaning of a whole query. If the match level threshold is too small, too many images may be returned to the users. On the contrary, too few images are displayed if the match level threshold is too high. Therefore, it is necessary to choose the best match level thresholds. Since the length of a query is a random variable, a fixed value for match level is not applicable to various queries. We thus define the match level as a linear function of query length:

*MatchLevel Threshold= coef * query.length()+ constant* where the *coef* is the scale parameter we need to explore in order to get the best results in a reasonable volume. And the *constant* is just an adjustable value.

We tested those 12 queries in Table 1 in order to select the best *coef*. Figures 5a and 5b shows the relationship between precision and *coef* and recall and *coef* respectively.

From Figure 5a, when *coef* is > 0.6, the precision will be greater than 85%. From Figure 5b, we can see that when *coef* is < 0.6, the recall is greater than 60% which is very satisfactory to a large image database. Therefore, observing the combined effect, we select 0.6 as the optimal value of *coef*.

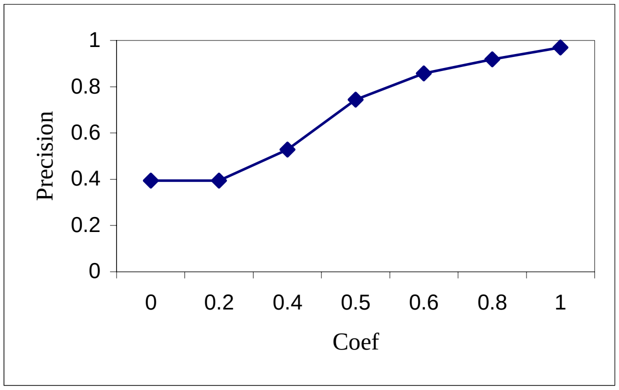


Figure 5a: Precision Vs. Coef

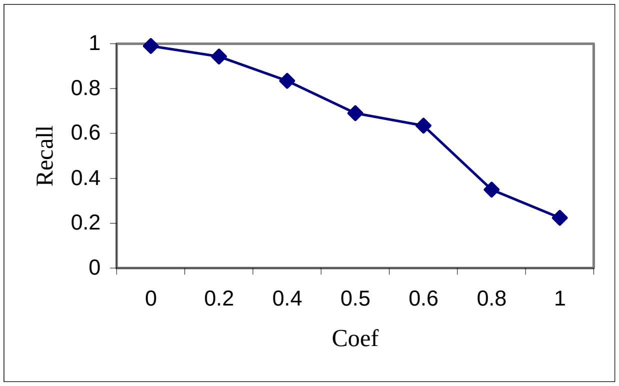


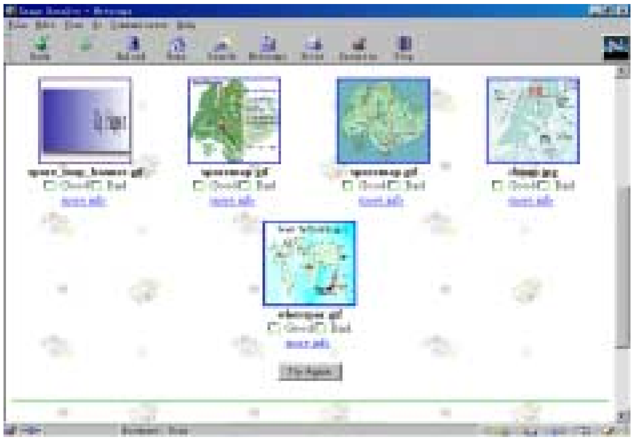
Figure 5b: Recall Vs. Coef

Impact of Match Scale

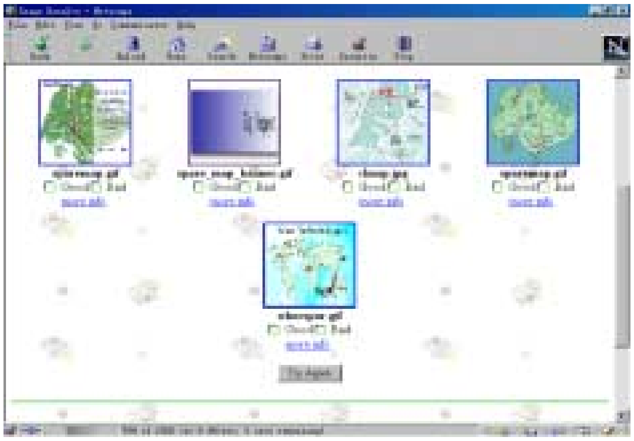
Match scale explores the importance of match order in the lexical chain. It has the effects in terms of image ranking during presentation of the returned images. Images with higher similarity measures will be returned to users ahead of images with lower similarity values. Figure 6 shows a sample results obtained from Q1. As shown, by considering the match scale, we can get more relevant images being displayed earlier, i.e., ranked higher.

6.3 On Feedback Mechanisms

In this experiment, we study the effectiveness of the two proposed feedback mechanisms: *semantic accumulation* and *semantic integration and differentiantion*. Figure 7 shows the improvement by the two methods respectively. *Opt* is the basic Weight ChainNet model without feedback. *Accu* denotes the semantic accumulation method. And *I&D* represents the semantic integration and differentiation method. We note that Accu and I&D represents one application of the feedback loop after Opt returns its resultant images.



(A) Q1 results before Applying Match Scale



(B) Q1 Results after Applying Match Scale.

Figure 6: Image Results for Q1

From Figure 7, we can see that both methods have improved the precision very much, especially for semantic I&D. We also observe that semantic I&D outperforms Accu. Two reasons account for this. First, in Accu, the whole feedback image ChainNet is used for refining the query. Some noise may be introduced. Second, Accu did not remove those unrelated image from the results. Furthermore, semantic I&D integrates the most relevant LCs in each ChainNet. These LCs do not carry much noise at all. We would like the reader to bear in mind that the comparison is baised against Accu in the sense that Accu employs only one feedback image, while I&D employs several.

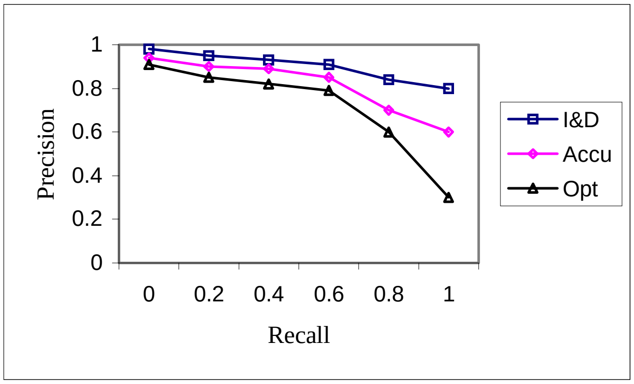


Figure 7: Comparison of feedback mechanisms

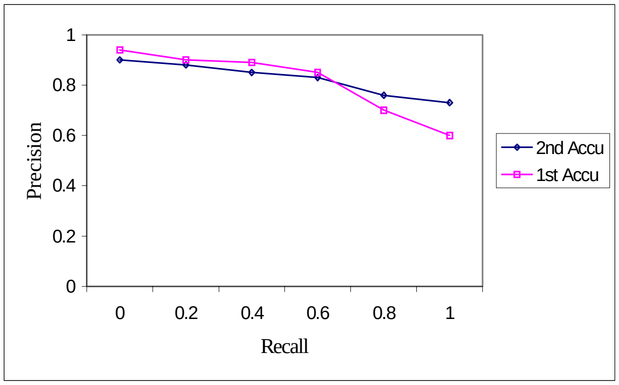


Figure 8: 1st and 2nd try by Semantic Accumulation

To clearly see the effect of the noise that semantic accumulation brought, Figure 8 presents the results for the first feedback and second feedback by semantic accumulation. We can see that the second try on the feedback actually has a bit lower precision, but with relatively higher reacall. But semantic accumulation method has the advantage that the returned image are more semantically related to the specific image selected by user - the feedback image.

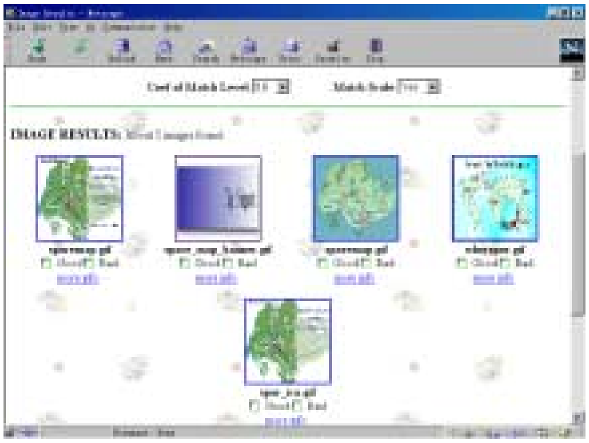


Figure 9: One-step feedback of Accu for Q1.

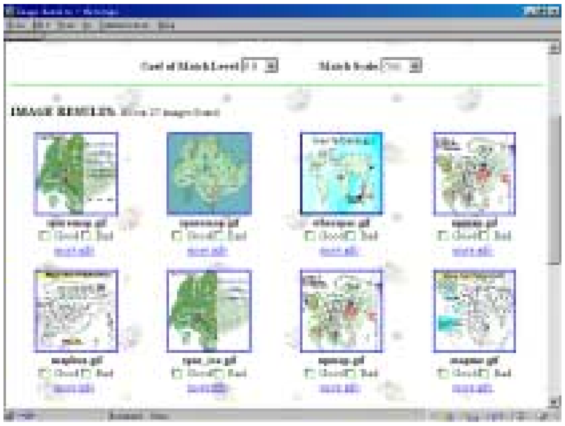


Figure 10: One-step feedback of I&D for Q1.

Figure 9 shows a sample feedback run of the Accu method for Q1. Compared to the results generated from OPT (the basic Weight ChainNet model without feedback), we see that the set of images retrieved are more relevant.

Figure 10 shows a sample feedback run of the I&D method for Q1.From earlier results without feedback mechanism (see Figure 6), we have identified two relevant and one irrelevant image. As shown in Figure 10, the resultant images are not only more relevant than OPT and Accu approaches, the irrelevant image has also been pruned. In addition, more relevant images have been retrieved.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a new model to represent the content of images embedded in WWW pages. The proposed Weight ChainNet model combines different types of lexical chains obtained from the surrounding text of an image. Our experimental study showed that the approach can be used as an effective means to represent image semantics. We also proposed two novel feedback mechanisms. In particular, the semantic integration and differentiation method returned more accurate results than semantic accumulation with higher recall. We plan to extend this work in the following ways. First, since we are mainly concerned with the object and event, it may be helpful to guess the lexical chain meaning by applying AI techniques. We are currently looking into some of these techniques. Second, the proposed approach is essentially an Information Retrieval (text-based) approach. We plan to integrate with content-based retrieval methods that capture the visual content of the images. Finally, we are exploring the use of query expansion mechanism [20,21] to enrich the content of the image, i.e., each LC is also expanded using WordNet.

8. ACKNOWLEDGMENTS

This work is part of the VIPER (Visual Property-based Search Engine for image Retrieval) project developed at the National Unviersity of Singapore (see http://unicorn.comp.nus.edu.sg). Discussions with and comments from members of the project team have been helpful.

9. REFERENCE

[1] S.Al-Hawamdeh, B.C. Ooi, R.Price, T.H.Tng, Y.H.Ang, and L.Hi, Nearest neighbor searching in a picture archival system. In *Proceedings of ACM International Conference on Multimedia and Information System*, Pages 17-34, 1991

[2] A.E. Cawkell, Imaging systems and picture collection management: a review. *Information Service & Use*, 12:301-325, 1992

[3] R.Price, T.S Chua, and S.Al-Hawamdeh, Applying relevance feedback on a photo archival system. *Journal of Information Science*, 18:203-215, 1992

[4] J.R. Smith and S.F. Chang. Image indexing and retrieval based on color histograms. In *Proceedings of ACM Multimedia '96*, Pages 87-98

[5] T.S. Chua and W.C. Low. Image retrieval using multiple features and domain knowledge. In *proceeding of International Symposium on Multimedia Information Processing*, Dec, 1997, Pages 543-548

[6] W.Niblack, R.Barber, and W.Equitz. the qbib project: querying images by content using color, texture, and shape. *Technical report, IBM* RJ 9203(81511), Feb, 1993

[7] Greg Pass, Ramin Zabin, and Justin Miller. Computing images using color coherence vectors. In *the Fourth ACM Internaltional Multimedia Conference*, 1996, pages 65-73.

[8] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1): 11-32, 1991

[9] T.S. Chua and W.C. Low, and Ch.X. Chu, relevance feedback techniques for color-based image retrieval. In *Proceeding of Multimedia Modelling'98, IEEE Computer Society*, Oct, 1998.

[10] Frisse M.E, (1988). Searching for information in a hypertext medical handbook. *Communications of the ACM*, 3 I(7), pp.880-886.

[11] Frei H.P, and Stieger D. (1992). Making use of hypertext links when retrieving information. *Proceedings ACM-ECHT'92*, Milan, Italy, pp. 102-111.

[12] Croft W.B., and Tutle H.R. (1993). Retrieval strategies for hypertext. *Information Processing & Management*,29(3), pp. 313-324.

[13] Dunlop MD, and Van Rijsbergen C.J,"Hypermedia and free text retrieval", *Information Processing & Management,* 29(3), 1993, Page. 287-298.

[14] Agosti M., and Smeaton A, (1996). Information Retrieval and Hypertext, *Kluwer Academic Publishers*, The Nether-lands.

[15] Harmandas, M. Sanderson and M. D. Dunlop, Image retrieval by hypertext links, *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, 1997, Pages 296 - 303

[16] Dunlop M.D. (1991). Multimedia Information Retrieval, Ph.D. Thesis. *Computing Science Department, University of Glasgow, Report* 199l/R21.

[17] J.Morris and G.Hirst, Lexical Cohesion Computed by Thesaural Relation and an Indicator of the Structure of Text, *Computational Linguistics,* vol.17, no.1, 1991, Page 22-48.

[18] Shih-Fu Chang, William Chen, and Hari Sundaram, Semantic Visual Template - Linking Visual Fetures to Semantics. *IEEE Intern Conference on Image Processing, Chicago IL*, Oct 1998

[19] G.A. Miller, R. Beckwith, C. Felbaum, D. Gross and K. Miller, Introduction to WordNet: An On-line Lexical Database. Revised Version 1993.

[20] Ellen M. Voorhees and Yuan-Wang Hou, "Vector Expansion in a Large Collection", First Text REtrieval Conference (TREC-1), 1993.

[21] E.M.Voorhees, "Query Expansion using Lexical-Semantic Relations.",ACM-SIGIR,1994.

[22] K.L. Tan, B.C. Ooi and C.Y. Yee, "An Evaluation of Color-Spatial Retrieval Techniques for Large Image Databases", Multimedia Tools and Applications, accepted for publication.

[23] B.C. Ooi, K.L. Tan, T.S. Chua and W. Hsu, "Fast Image Retrieval Using Color-Spatial Information", VLDB Journal, Vol. 7, No. 2, 115-128, May 1998.