Integrating Textual Ontology and Visual Features for Content Based Search in an Invertebrate Paleontology Knowledgebase

By

Yang Tian

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____

Dr. Bo Luo, Chairperson

Committee members

_____

Dr. Fengjun Li

_____

Dr. Guanghui Wang

Date Defended: _____

The Thesis Committee for Yang Tian certifies that this is the approved version of the following thesis:

Integrating Textual Ontology and Visual Features for Content Based Search in an Invertebrate Paleontology Knowledgebase

Dr. Bo Luo, Chairperson

Date Approved: _____

Acknowledgements

I would like to thank my advisor Dr. Bo Luo for his help on my research and study at University of Kansas. During our research on IPKB project, he provided many useful ideas and available approaches on both technical and non-technical problems. Also, I have learnt a lot by taking his courses such as Database Systems and Information Retrieval. All I learnt from his courses help me broaden my eyesight, gain more ideas, and also be able to go deeper in my research topic. He is always happy to teach me how to conduct research and how to bring new ideas into current jobs.

I would like to thank Dr. Guanghui Wang for being another major guider in this research project. His deep knowledge in computer vision and image processing brought many new ideas into our original focused research area. Besides, he also shared with me some techniques of writing papers, which is quite helpful and useful in my current and future work.

I would like to thank my teammate Ranjith Sompalli, who works together with me on this research project. Without his help and effort we cannot achieve what we have right now.

I would also like to take this opportunity to thank Dr. Fengjun Li for being members of my graduation committee and guide my research.

Finally, I really appreciate what all faculty, staff and students in EECS Department, ITTC and University of Kansas have done for me in these two years, to make me a person who is always proud of being a Jayhawker in my future life.

Abstract

The Treatise on Invertebrate Paleontology is a definitive work completed by more than 300 authors in the field of Paleontology, covering all categories of invertebrate animals. The digital version for the Treatise is consisted of multiple PDF files, however, these files are just a clone of paper version and are not well formatted, which makes it hard to extract structured data using only straightforward methods. In order to make fossil and extant records in the Treatise organized and searchable from a web interface, a digital library which is called Invertebrate Paleontology Knowledgebase (IPKB) was built for information sharing and querying in the Treatise. It is consisted of a database which stores records of all fossils and extant invertebrate animals, and a web interface which provides an online access.

The existing IPKB system provides a general framework for the Treatise's information showing and searching, however, it has very limited search functions, only allowing users querying by pure text. Details of structural properties in the fossil descriptions are not carefully taken into consideration. Moreover, sometimes users cannot provide correct and rich enough query terms. Although authors of the Treatise are all paleontologists, the expected users of IPKB may not be that professional.

In order to overcome this limitation and bring more powerful search features into the IPKB system, in this thesis, we present a content-based search function, which allow users to search using textual ontology descriptions and images of fossils. First, this thesis describes the work done by previous research on IPKB system. Except for the original text and image processing approaches, we also present our new efforts on improving these original methods. Second, this thesis presents the algorithm and approach adopted in the construction of content-

based search system for IPKB. The search functions in the old IPKB system did not consider the differences among morphological details of certain regions of fossils. Three major parts are discussed in detail: (1) Textual ontology based search. (2) Image based search. (3) Text-image based search.

Contents

LIST OF FIGURES ..... **viii**

LIST OF TABLES ..... **x**

Chapter 1. Introduction **1**

1.1 Introduction to the Treatise on Invertebrate Paleontology 1
1.2 Introduction to Invertebrate Paleontology Knowledgebase 1
1.3 Introduction to Content-based Search 2
1.4 Related Research 4

Chapter 2. IPKB System **6**

2.1 Text Processing 6
2.2 Image Processing 9
2.2.1 Segmentation Fossils and Labels 11
2.2.2 Detection of Fossil contours 13
2.2.3 Association of Fossil Objects and Sub-labels 14
2.2.4 Recognition of Sub-label Images 15
2.3 Web Interface 16
2.3.1 Searching 17
2.3.2 Displaying of Genera List and Genera Records 19
2.3.3 Browsing 20
2.5 Conclusion and Problems 22

Chapter 3. Textual Ontology and Visual Features Based Search **23**

3.1 Overview of Content-based Search 24
3.2 Textual Ontology based Search 25
3.2.1 Tagging the Morphological Descriptions 26
3.2.2 Extracting region names of fossils 27
3.2.3 Generating textual ontology of fossils 28
3.2.4 Calculating Similarity Scores 29
3.3 Image based Search 30
3.3.1 Preprocessing of Image Data Set 31
3.3.2 Visual Feature Extraction 32

3.3.2.1 Estimation of Cell Orientation 34
3.3.2.2 Calculation of Local Histogram 37
3.3.2.3 SHELO Composition 37
3.3.2.4 Square Root Normalization 38
3.3.3 Dimensionality Reduction and Group Comparison 38
3.4 Text-Image based Search 39
3.4.1 PTFIDF 40
3.4.2 Integration of Textual Ontology and Visual Features 41
Chapter 4. Experimental Results and Analysis 42
4.1 Textual Ontology based Search 42
4.1.1 Experimental Settings 42
4.1.2 Scoring Mechanism Demonstration 43
4.1.3 Textual Ontology based Search Results 44
4.2 Image based Search 46
4.3 Text-Image based Search 49
4.3.1 Experimental Settings 49
4.3.2 Results and Analysis 51
4.3.3 Summary 53
Chapter 5. Conclusion and Future Work 55

LIST OF FIGURES

Figure 1.1: Overview of the IPKB system 2
Figure 1.2: System Overview of Content-based Search 4
Figure 2.1: Example of Genus Vallites 7
Figure 2.2: Example of Family Homoceratidae 8
Figure 2.3: Semi-structured Results 9
Figure 2.4: A Sample of Extracted Images 10
Figure 2.5: Image Processing in the Treatise 12
Figure 2.6: The distance matrix 14
Figure 2.7: Homepage of IPKB System 16
Figure 2.8: Web Interface for Advanced Search 18
Figure 2.9: Genera list returned by quick search 20
Figure 2.10: Details of genus Admixtella 21
Figure 3.1: Examples in the image data set of IPKB system 31
Figure 3.2: General framework of RST-SHELO 33
Figure 3.3 Examples of image preprocessing 34
Figure 3.4: Pixel (i, j) and its corresponding neighbor cells 36
Figure 3.5: Local histogram calculation with 2*2 blocks 37
Figure 3.6: Two fossil images for the same genus 39
Figure 4.1: Web interface for textual ontology based search 43
Figure 4.2: Fossil images for Lingula (a), Barroisella (b), Semilingula (c), and Credolingula (d) 43
Figure 4.3: Similarities between genera 44
Figure 4.4: Top 2 results of advanced search using “small and rounded to subtriangular shell” . 45
Figure 4.5: Top 3 results of ontology search using “shell”-“small and rounded to subtriangular” 46
Figure 4.6: Image retrieval results using hand drawn sketches as queries 47
Figure 4.7: Image retrieval results using photos as queries 48

Figure 4.8: Structure of tables for PTFIDF features 50
Figure 4.9: Structure of table irstshelo_features 51
Figure 4.10: The query Image 51
Figure 4.11: Top 3 results of text-image based search 52
Figure 4.12: The last two results on the first page in textual ontology based search 53

LIST OF TABLES

Table 1: Morphological Descriptions for Genus Pseudolingula and Lingulepis 26
Table 2: Examples of POS tagging 26
Table 3: Results of NNP_GLO tagging on genus Pseudolingula 28
Table 4: Textual Ontology of genus Lingula 29

Chapter 1. Introduction

1.1 Introduction to the Treatise on Invertebrate Paleontology

Invertebrate paleontology, which is also called invertebrate paleozoology or invertebrate paleobiology, is a discipline which conducts scientific study of prehistoric invertebrates by analyzing invertebrate fossils in the geologic record [2]. Covering all categories of these invertebrate fossils, including phylum, class, order, family and genus, the Treatise on Invertebrate Paleontology [1] is a definitive compilation with 50 volumes and multi-authored by more than 300 experts in this field. It is agreed that the Treatise is the most authoritative definition book of invertebrate fossils and can be found in almost every good library worldwide.

In PDF version, the Treatise is separated into 23 different parts from Part A to Part W with multiple volumes in each part. Our work is mainly focused on Part H, Part B, Part L and Part O, which contains 10 volumes in total. Looking inside of these volumes, we find that fossils are categorized in a hierarchical way. All of them are described in paragraphs. Although rules are made to restrict how to organize descriptions of fossils, there are still quite a lot of violations. Paragraphs are mixed together with images, tables and their titles. Fortunately, there are very little tables in each volumes thus they can be removed manually. However, there are a lot of images in each volumes which are need to be extracted and processed. All these properties of the Treatise bring about challenges to build an organized and searchable digital library for all fossil records in these volumes of the Treatise.

1.2 Introduction to Invertebrate Paleontology Knowledgebase

Facing all problems of the Treatise mentioned above, an Invertebrate Paleontology Knowledgebase system was constructed with database and web interface. Figure 1.1 describes the

general structure of the IPKB system. The system contains several major parts: text processing, image processing, and web interface. The PDF Files are the data source of this system. For our current IPKB system, it contains totally 10 volumes from Part H, Part B, Part L and Part O. Text and Images are processed separately and then the extracted and parsed data are stored in our database. In this figure we omit the details of web interface which will be discussed in next sections.

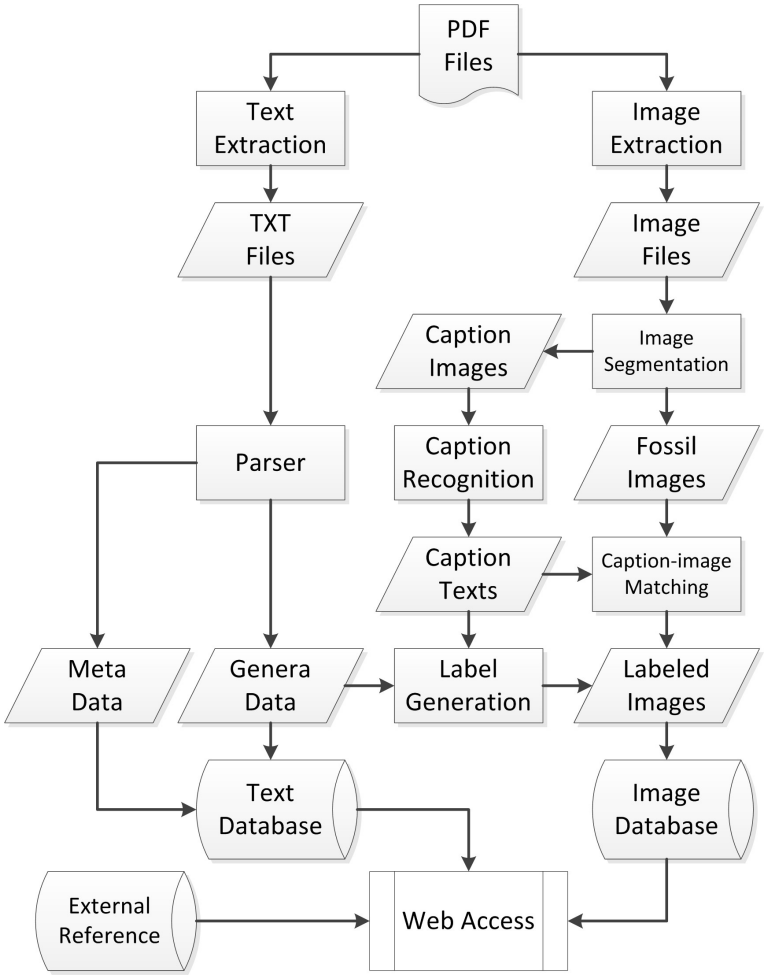


Figure 1.1: Overview of the IPKB system [3]

1.3 Introduction to Content-based Search

In previous work, IPKB system only has search functions using only pure text queries. In many conditions, it cannot meet users' requirements. In the description paragraph for each genera, there

are property details of the shell, ventral, dorsal or even others. If a user query with a string “small shell, round circular ventral area”, his/her intension is to search for fossils with small shells and round circular area. However, when using straightforward text searching, the system only returns list of genus which contain these query terms in their descriptions in a certain rule, without considering the dependency between property names(“shell”, “ventral”) and property details(“small”, “round circular”). The returned results are definitely not expected by the users. In order to solve this problem, we present a textual ontology based approach, which is capable of extracting textual ontology information based on properties for each genera in the database, and also based on extracted ontology data, calculating similarities using a newly designed algorithm.

Moreover, sometimes one cannot describe a fossil in words correct and exact. The textual ontology query descriptions may not rich enough for searching. In this situation, using fossil images to improve the results is a good option. However, we cannot completely rely on images. It is not easy for even professionals to identify the genera of a fossil through only several photos of it. Therefore, the purpose of using images is to rank genera which are visually similar to the query images higher than others.

Figure 1.2 demonstrates the overall framework of content-based search, which is a new part of IPKB system. It aims at solving two major problems mentioned above and gaining improvement over old search functions by integrating textual ontology and image information. Three major components: textual ontology based search, image based search and image-text based search are outlined with different colors. Notice that although image search is not reliable in practical, the algorithm of feature extraction and iRST-SHELO set generated in this part are still used in image-text based search.

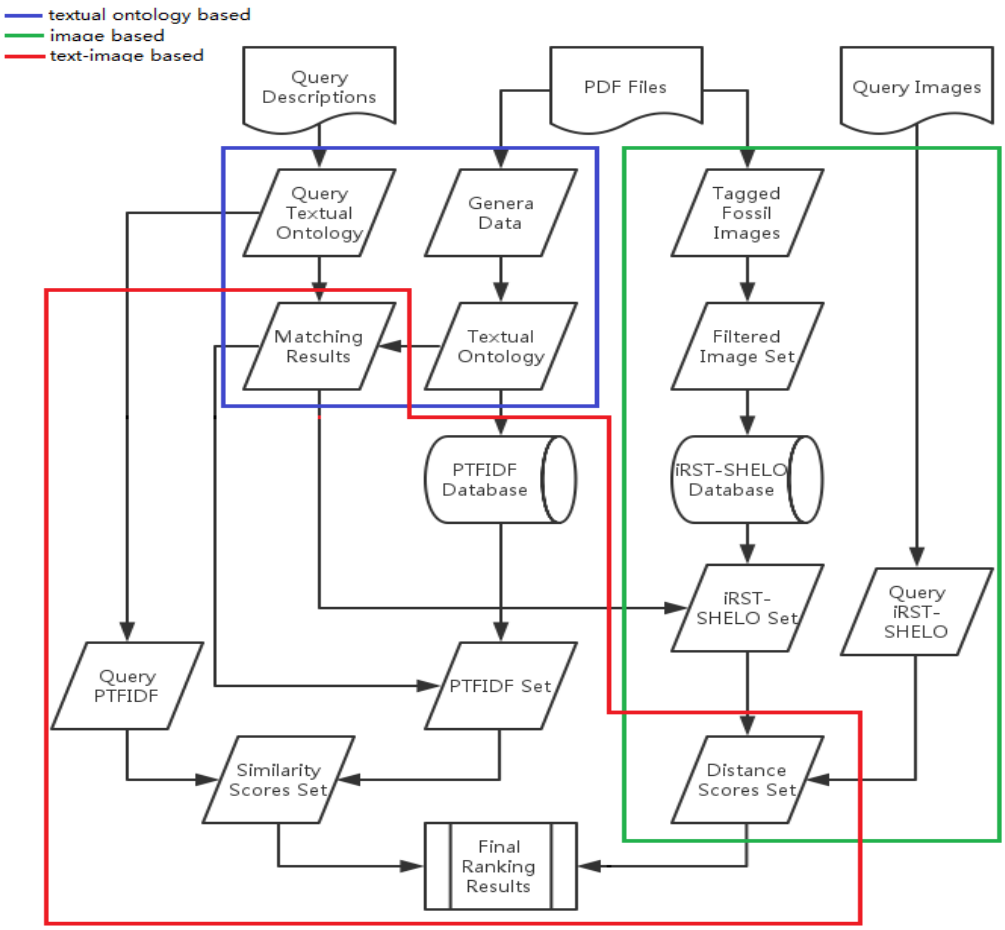


Figure 1.2: System Overview of Content-based Search

1.4 Related Research

A digital library is a special library which focus on collecting digital data in different format. Looking back to 1966, we can found an early digital library called Education Resources Information Center [4]. Traditional libraries are limited by storage space; digital libraries have the potential to store much more information, simply because digital information requires very little physical space to contain it [5]. A large number of digital libraries focus on specialized area and provide search functions, such as ACM Digital Library [6] and Digital Himalaya [7].

A digital library is actually a type of information retrieval system. Tf-idf is a classical and effective weighting factor for information retrieval and text mining, however, sometimes it is not

enough. In the work done in [8, 9] proposes techniques to identify semantically dependent terms in a corpus. Except for text information retrieval, image retrieval is also a very hot topic. Some traditional image retrieval approaches use metadata to annotate images, such as [10], [11], and [12]. Instead of doing image annotation, content-based image retrieval (CBIR) has been extensively studied recent years. Research approaches explained in [13] make use of some content-based image features such as SIFT [14], GIST [15] and HOG [16]. Also there is some work done [17, 18] in the field of sketch based image retrieval, which is an interesting branch of CBIR. Sometimes features generated for a single view cannot represent a sample very well. Facing this challenge, [19, 20, 21, 22] focus on fusing or integrating features in different views, for example, features from text view and image view.

Chapter 2. IPKB System

In Figure 1.1, we describe the overall framework of IPKB system. From this figure we can see how the system is constructed. Given PDF files of the Treatise as the data source, we extract Text part and Images part separately. Adobe Acrobat itself provides a function which allow us to separate text and images in PDF files. However, it brings about errors into extracted paragraphs. Therefore, we use a third party library Apache PDFBox (https://pdfbox.apache.org/) to extract text which avoids a lot of problems caused by the Adobe Acrobat's extracting function. In this section, we discuss the details of several major components which are built upon extracted text and image data.

2.1 Text Processing

In order to parse and process text part in the Treatise, we need to look into the content of these text. Part H contains fossil records under the Phylum "Brachiopolda", all of which have been parsed and imported to the database of existing IPKB system. In our research, we enlarge the database using data generated from Part B (Phylum "Charophyta"), Part L (Phylum "Mollusca") and Part O (Phylum "Arthropoda"). What we interested in in the Treatise's text part is entries of taxon records, such as Order, Class, Family and Genera.

Genera is the basic taxon and for each genus record in the text part, we can find a paragraph description which starts with the name of this genus and the following are organized in a restricted order. Figure 2.1 shows the structure of a genus. From this example, we can see several fixed parts:

- Genera name ("Vallites").
- Author information ("RUZHENTSEV & BOGOSLOVSKAIA, 1971, p. 20").
- Reference ("[*Homoceras Henkei SCHMIDT, 1934, p. 453; OD]").

- Morphological description (“Adult whorls with … shorter than ventral lobe. Seven species.”).
- Geological Information (“Pennsylvanian …Uzbekistan (Fergana).”).
- Indices of Figures (“FIG. 53,5a-d … Bogoslovskaia, 1971).”). From the example we can see that there might be multiple figures for each genus.

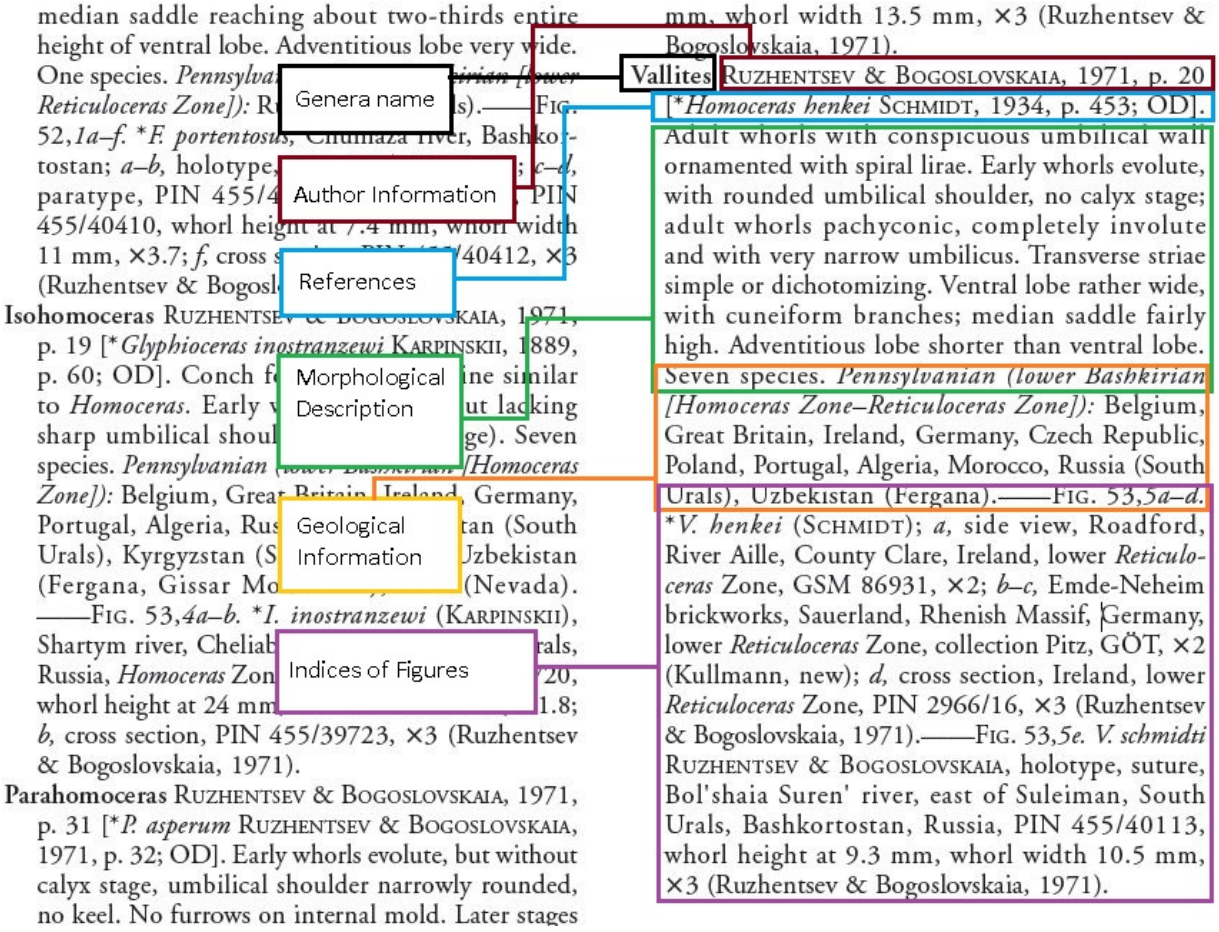
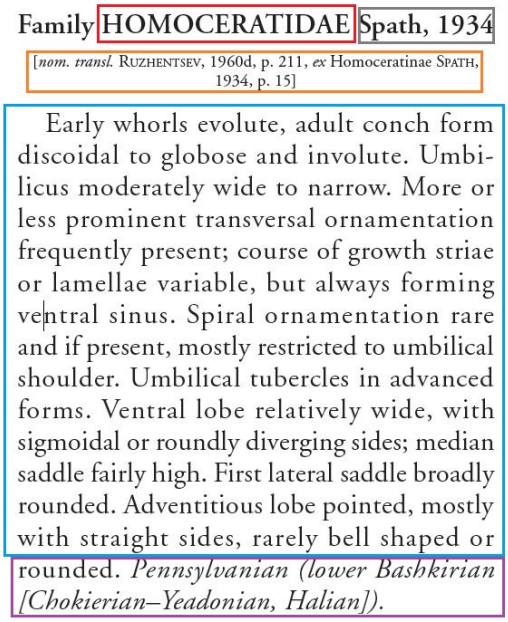


Figure 2.1: Example of Genus Vallites

Genera names start with a capital letter which is followed by several letters in lower case. Then the author name after the genera names are all capital letters. This is a special property which can be used to recognize the start point of a genus record. The reference part is put inside a square bracket “[]”. The geological information part is consisted of a fix range of terms thus can be easily

located. It is simple to find the start point of indices of Figures because it always starts with “FIG. XX”.

Except for the basic taxon genera, we also need to parse data of upper layer taxa such as order, family, etc. These taxa also follow certain orders. For example, in Figure 2.2, the family is consisted of several pieces of fixed information such as family name, author, reference, description and geological information. Also notice that before parsing these taxa from text part, we need firstly clean up unwanted words and lines in the text, for example, page numbers, page headings, figure titles and sub-labels for images in figures.


Family HOMOCERATIDAE Spath, 1934

[*nom. transl.* Ruzhentsev, 1960d, p. 211, *ex* Homoceratinae Spath, 1934, p. 15]

Early whorls evolute, adult conch form discoidal to globose and involute. Umbilicus moderately wide to narrow. More or less prominent transversal ornamentation frequently present; course of growth striae or lamellae variable, but always forming ventral sinus. Spiral ornamentation rare and if present, mostly restricted to umbilical shoulder. Umbilical tubercles in advanced forms. Ventral lobe relatively wide, with sigmoidal or roundly diverging sides; median saddle fairly high. First lateral saddle broadly rounded. Adventitious lobe pointed, mostly with straight sides, rarely bell shaped or rounded. *Pennsylvanian (lower Bashkirian [Chokierian–Yeadonian, Halian]).*

Figure 2.2: Example of Family Homoceratidae

Considering all issues discussed about, we use techniques based on regular expression matching to implement a Java program which is capable of parsing unstructured genera and other taxa data and then store them in semi-structured XML format. Figure 2.3 demonstrates our approach by giving out a semi-structured genus (a) and a semi-structured family (b).

```
<Genus name="Micragnostus">
<phylum>ARTHROPODA</phylum>
<subPhylum>TRILOBITOMORPHA</subPhylum>
<class>TRILOBITA</class>
<subClass>UNKNOWN</subClass>
<order>AGNOSTIDA</order>
<subOrder>AGNOSTINA</subOrder>
<superFamily>AGNOSTOIDEA</superFamily>
<family>AGNOSTIDAE</family>
<subFamily>AGNOSTINAE</subFamily>
<authorInfo>HOWELL, 1935b, p. 233 </authorInfo>
<hyperLinkInfo>[*Agnostus calvus LAKE, 1906, p. 23; OD; lectotype (LAKE, 1906, pl. 2, fig. 18; SD
FORTEY, 1980b, p. 23), A548, SM, Cambridge]. </hyperLinkInfo>
<description>En grande tenue; non- scrobiculate; border furrows nondeliquiate; acro- lobes
unconstricted; median preglabellar furrow absent. Glabella with semiovate anterior lobe; F3
straight; posterior lobe parallel-sided with F2 weakly developed or absent; glabellar culmination
broadly rounded; glabellar node level with or slightly behind F2. Basal lobes small. Pygidium
minutely bispinose; axis relatively short, uncon- stricted or weakly constricted over M2; M1
trilo- bate; F1 traversing axis; F2 well impressed, slightly deflected by axial node. Posterior
lobe semiovate. Median postaxial furrow absent. </description>
<timePeriod>Upper Cambrian Lower Ordovician (Arenig): Wales, Spitsbergen, AsaphellusD. bifidus
Zones; Argentina, N. argentinaK. meridionalis Zones; Sweden, M. dalecarlicus Zone.</timePeriod>
<figureIndex>Fig.
219,4.*M.calvus(LAKE),LowerOrdovician(Tremadoc),northernWales(NantRhosDdu);lectotype,SMA548,10(new).
</figureIndex>
</Genus>
```

(a)

```
<Family name="PTYCHAGNOSTIDAE">
<authorInfo>Kobayashi, 1939 </authorInfo>
<hyperLinkInfo>[Ptychagnostinae KOBAYASHI, 1939a, p. 151] [=Triplagnostinae KOBAYASHI, 1939a, p.
145; Lejopyginae HARRINGTON in KOBAYASHI, 1939a, p. 128; Tomagnostinae KOBAYASHI, 1939a, p. 148;
Canotagnostidae RUSCONI, 1951, p. 13] </hyperLinkInfo>
<description>Usually en grande tenue, with nondeli- quiate border furrows and unconstricted
acrolobes; usually with median preglabellar furrow and elongate basal lobes; axial glabel- lar
node variable in position. Pygidium usu- ally simplimarginate, rarely weakly zonate, nonspinose or
bispinose; with basic articulat- ing device; axiolobate with F1 furrow well developed, F2 furrow
rarely absent; posterior axial lobe usually long, acuminate or rounded posteriorly, occasionally
with a transverse depression in anterior half; me- dian postaxial furrow usually present.
</description>
<timePeriod>Middle Cambrian lower Upper Cambrian.</timePeriod>
</Family>
```

(b)

Figure 2.3: Semi-structured Results. (a) Genus “Micragnostus”. (b) Family “Ptychagnostidae”.

2.2 Image Processing

Images we extracted from PDF files of the Treatise are grayscale photos of fossils which are taken from different angles. In each image, for example, in Figure 2.4 from Part B, there are multiple fossil objects, each of which are tagged by a sub-label such as “1a”, “2b”, etc. There are also other non-label words (for example, “Marshallites”) mixed together with these objects and

labels. The purpose of Image Processing is to divide this kind of collection of fossil objects into smaller images of single fossil objects. These smaller images should be tagged by its corresponding sub-labels. Remember that when we parsed genera information in text processing step, the indices of figures for each genus list figure number and sub-labels for this genus. Thus these tagged smaller images can be correctly and efficiently put into their corresponding genus groups.

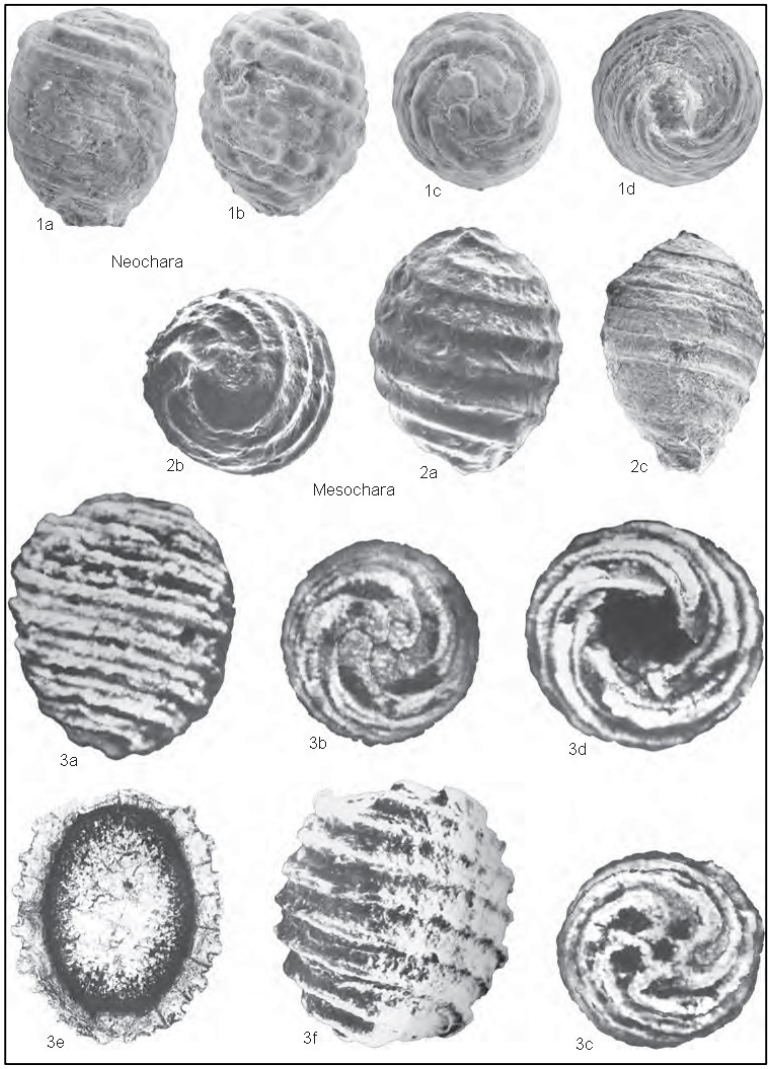


Figure 2.4: A Sample of Extracted Images

In order to make the task of image processing automatic, the previous work designed an approach consisted of several major steps:

- Segment fossil objects and labels in the image.
- Detect contours of fossil objects.
- Associate each fossil object with its corresponding sub-label image.
- Recognize words in sub-label images, and tag fossil objects with recognized labels.

In the following we will present them in detail and propose our improvement on these steps.

2.2.1 Segmentation Fossils and Labels

As shown in Figure 2.4, fossils are exhibited in a blank background and mixed with tagging words. In previous research, an approach has been implemented to separate fossils objects apart. In this section we will describe this approach step by step using Figure 2.4 as the example.

In order to mark fossil objects and the background which contains sub-labels and other words, we use the opening by construction method. It is a morphological operation consisted of an image erosion step and then an image reconstruction [23] step. The purpose of image erosion here is to erode all non-fossil objects in the figure. Since sub-labels and other words inside the figure is usually much smaller than fossils, we use a disk-shaped structuring element with a diameter larger than the height of characters and smaller than the size of fossil objects. For figures in Part B, this diameter is set to 15. In image reconstruction step, the eroded image is used as the marker image and the original image is used as the mask. After that, we execute a dilation operation on the reconstructed image. Using the complement of this dilated result as the mask, and the complement of the previous reconstructed result as the marker, we execute another reconstruction operation followed by a complement operation. By converting this result to a binary image, we get all fossil objects masked by black pixels and other parts with white pixels (see Figure 2.5(b)).

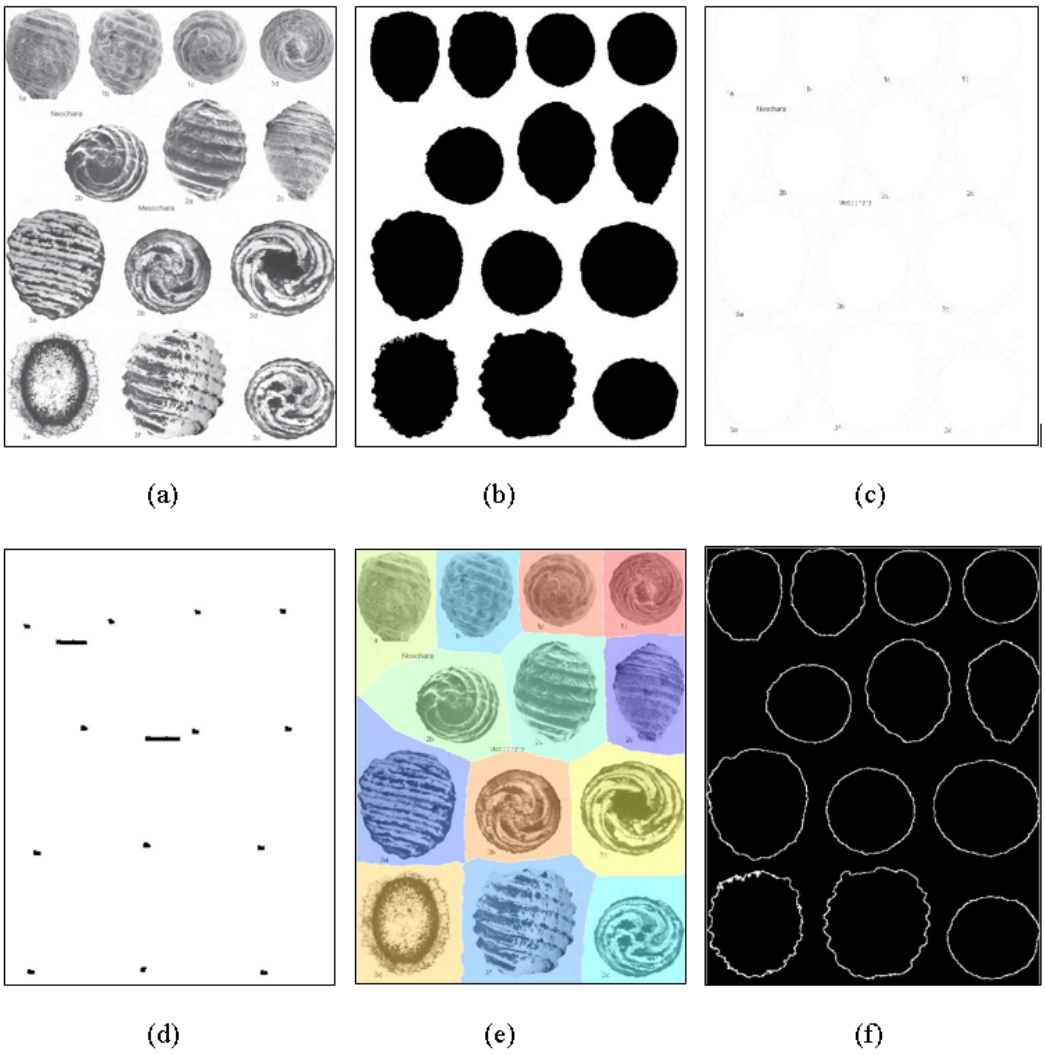


Figure 2.5: Image Processing in the Treatise. (a) An figure extracted from Part B. (b) fossils are masked. (c) Sub-labels and other words with the background. (d) Blocks of sub-labels and words identified. (e) Areas of fossils divided using watershed approach. (f) Contours of fossil objects.

Since fossil objects have been marked and identified, we need to identify sub-labels after that. By using the image generated in Figure 2.5(b) we can mask out all fossils from the original figure, then get the result in Figure 2.5(c). Then through a morphological opening operation followed by a converting-to-binary operation, we recognize blocks of sub-labels and other words (Figure 2.5(d)). Since words other than sub-labels are always much longer, we can easily remove

those longer blocks and keep only sub-labels we need. Remember that one of our purposes is to partition fossil objects in a big figure into individually small images. For that purpose, we use watershed algorithm which aims to find local minima in a grayscale image and then partition them into pieces [24]. Figure 2.5(e) shows the result of watershed approach executed on our image of masked fossils. These separated pieces of single fossils are stored in JPG files for future use.

2.2.2 Detection of Fossil contours

In the segmentation step of image processing, we have already masked all fossil objects in the figure. However, in order to accurately calculate the distances between fossils and sub-labels in future steps, we need to detect contours for all fossils. Contour detection, or edge detection has already by extensively studied in the field of image procession and computer vision. Since figures in our application are just sample grayscale and 2D, we can choose solution from classical algorithms, among which the sobel operator works very well for our figures.

Sobel operator is used very often in edge detection algorithm. This operator uses two 3 by 3 kernel matrices to calculate the gradient for image pixels. These two kernels, one for horizontal direction and another for vertical direction, are convolved with the original image as follows:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A.$$

where A denotes the source image, G_x and G_y denote two images in which every point contains its corresponding gradient in horizontal direction and vertical direction respectively, * denotes a 2D convolution operation. Combining G_x and G_y together, the magnitude of the gradient can be calculated using:

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

In our case, sobel operation is executed upon mask image of fossils which is actually a binary image. Therefore, for all non-edge pixels in this image, the element of G is simply 0. In this way we can generate a contour image efficiently as Figure 2.5(f).

2.2.3 Association of Fossil Objects and Sub-labels

Each fossil object and its corresponding sub-label is expected to be placed most closely to each other in the original figures. However, sometimes this situation cannot be satisfied. The previous research designed a method to calculate the distances between fossil objects and sub-labels and decide the association between each fossil object and its corresponding sub-label. Considering such a case, a sub-label S is put between two fossils A and B and has the similar distances to two of them. At this time, if we can decide the association between A and another sub-label in the figure, then we can definitely assign sub-label S to fossil B. This is so-called bipartite graph matching problem, which means to find a perfect matching with minimum distance.

At first we calculate the distance between each fossil object and each sub-label. The distance here is defined as:

$$Distance(A, s) = \min_{i=1 \to M} d(c(s), a_i)$$

where M is the number of pixels for the contour of fossil A, c(s) is the center pixel of sub-label block s and a is a pixel of the fossil contour. The d (.) stands for Euclidean distance. Collecting these distances together, we constructed a minimum distance matrix as in Figure 2.6.

Labels \ Fossils	1	2	3	4	5	6	7	8	9	10	11	12
1	17.493	47.043	100	100	100	100	100	100	100	100	100	100
2	100	19.235	64.498	100	100	100	100	100	100	100	100	100
3	100	100	21.19	100	100	100	100	100	100	100	100	100
4	76.217	45.706	100	18.028	68.622	100	100	100	100	100	100	100
5	100	100	76.968	100	21.587	74.465	100	100	100	100	100	100
6	100	100	100	100	100	17.464	100	100	100	100	100	100
7	100	100	100	88.482	100	100	16.971	98.732	100	100	100	100
8	100	100	100	100	100	100	100	23.77	100	100	100	100
9	100	100	100	100	100	100	100	25	100	16.971	100	100
10	100	100	100	100	100	100	100	100	17.493	100	100	100
11	100	100	100	100	100	100	100	100	31.385	100	18.385	100
12	100	100	100	100	100	100	100	69.318	100	100	50.537	16.031

Figure 2.6: The distance matrix

In this matrix, distances larger than a threshold 100 is set to 100. These distances are ignored and two sides of it (fossil no. and label no.) will not be associate. We only considering distances less than the threshold. Looking through the columns one by one, we find that column 10 and 12 only contain one distance less than 100 respectively. Thus we can associate fossil 10 with sub-label 9, fossil 12 with sub-label 12, and then remove row 9 and 12 from the matrix. By repeating this scanning and removing operation, we can finally make all fossils linked to their corresponding sub-labels. The matrix in Figure 2.6 is built upon a figure from Part H. In other parts of the Treatise, the threshold value may be different. For instance, for figures in Part B, it is better to set threshold to 50 rather than 100.

2.2.4 Recognition of Sub-label Images

In the previous step, sub-label blocks are linked to fossil objects. In order to tag these fossil objects with true sub-labels, we need to recognize the text information in sub-label images. All sub-labels in images are small in size and consisted of limited characters and numbers such like "1a", "2b", "3c", etc. Therefore, we prepared some template sub-label images for recognition. Among all the distances between the sub-label image and these templates, the smallest one is the match. The distance here is calculated using sum of square error (SSE) as follows:

$$\boldsymbol{SSE} = \frac{\sum_{(x,y)} \left(\boldsymbol{I_{temp}(x,y) - I_{targ}(x,y)} \right)^2}{\sum_{(x,y)} \boldsymbol{I_{temp}(x,y)^2} + \sum_{(x,y)} \boldsymbol{I_{targ}(x,y)^2}}$$

In the above formula, $I_{temp}(x, y)$ and $I_{targ}(x, y)$ are the pixel locations in template image and the sub-label image respectively. In most cases, the size of the template image and sub-label image are different. Thus we need to try multiple positions for comparison using a sliding window. Remember that in text processing part, we parsed genera information with their corresponding indices of figures. Thus after assigning recognized sub-label and figure index to a fossil object, we

can easily get all fossil objects for a genus by only using a genus name.

2.3 Web Interface

Through text processing and image processing, we obtain a large amount of semi-structured data. These semi-structured data cannot be used directly thus we write a Java problem to convert them into structured data and import these data into database. In order to build a web interface which allows users to browsing and searching fossil information conveniently, we have built a three-level architecture in which the database in built by MySQL, the Web server is built by Apache and PHP, and the client side is implemented using HTML, CSS and JavaScript. The home page of this website can be seen in Figure 2.7 below.

Figure 2.7: Homepage of IPKB System

2.3.1 Searching

The most important function of IPKB system is searching. In previous research, we designed and implemented two types of search function: quick search and advanced search. The search results displayed by the web interface is actually the list of genera and in each page, there are at most 10 items shown. In this section we will discuss details of searching functions.

Quick search allows users to input a query string which is a genus name, family name, or a class name, etc. The searching results are list of genus which match this query on genus name, family name, class name, etc. Quick search also support approximate comparison. The algorithm is as follows:

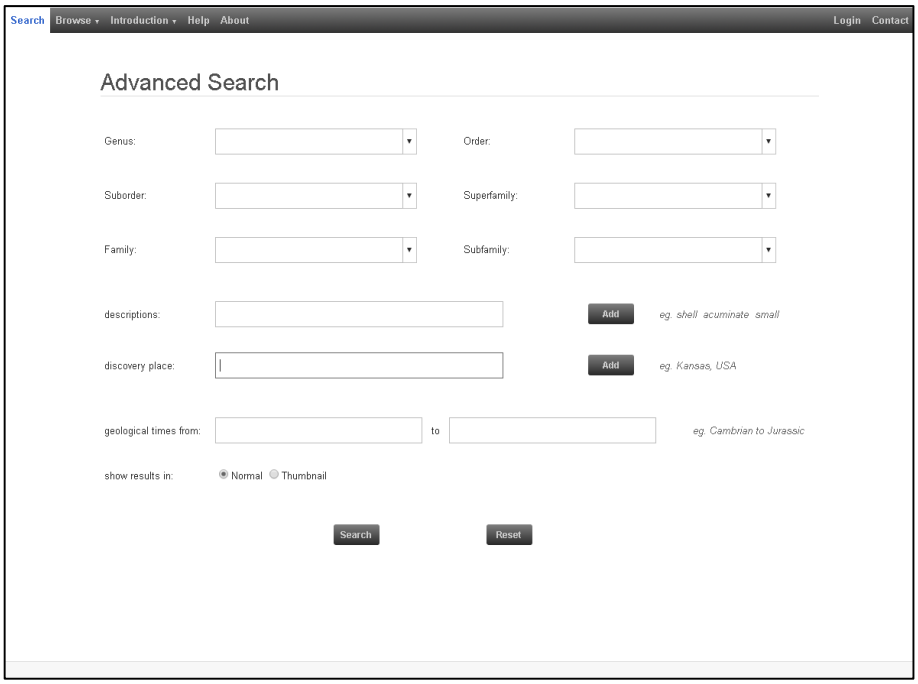
- Search for exact match of genus name.
- If there is no match, execute a full-text search over upper-layer taxon names.
- If there is no match, search for approximate match of genus name.
- If there is still no match, terminate the search process.

Exact match results are entries which contains the same keyword(s) in the query string. Approximate matching means partial matching. For example, “paras” is an approximate match of “Paraschartymites”. The search results are ranked by relevance.

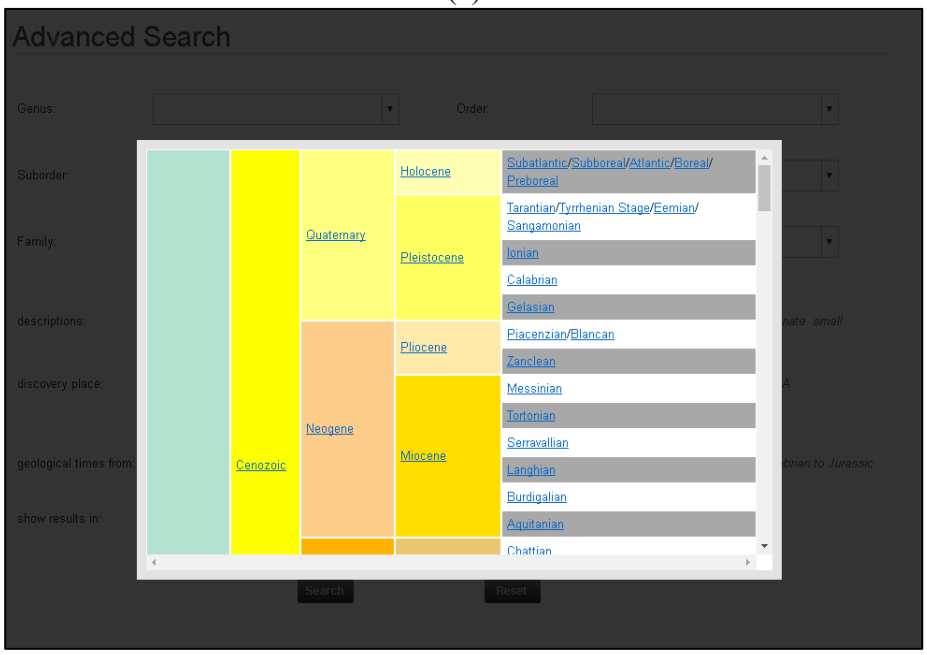
In advanced search, users can input their queries in a more flexible way. They can input the taxon names, descriptions or geological information separately, and advance search module will take in all these queries and return ranked relevant results.

The fossils in our database are dated from lower Cambrian period (approximately 520 million years ago) to now. Users do not need to input the period by themselves. Instead, the advanced search function provides a board which allow users to choose period from a fixed number

of options. Figure 2.8 (a) shows the webpage of advanced search module, and in Figure 2.8(b) we can see the board for time period selection.



(a)



(b)

Figure 2.8: Web Interface for Advanced Search

A full-text search algorithm has been proposed for both quick search and advanced search.

In the database, the relevant columns are included in a FULLTEXT index. Each column are given a weight value to represent its importance in the search context. In our algorithm, we set 3.0 to taxon names, 1.5 to geological information and 1.0 for morphological descriptions. The idea of full-text search comes from TF-IDF weighting approach. The significance of a term can be computed using:

$$s = \frac{\log(dtf) + 1}{\sum(\log(dtf) + 1)} \times \frac{U}{1 + 0.0115U} \times \log\left(\frac{N - nf}{nf}\right)$$

where dtf denotes the term frequency, which means the appearing times of a term in a database record, $\Sigma\ (\log(dtf) + 1)$ denotes the sum of (log (dtf) +1)'s for all words in the same database record. U is the number of unique words in the database record, *N* is the total number of database records and nf is the number of database records containing this term. The combined relevance of a term and database record can be calculated using:

$$R = w \times s \times qf$$

where w is the weight value set to columns previously, s is the significance calculated above, qf is the term frequency in the query input. For most terms the qf is just 1 which means this term only appears once in the query. Terms which are less than 4 letters will be ignored by full-text search. A list of stop words is also given to removes terms appearing too many times.

2.3.2 Displaying of Genera List and Genera Records

After quick search or advanced search, a list of genera will be generate by the web interface. No more than 10 genera items are shown on each page of search results. Figure 2.9 shows the quick search results with query string "admi". From this list of results we can see brief information about each genus. By clicking the genus name at the up left corner of each record, one can view this genus' detail page. Similarly, by clicking likes of upper-layer taxon names which are listed under the genus name, one can see the detail page of the corresponding taxon. Other information such as morphological description and geological information are also shown on the webpage.

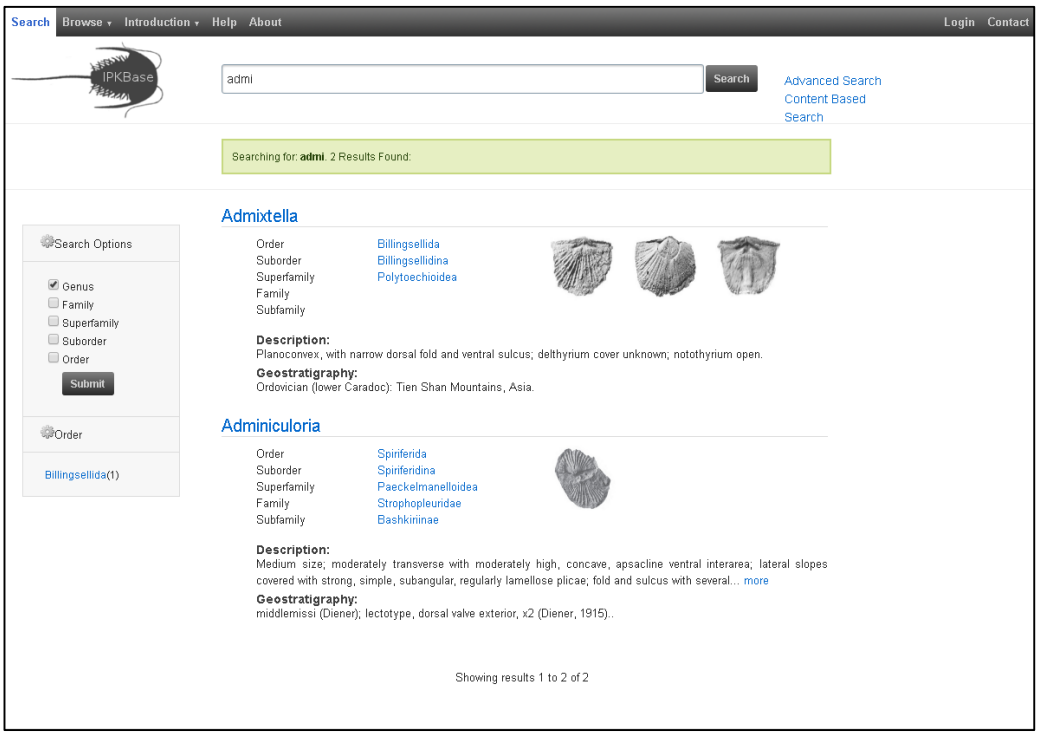


Figure 2.9: Genera list returned by quick search

By clicking the first record “Admixtella” in Figure 2.9, the detail page of this genus can be seen in Figure 2.10. From this page we can see more the complete information of genus Admixtella. Images we generated in image processing step are shown in this page together with their sub-label and indices of figures. Notice that the map under the “Geology” section shows discovery locations of this genus. This function is implemented using Google Earth API. In our application, we firstly map each location name in our database to exact longitudes and latitudes, which are acceptable by Google Earth API, then use these coordinates to mark all locations on the map.

2.3.3 Browsing

The browsing module provides an overview of hierarchical structure of all fossil categories in our database. This webpage has a scroll bar on the left which contains a tree-structure list of orders. By clicking an item in this list, details of this order will be loaded to the right side of the webpage. At the same time, the sub-categories of this order will appear under this order in the left list, and so on. By using browsing function, if a user know the categories of a genus, he/she can

quickly locate this genus and see its details without inputting any query words.

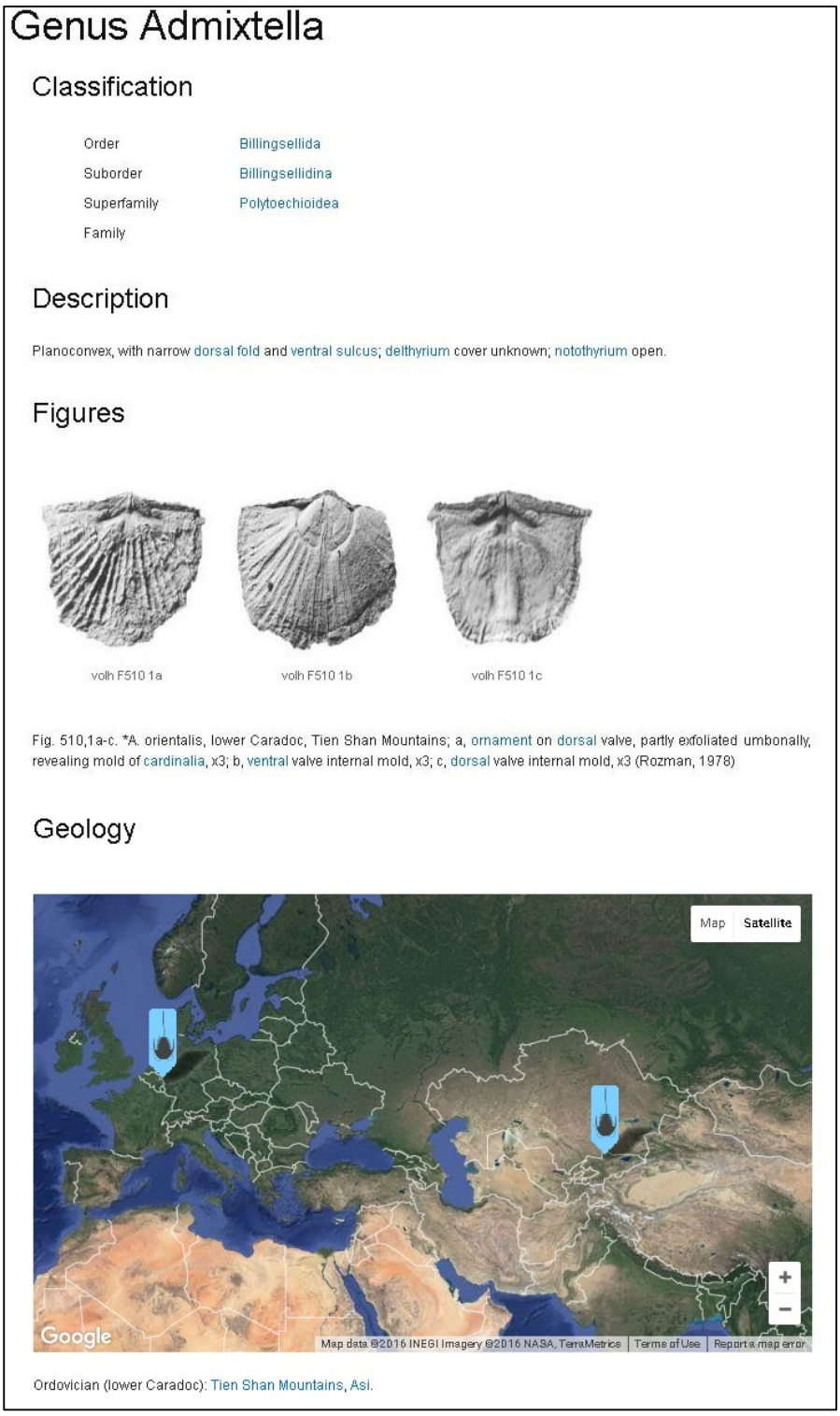


Figure 2.10: Details of genus Admixtella

2.5 Conclusion and Problems

Invertebrate Paleontology Knowledgebase (IPKB) is a project which targets at building a digital library for the Treatise of Invertebrate Paleontology. In this chapter we present the process of extracting and parsing text and image data in the the Treatise, the approach to build the whole system, and the design and implementation of the web interface.

In text processing part, efforts are taken to analysis parsing content of paragraphs and convert unstructured text to semi-structured data. In image processing part, we presented an approach to segment images, identify labels and associate fossil objects and labels together. In web interface part, we use several examples to demonstrate our user interface of quick search, advanced search, genera lists and browsing function.

In this phase, IPKB provides a general framework of searching and browsing invertebrate fossil records, which allow users to input text queries and obtain results (Or without input queries, for instance, browsing). However, there is still spaces for us to improve its search function. In the next chapter, I will discuss our efforts on improve the search function in IPKB system using textual ontology information and visual features.

Chapter 3. Textual Ontology and Visual Features Based Search

In chapter 2 we describe the framework and details of IPKB, a digital library for the Treatise, which provides various functions such as searching and browsing. However, the search function in this old IPKB system are only text based and completely depending on the query strings input by users. In many conditions, this type of search functions cannot satisfy users' intention.

First, let us consider such a scenario, a user inputs a query string "small shell, round circular ventral area". The intention here is to search for fossils with small shells and a round circular shaped ventral area. When using quick search and advanced search functions described in chapter 2, the system just compares the query string with the whole database record of each genus, without considering these properties separately. For example in our database, a morphological description for a genus contains a piece "round circular shell, small ventral area". It is a perfect match for the query string in quick search and advanced search, however, it is not the result expected by the user.

Moreover, as we mentioned previously, there are a large number of complicated terms in the field of invertebrate paleontology, which are hard to correctly spell them even for professional or experts who have experiences in this field over years. The text query information given by users may not be correct and rich enough for the system to return reasonable results.

Facing up with such limitations existing in the old IPKB system, we propose a content-based search function, which integrates textual ontology information and image information to improve the search results and overcome challenges discussed above. In this chapter we firstly give an overview of this content-based search module and then present all design and implementation details in this part.

3.1 Overview of Content-based Search

Figure 1.2 shows the overview of content-based search function. There are three major components, textual ontology based search, image based search and text-image based search in this function. For simplicity we omit the process from PDF files to Genera Data and Tagged Fossil Images, which is described more clearly in chapter 2. Notice that the process of generating PTFIDF Database and iRST-SHELO Database are executed offline.

Textual ontology based search allows users to input multiple pairs of a region name of a fossil (for example, "ventral") together with the structural description of this region (for example, "round circular"). The results returned by this function are fossil records which have similar structural properties in corresponding regions. In order to implement such a function, we need first delve deep into morphological descriptions for fossils. Region names and their corresponding region descriptions should be extracted from the unstructured description paragraphs for each genus in the database. These region-description pairs are the major factor we considered in our proposed textual ontology based search. There are several steps involved in this search function:

- Tagging the morphological descriptions.
- Extracting region names of fossils.
- Generating textual ontology of fossils.
- Calculating similarity scores.

All these steps will be explained in detail later in this chapter.

In image-based search, users can provide multiple images as the query input. A query image can be a photo of fossils with a light background or a picture of hand drawn sketch on a blank canvas. Using all fossil images in our database and query images, we designed a way to select out

images can be used to represent a genus and then extract visual features, which are used to compare similarities between images and rank the search results. There are totally for major steps in this module:

- Image set preprocessing.
- Extraction of visual features.
- Dimensionality reduction.
- Group comparison.

Image-based search did provide an option which accepts non-text query input. However, it is impossible to reply purely on images to search for genera. Given several photos of a fossil and without any other hints, one cannot easily identify its genus even if he/she is a professional paleontologist or has experience in this field over years. Also, there are more than 18,000 images in our database. Comparing all of them with the query images are slow and impractical for our application. The results of textual ontology-based search can be used to narrow our search range and improve the speed of searching.

Therefore, we propose a text-image based search which is capable of integrating textual ontology and image information to improve the search results. Two types of features PTFIDF and iRST-SHELO are designed and generated in order to gain benefits from both data source: textual ontology and fossil images. In the following, we will present three components of content-based search in detail.

3.2 Textual Ontology based Search

There are four steps discussed in this section. The first three steps aim at generating textual ontology for fossils and the last step calculates similarities based on these textual ontologies.

3.2.1 Tagging the Morphological Descriptions

The task of this part is to mark each term in the description sentences with a special tag. Stanford NLP tagger API [25] provided a tagging approach which can be used to mark each term with its part of speech. We use some examples to demonstrate our approach. Table 1 shows original descriptions for two genera from Part H of the Treatise. From this table we can see that sentences are separated by semicolons thus it is easy to divide these sentences into parts. For each sentence, we execute POS tagging approach to tag all terms. Several examples of POS tagging results for sentences in genus Pseudolingula's morphological description are listed in Table 2.

Genus Name	Pseudolingula	Lingulepis
Morphological Description	**Shell elongate subrectangular**; ventral pseudointerarea with narrow, deep pedicle groove; **ventral visceral area extending anteriorly beyond midvalve**; oblique muscle scars placed on broadly divergent ridges; dorsal visceral area bisected by broad median septum, extending from beak to anterior lateral muscle scars; **dorsal central and anterior lateral muscle scars placed close together**; vascula lateralia of both valves short, submedian, slightly converging; vascula media absent; inner surface of both valves with conspicuous wrinkling directly lateral to vascula lateralia.	Shell strongly acuminate, subtriangular, unequivalved; ventral beak strongly elongate; pseudointerarea elongate triangular, with narrow pedicle groove and flexure lines; dorsal pseudointerarea poorly known; ventral visceral area short, not extending to midvalve; dorsal visceral area slightly thickened, extending to midvalve, bisected by two divergent ridges; ventral vascula lateralia strongly arcuate.

Table 1: Morphological Descriptions for Genus Pseudolingula and Lingulepis.

Sentence	Tagged Terms
"Shell elongate subrectangular"	shell/NNP elongate/JJ subrectangular/JJ
"ventral visceral area extending anteriorly beyond midvalve "	ventral/JS visceral_area/NNP extending/VBG anteriorly/RB beyond/IN midvalve/NN
"dorsal central and anterior lateral muscle scars placed close together"	dorsal/JS central/JS and/CC anterior/JS lateral/JS muscle/NNP scars/NNP placed/VBN close/RB together/RB

Table 2: Examples of POS tagging

Although there are quite a number of technical terms in our data, the Stanford NLP tagger still works because it support approximation. With all words tagged, we are able to go further into the next step.

3.2.2 Extracting region names of fossils

In this step we need to identify non-independent terms in sentences. Non-independent terms means terms which are related to each other semantically and usually contains important information. For example, in our corpus, "small" is frequently used to describe "shell", thus these two words can be said to be non-independent. The research in [26] demonstrates that the most frequently occurring pairs of adjacent Noun + Noun and Noun + Adjective terms can be defined as semantically related to each other. Based on this conclusion, we extract all pairs of N + N and Adj + N pairs for future use.

The research in [26] also provides a U-test approach which can be used in our paleontology corpus to identify related or non-independent terms. U-test approach calculates a u-score between two terms using the following formula:

$$|u| \approx \left| \frac{nC(w,w') - C(w)C(w')}{\sqrt{nC(w)C(w')}} \right| > u_{\alpha/2}$$

where n is the number of sentences in the corpus, w and w' stand for two terms, C (w) gives our the number of sentences in which term w appears, $u_{\alpha/2}$ is the α/2 quantile of standard normal distribution. α is a significant level. When $|u| > u_{\alpha/2}$, w and w' are non-independent terms, otherwise they are said to be not semantically related to each other.

Using this approach, we calculate u-scores for all previously extracted pairs of N + N terms and Adj + N terms separately. By setting α to 0.01, we pick out all pairs with scores larger than $u_{\alpha/2}$ as our non-independent terms. These extracted term pairs are tagged with a new shortcut "NNP_GLO". There are totally 471 term pairs generated from Part H of the Treatise and the expected fossil region names such as "shell", "visceral area", etc., can be found among these

"NNP_GLO" tagged terms. Table 3 shows some results after NNP_GLO tagging.

Genus Name	Pseudolingula
Sentences in Morphological Description	Shell elongate subrectangular ventral pseudointerarea with narrow deep pedicle groove ventral visceral area extending anteriorly beyond midvalve oblique muscle scars placed on broadly divergent ridges dorsal visceral area bisected by broad median septum extending from beak to anterior lateral muscle scars …
NNP_GLO Results	shell/NNP_GLO elongate/JJ subrectangular/JJ ventral/JS pseudointerarea/NNP_GLO with/WITH narrow/JJ ,/, deep/JJ pedicle_groove/NNP_GLO ventral/JS visceral_area/NNP_GLO extending/VBG anteriorly/RB beyond/IN midvalve/NN oblique/JJ muscle_scars/NNP_GLO placed/VBN on/IN broadly/QL divergent/JJ ridges/NNP_GLO dorsal/JS visceral_area/NNP_GLO bisected/VBN by/BY broad/JJ median/JS septum/NNP_GLO ,/, extending/VBG from/FROM beak/NNP_GLO to/TO anterior/JS lateral/JS muscle_scars/NNP_GLO …

Table 3: Results of NNP_GLO tagging on genus Pseudolingula

3.2.3 Generating textual ontology of fossils

Through observing fossil records after NNP_GLO tagging, we found that most sentences in a genus description describe regions or sub-regions about the shell, dorsal view or ventral view. These regions and sub-regions can be described using solely adjectives or related to other regions of this fossil. Based on above observations, we use the following algorithm to identify and extract textual ontologies:

- Identify "with" and "and" in sentences to separate long sentences to shorter ones.
- Set all NNP_GLO tagged terms which start with "shell", "ventral" or "dorsal" as region (sub-region) names.
- The adjective which occurs before a region (sub-region) name is considered as part of this region (sub-region) name.
- All terms follow the region (sub-region) name is considered as the decryption for

this region (sub-region).

The extracted fossil ontology gives out its most prominent region names together with sets of terms structurally describing their corresponding regions. Table 4 shows an example of extracted textual ontology. Notice that all textual ontologies are stored in JSON format in order to facilitate the calculation of similarity scores between two genera.

Genus Name	Lingula
Morphological Description	Shell elongate oval to subrectangular; ventral pseudointerarea with wide triangular pedicle groove; ventral visceral area extending to midvalve, with impression of pedicle nerve curving around unpaired umbonal muscle scar; dorsal interior with visceral area extending somewhat anterior to midvalve; dorsal central and anterior lateral muscle scars closely spaced, bisected by weak median septum; vascula media absent.Shell of living species poorly mineralized laterally.
Textual Ontology	shell :elongate oval to subrectangular ventral pedicle_groove :wide triangular ventral visceral_area :extending to midvalve ventral impression of pedicle_nerve :curving around unpaired umbonal muscle_scar dorsal visceral_area :extending somewhat anterior to midvalve dorsal central muscle_scars :and closely spaced , bisected by weak median_septum dorsal anterior lateral muscle_scars :and closely spaced , bisected by weak median_septum dorsal vascula_media :absent

Table 4: Textual Ontology of genus Lingula

3.2.4 Calculating Similarity Scores

With all textual ontology data prepared, the last step in our approach is calculating similarity scores between genera. We calculate four scores for each genera pair: shell score, ventral score, dorsal score and other score. The shell score is computed by comparing the similarity upon two genera's shell properties. The ventral scores and dorsal scores can be calculated as follows:

$$Score_{ventral(dorsal)} = \frac{\sum s\_sub_{ventral(dorsal)}}{C\_ventral(dorsal)}$$

where s_sub is the score of sub regions in ventral or dorsal view, the summation here is to sum up all similarity scores of sub regions, C_ is the total number of matching sub regions in ventral or dorsal view. Using the similar way of calculating ventral and dorsal scores, we can easily get other

regions' scores. Sum all four scores up and divide the summation by 4, we make the final score scaled to 0-1.

For two regions need to be compared with each other, the algorithm of calculating this sub score is as follows:

- If descriptions of two regions are exactly the same to each other, the sub score is 1.
- Else if descriptions of two regions have sub descriptions separated by commas in their sentences, then sub score = number of matching sub descriptions/number of all sub descriptions.
- Else if no matching sub descriptions, then sub score = number of matching terms/number of all terms.
- Else score is 0.

Similarity scores generated using above algorithm are always between 0 and 1. By comparing the similarities between the query input and textual ontologies, we can rank the search results based on similarity scores in a descending order.

3.3 Image based Search

Without the help of text information, one can hardly identify the genus of a fossil from a few photos unless he/she is a professional paleontologist or has years of experience in this field. Purely image-based searching cannot help us identify the genus of a fossil in most cases, since two fossils belongs to different taxa may look pretty similar to each other and the differences are too trivial to be identified by computer vision techniques. Therefore, the purpose of image-based search in our application is actually helping to search for genera with images which are visually similar to query images, rather than identifying the exact genus for the fossil in the query images. In this section, we discuss our techniques of implementing image-based search step by step. Image

data set and visual features set constructed in this module are also used in text-image based search module.

3.3.1 Preprocessing of Image Data Set

In the existing IPKB system, we have already built an image data set containing more than 30,000 images, each of which is tagged with its figure index and sub label so that we can easily identify its corresponding taxon. Through observing images in each data set, we found various types of images as photos of a fossil from taken from certain angle, photo-like images of a fossil, photos of a small region of a fossil, sketches, etc. We can see this diversity from Figure 3.1.

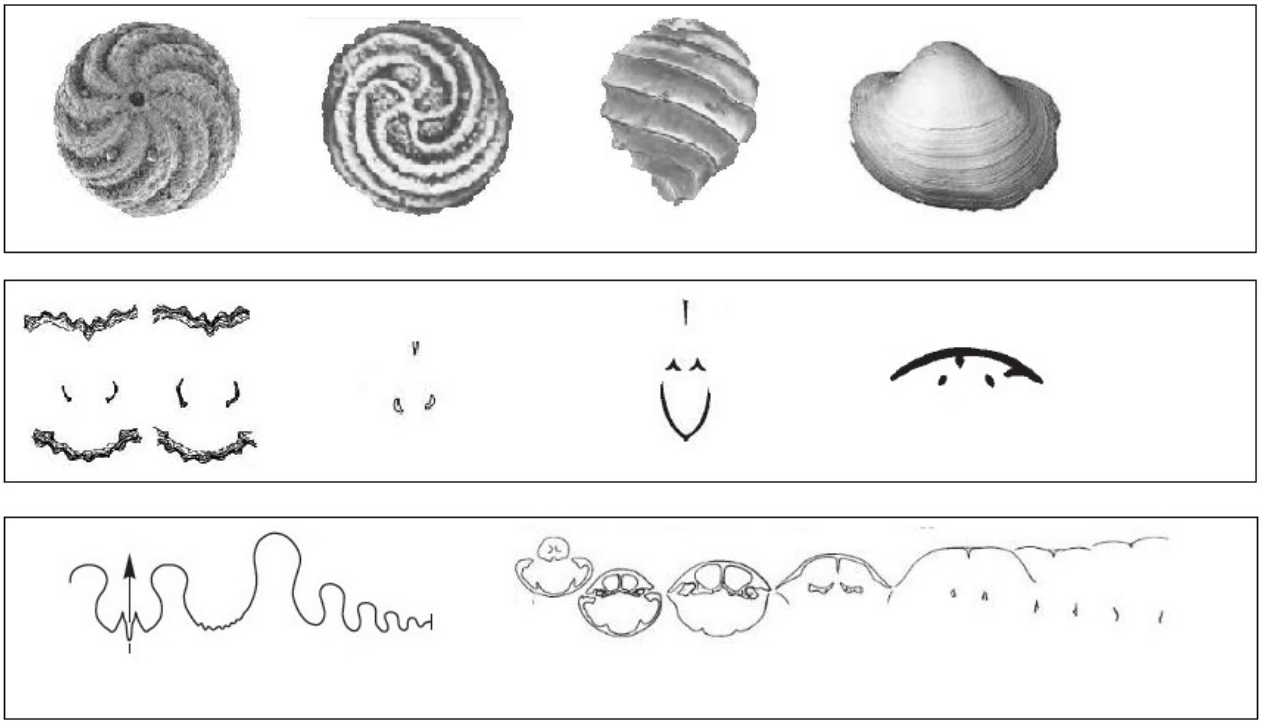


Figure 3.1: Examples in the image data set of IPKB system

In this figure, the fossil images in the first line is clear and complete enough to be selected as our candidate images. However, images in the second and third line, which are just very simple lines which cannot represent the shape or structure for a fossil. Visual features generated from such kinds of images are useless thus we should not use them for future image search. Therefore, we

designed a prepressing method to eliminate these useless images and collecting other images to build the image data set for image-based search and text-image based search.

Through observing images in Figure 3.1, we can easily identify the difference between useful and useless images. Useful images usually contains very rich information thus have more black pixels on canvas. On the contrary, useless images are sketch-like and have more white pixels on canvas. Based on this analysis, we first chop each image by cutting its margins which are consisted of solely white pixels at four directions. In this way we can keep the objects in the center of the canvas, which is convenient for our future calculation and comparison. The image set can be divided into sub set using:

$$im \in \begin{cases} G_1 & if\ t < \frac{M}{C}, \\ G_2 & otherwise. \end{cases}$$

where G_1 and G_2 denotes two sub set after division, M denotes the number of white pixels on the canvas, C is the number of all pixels in the image, and t is a threshold value.

After parameter tuning, we are able to put 90% of useful images into G_1, which is selected as our final image data set. We still need to do some manual work to put rest images into the final image set, but this division approach helps us save a lot of time.

3.3.2 Visual Feature Extraction

In this step we try to extract visual feature vectors to represent each image in our image data set. Since there are only grayscale images in our data set, we do not consider color issues. Instead, properties such as shape, contour (edge) and texture are more important and should be taken into account. In order to compare similarities based on all these properties or at least most of them, we try to get hints from the field of sketch-based image retrieval, which is a branch of content-based image retrieval.

Jose in [17] proposes a sketch-based image retrieval descriptor called RST-SHELO, which is an improved version of SHELO in [18]. The general process of RST-SHELO can be seen in

Figure 3.2. From this figure we can see that there are three major stages in RST-SHELO:

- Image preprocessing, which convert images into sketches used in next steps.
- Feature extraction, which contains a SHELO process and a normalization step.
- Similarity searching and ranking.

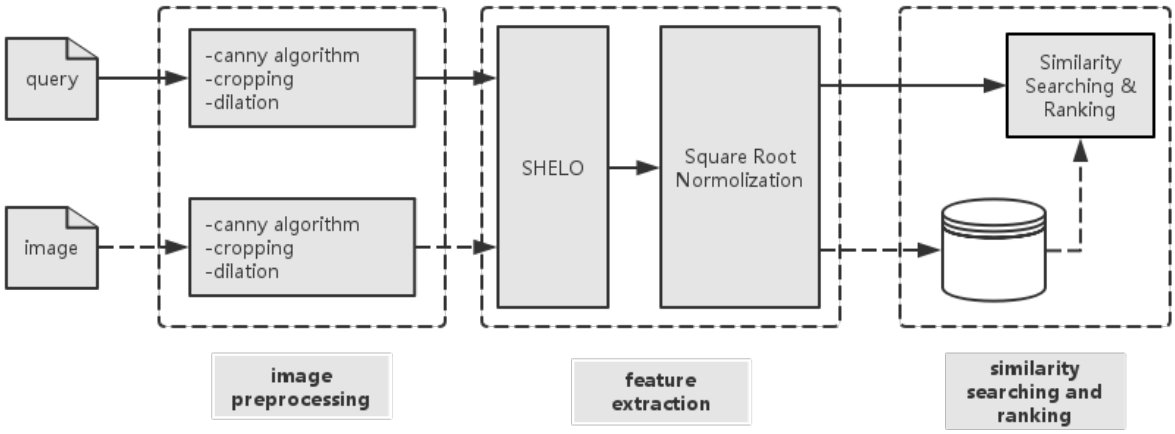


Figure 3.2: General framework of RST-SHELO

Notice that in [17] a Sketch Token algorithm is used for sketch generation in image preprocessing stage. In our application we chose to use canny algorithm instead, and there are two reasons for us to do so. First, Sketch Token approach aims at extracting contours or edges of objects in the image. Details inside the contours are usually disregarded by this approach. However, in our cases, texture inside the contour of a fossil is important information to represent this fossil. Second, Sketch Token, which extracts and combines multiple types of features from an image, is much more time-consuming than the straightforward canny algorithm. Therefore, in our approach, we use canny algorithm instead of Sketch Token approach.

In image preprocessing, we firstly use canny algorithm to get sketch and contour of the image. Then we crop all margins for this "sketch image" and remain the object in the center of the canvas. After that, we do a dilation operation to make these sketches more dramatic and facilitate the future feature extraction. Figure 3.3 shows two examples of our image preprocessing stage.

From this figure we can see that canny algorithm keeps both contours and some textures inside contours. After cropping, sketches of fossils are kept in the center of the canvas as expected.

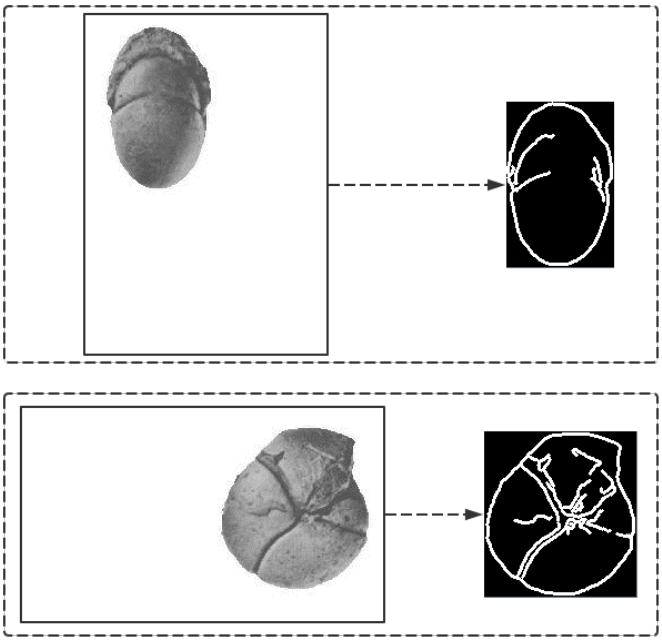


Figure 3.3 Examples of image preprocessing

Feature extraction is the most important step in RST-SHELO approach. It is consisted of two parts: SHELO descriptor generation and square root normalization. There are three major steps for SHELO descriptor generation:

- Estimation of Cell Orientation.
- Calculation of local histogram.
- SHELO composition.

In the following we will discuss details of these three steps as well as square root normalization.

3.3.2.1 Estimation of Cell Orientation

In this step, we first divide the preprocessed image into W*W cells. Since images may of different size, thus the size of cells varies depending on the size of the image. The orientation for

a cell is decided by all pixel orientations inside this cell together with its neighbor pixels' orientations. In order to generate an orientation for each cell, we start from deciding four neighbor cells for a pixel in the image. Suppose we have a pixel (i, j) in a preprocessed image. A pair of value x and y can be easily decided by:

$$x = \left(\frac{j}{N}\right) * W, y = \left(\frac{i}{M}\right) * W.$$

where M and N is the number of pixels in each row and column respectively. Since the image are divided into W*W cells, thus each cell can be labeled by a pair of value p and q, both of them are assigned values between 1 and W. After calculating x and y, we can generate four values as follows:

$$left_{position} = \lfloor x - 0.5 \rfloor$$

$$right_{position} = \lfloor x + 0.5 \rfloor$$

$$north_{position} = \lfloor y - 0.5 \rfloor$$

$$south_{position} = \lfloor y + 0.5 \rfloor$$

With these values, the four neighbor cells for the pixel (i, j) are cells labeled by $(left_{position}, north_{position})$, $(right_{position}, north_{position})$, $(left_{position}, south_{position})$ and $(right_{position}, south_{position})$. Notice that a neighbor cell can be the cell in which pixel (i, j) falls. Figure 3.4 shows the relationship between a pixel and its corresponding neighbor cells.

Since four neighbor cells are recognized for pixel (i, j), we further calculate a weight for each neighbor cell of this pixel. The algorithm to calculate these weights are described in the following:

- Calculate the distance from p to the most left side of the cell in which pixel (i, j) falls:

$$distance_x = x - \lfloor x \rfloor.$$

- If $distance_x < 0.5$, then:

$$left\_weight = 0.5 - distance_x,$$

$$right\_weight = 1 - left\_weight.$$

- Else:

$$right\_weight = distance_x - 0.5,$$

$$left\_weight = 1 - right\_weight.$$

- North_weight and south_weight can be calculated using value of y in the similar way discussed above.

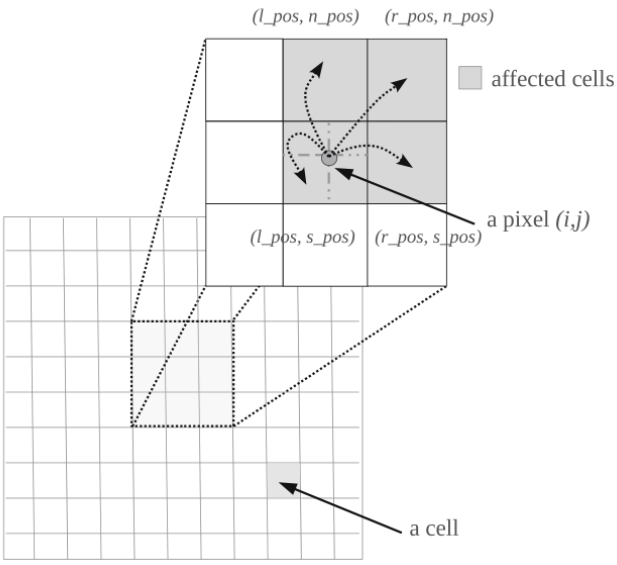


Figure 3.4: Pixel (i, j) and its corresponding neighbor cells

The cell orientation can be calculated as follows:

- Initialize two W*W matrices A and B to all zeros.
- For each pixel in the image, we decide its four neighbor cells.
- For each neighbor cell (x, y) of a pixel (i, j), we calculate the following values step by step:

$$c_x = \cos^2(\theta_{i,j}) - sin^2(\theta_{i,j}),$$

$$c_y = 2\cos(\theta_{i,j})\, sin(\theta_{i,j}),$$

$$A(x,y) += \eta_{i,j} * w_x * w_y * c_x,$$

$$B(x,y) += \eta_{i,j} * w_x * w_y * c_y,$$

- In above formulas, θ stands for the angle of gradient of a pixel in the image, η stands for the magnitude of the gradient, w is the calulcated weight for the neighbor cell.
- The orientation for cell (x, y) can be calculated using:

$$orentation(x,y) = 0.5atan2\big(A(x,y), B(x,y)\big).$$

Notice that the gradient of pixels are computed using sobel operator.

3.3.2.2 Calculation of Local Histogram

With all cell orientations generated, we need to construct a local histogram of them. In this step we execute another division operation on the image which separate pixels on the image into B*B large blocks rather than small cells. We build a K-bin histogram for each block's cell orientations. Just as a pixel has its neighbor cells, a cell has its corresponding neighbor blocks. We use the similar approach in calculation of cell orientation to compute the weights for bins (see figure 3.5).

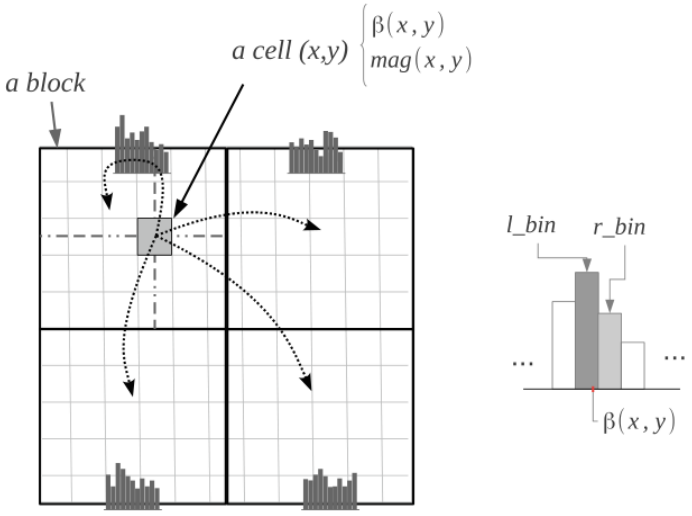


Figure 3.5: Local histogram calculation with 2*2 blocks

3.3.2.3 SHELO Composition

In the previous step, all local histograms are calculated. In this step, each one of them is

normalized to the unit and then the normalized histograms are conjuncted together to form a SHELO descriptor.

3.3.2.4 Square Root Normalization

The purpose of square root normalization is to avoid the calculated distances to be biased by large differences in a few dimensions of features. The idea of Hellinger kernel [28] is used here. By squaring root of each element of a SHELO feature, we can reduce part of penalties brought about by a few dimensions. The final features after this normalization are so-called RST-SHELO.

3.3.3 Dimensionality Reduction and Group Comparison

RST-SHELO features perform pretty well in representing fossil images in our database. However, the size for these feature vectors, which is 1296, are too large for our usage. In order to keep both correctness and efficiency of group comparisons in next step, we use principal component analysis [27] (PCA) to reduce the dimension of RST-SHELO descriptors. By setting the threshold to 80%, we are able to reduce the size of feature vectors from 1296 and 206. We call this descriptors after PCA iRST-SHELO.

All iRST-SHELO features for images in the image data set are generated offline and stored in the database for our website. The searching function in IPKB system requires the returned results to be a list of genus information, however, as we mentioned previously, most genera in our database owns multiple fossil images which show the appearances of fossils from different angles. Also, we allow the users to upload more than one images as query information. Therefore, we need to find a way to compare the similarities between groups of images. In many situations, entries in a group are pretty similar to each other, but not for our cases. For instance, Figure 3.6 shows two fossil images belong to the same genus in Part L. They looks totally different from each other because Figure 3.6(a) is the front view and Figure 3.6(b) is the side view.

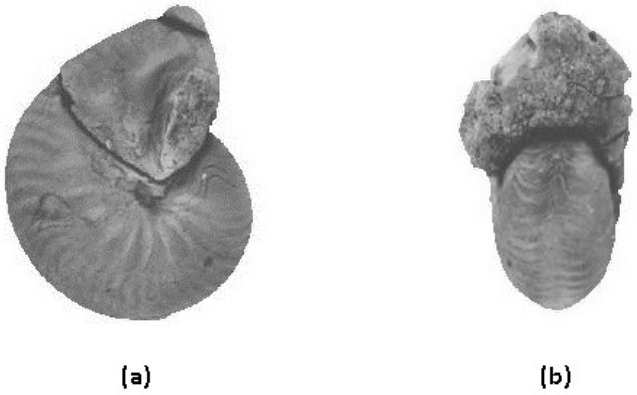


Figure 3.6: Two fossil images for the same genus

Considering the situation above, we can define the group distance. Groups in our image data set are actually sets of images for all genera. Distance between two groups in our application stands for the distance between the two closest entries from these two groups, with an entry from the first group and the other entry from the second group. We use a modified version of Heuristic Group Linkage Measure [29] for the distance calculation. The distance between two groups in our image data set can be computed using:

$$Distance(G_1, G_2) = \min\left(dist\left(I_{1i}, I_{2j}\right)\right),$$

$$\left(I_{1i}, I_{2j}\right) \in (G_1 \times G_2).$$

where G1 and G2 are two different groups, I stands for an image belongs to a group. The set of query images is considered as a query group, which is compared with groups of genera. The distance between two images are calculated using Euclidean distances between their iRST-SHELO descriptors.

3.4 Text-Image based Search

Textual ontology based search gives out results based on morphological descriptions of certain regions of fossils. Using this results together with the iRST-SHELO features extracted in image-based search, we can implement a text-image based search gaining both correctness and

efficiency which cannot be obtained at the same time in either pure textual ontology or pure image based search. The final list of results should be ranked based on information from both text and images. In this step, we generate a PTFIDF features using textual ontology information, and design an approach to integrate PTFIDF features and iRST-SHELO features together.

3.4.1 PTFIDF

Parallel TFIDF, or PTFIDF, is a modified version of classical TFIDF. TFIDF is a weighting technique based on term frequency and document frequency. Term frequency for a term in a document stands for the number of times this term appearing in this document. Document frequency for a term is the number of documents containing this specific term. Let $tf_{t,d}$ be the term frequency for term t and document d, df_t be the document frequency for term t, N be the total number of documents, then the TFIDF weighting for term t in document d can be calculated using:

$$w_{t,d} = tf_{t,d} * log_{10} \frac{N}{df_t}$$

For a document d, it can be represented using a vector as follows:

$$\overrightarrow{w_d} = (w_{1,d}, w_{2,d}, \dots, w_{V,d})$$

In the above formula, V is the number of unique terms in the corpus. Each document in a corpus as well as query documents can be represented by such a feature vector, thus the cosine similarity scores can be computed to evaluate the similarities and decide the final ranking of retrieval results. The cosine similarity between a query document and a document in the corpus can be calculated as follows:

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

In the corpus of textual ontology information, for each genus, we have several property-description pairs. They can be assigned to four categories: shell-based, ventral-based, dorsal based and other categories. We consider each category as a smaller corpus, then for each genus, it can

be represented using four different TFIDF feature vectors. We call this category-based TFIDF features PTFIDF, which means parallel TFIDF. The similarity scores calculated based on PTFIDF are used to be fused with iRST-SHELO features in the next stage.

3.4.2 Integration of Textual Ontology and Visual Features

Based on PTFIDF and iRST-SHELO features generated previously, we re-rank the results returned by textual ontology based search by combining them together. The proposed approach to decide the similarities between the query input and a genus in the corpus is as follows:

- First, calculate the Centroid of four PTFIDF features for both the query input and the genus, we get two feature vectors **q** and **g**.
- Second, compute the cosine similarity S between vectors **q** and **g**.
- Third, calculate the group distance D between the query image set and the image set of the genus.
- Fourth, the final similarity score is decided using:

$$S' = S + \alpha \frac{1}{1 + \ln(D + 1)}$$

Since D is always no less than zero, we can constraint the second part of right side of this formula to be always no large than α. α is the parameter for balancing the weight between PTFIDF and iRST-SHELO features.

Chapter 4. Experimental Results and Analysis

In chapter 3 we talked about the approach used in content-based search function. Details of textual ontology based search, image based search and text-image based search were discussed separately. In this chapter, we will show some experimental results we generated among designing and implementing these functions, and prove that the content-based search did integrate information from both textual ontologies and images, thus overcome limitations in the old IPKB system.

4.1 Textual Ontology based Search

In this section, we will do two experiment. In the first experiment, we use the textual ontology similarity scores to compare the differences between genera. Through this experiment, we are able to demonstrate the availability and rationality of this scoring mechanism. Then in the second experiment, we gave out some results of queries using purely textual ontology based search without any help from image information.

4.1.1 Experimental Settings

Our current textual ontology framework is constructed upon genera data from Part H of the Treatise. There are totally 3829 genera which containing more than 29000 sentences of morphological descriptions and these sentences are used to build the textual ontology's region name-description pairs. We write a Java program to convert unstructured morphological descriptions for each genus to textual ontology based region name-description pairs, and store them in JSON format for the future usage.

The web interface for users to input textual ontology based queries is as in Figure 4.1. We can see that this page allows users to input at least one and up to ten region name-description pairs as queries.

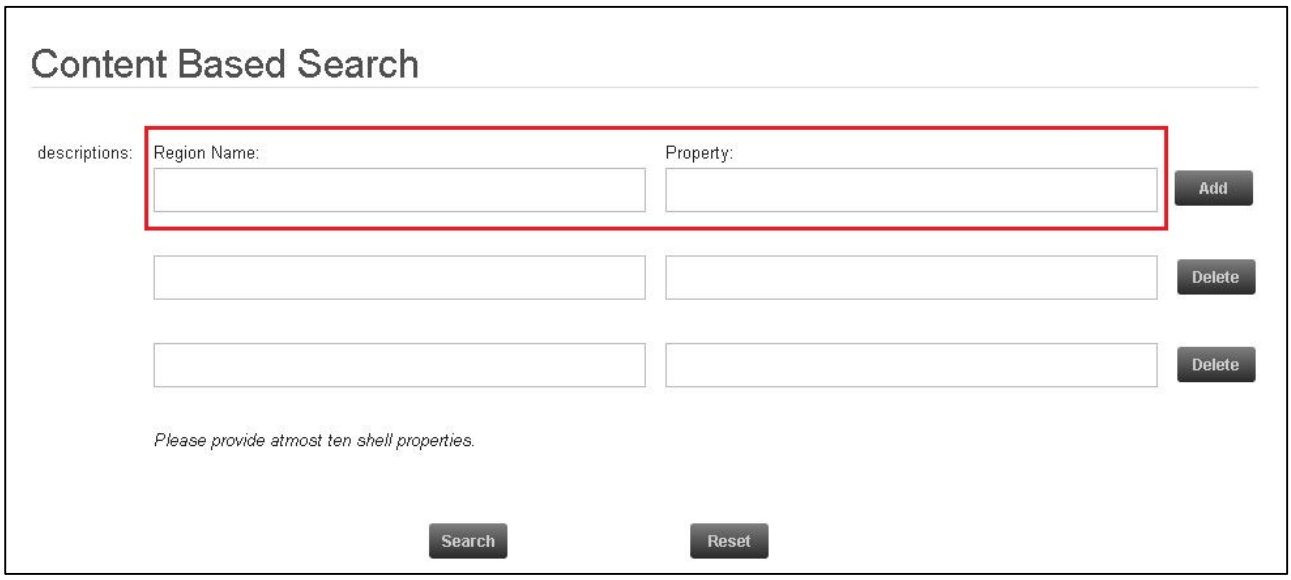


Figure 4.1: Web interface for textual ontology based search

4.1.2 Scoring Mechanism Demonstration

In order to demonstrate the availability and correctness of textual ontology based scoring mechanism, in this experiment, we choose to compare the differences between genera in our corpus rather than differences between queries and genera. We choose four genera: Lingula, Barroisella, Semilingula and Credolingula from Part H of the Treatise as our samples. The fossil images for each genus are shown in Figure 4.2.

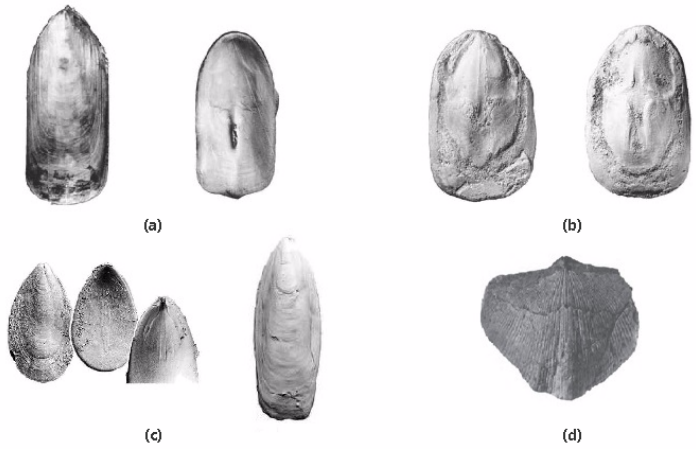


Figure 4.2: Fossil images for Lingula (a), Barroisella (b), Semilingula (c), and Credolingula (d)

Observing the appearances of these fossils, we can easily identify that genus Lingula and Semilingula are most similar to each other. Semilingula also looks similar to Lingula, but not that much as Semilingula. Credolingula looks quite different from each other. Now we calculate similarities between Lingula and other three genera, then the results are shown in Figure 4.3:

Genus A	Lingula	Lingula	Lingula
Genus B	Semilingula	Barroisella	Credolingula
Similarity	1	0.25	0

Figure 4.3: Similarities between genera

Similarity scores in Figure 4.3 demonstrate our observation on Figure 4.2, which shows the availability and correctness of the scoring mechanism we adopt in textual ontology based search.

4.1.3 Textual Ontology based Search Results

After demonstrating the scoring mechanism, we compare the results of textual ontology based search with the advanced search function to show the benefit of the new approach. However, because of the limitation of space, we only show results for one query input. In this experiment, we use “small and rounded to subtriangular shell” as the query for advanced search function, and “shell”- “small and rounded to subtriangular” as the query for textual ontology based search function. We do not compare textual ontology based search and quick search because quick search does not allow users to input query terms other than taxon names.

Given the query input above, the advance search returns 100 results in total and 10 results on each web page. The top 2 genera results on the first page can be seen in Figure 4.4. By observing these two genera, we can find terms like “shell”, “subtriangular” and “rounded” occurs multiple times in their morphological descriptions. However, the results are not expected by the user. In the description of Longipegma, the “subtriangular” is actually used to describe structure detail of the dorsal valve region rather than the shell.

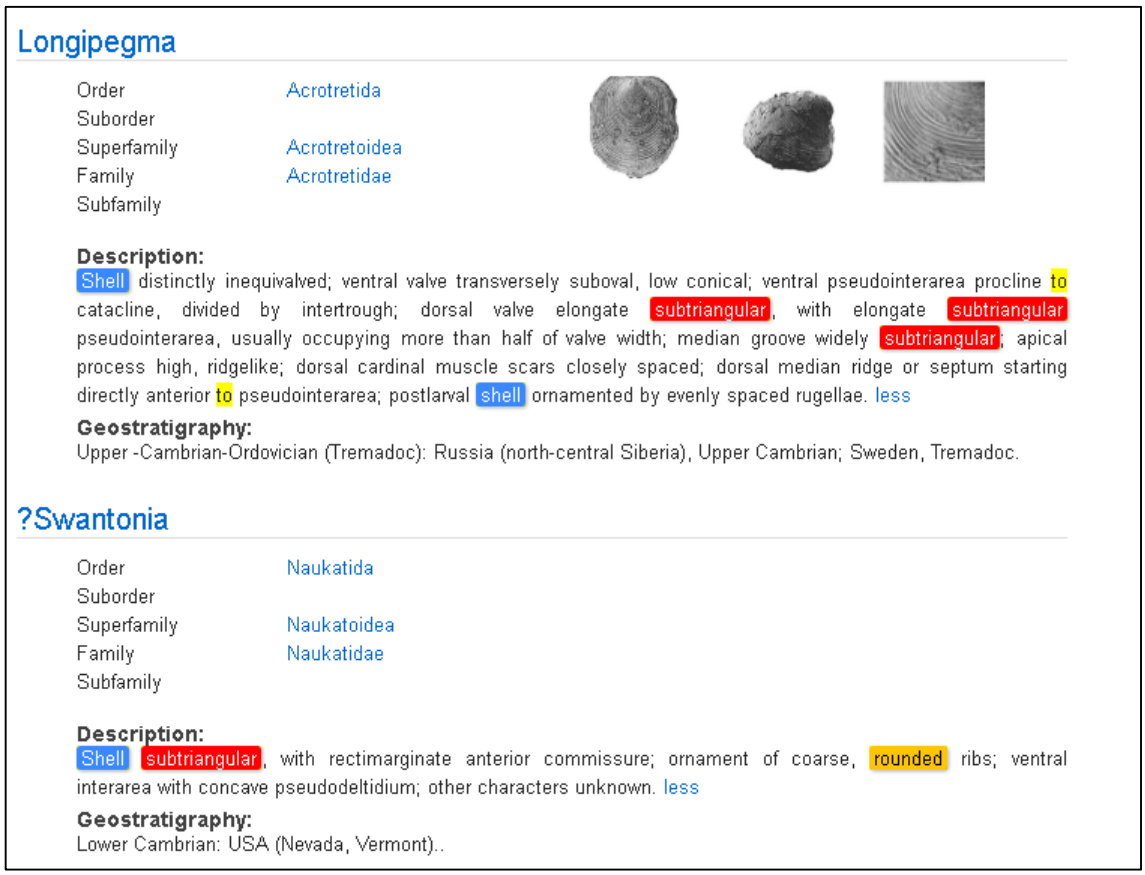


Figure 4.4: Top 2 results of advanced search using “small and rounded to subtriangular shell”

Using the same query information in textual ontology based search, 59 results are returned in total. Figure 4.5 shows the top 3 results of this searching. We can see that in the first three searching results, “small”, “rounded” and “subtriangular” are all used to describe the “shell” rather than regions in “dorsal” or “ventral” view. Morphological descriptions are correctly linked to their corresponding region names, as indicated in Figure 4.5. Obviously, the textual ontology search shows more advantages on meeting users’ intension than in such kinds of situation than the original textual based advanced search function.

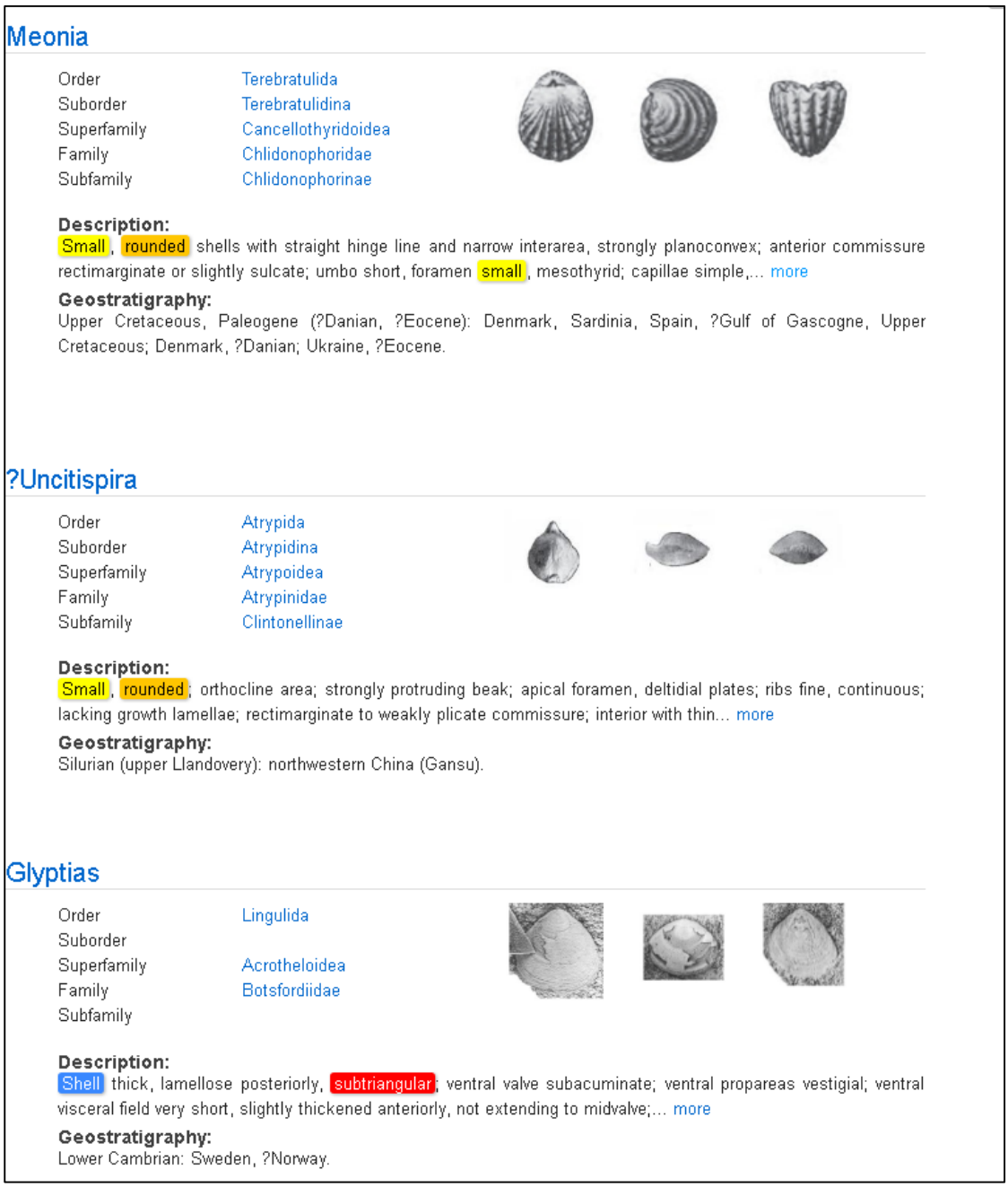
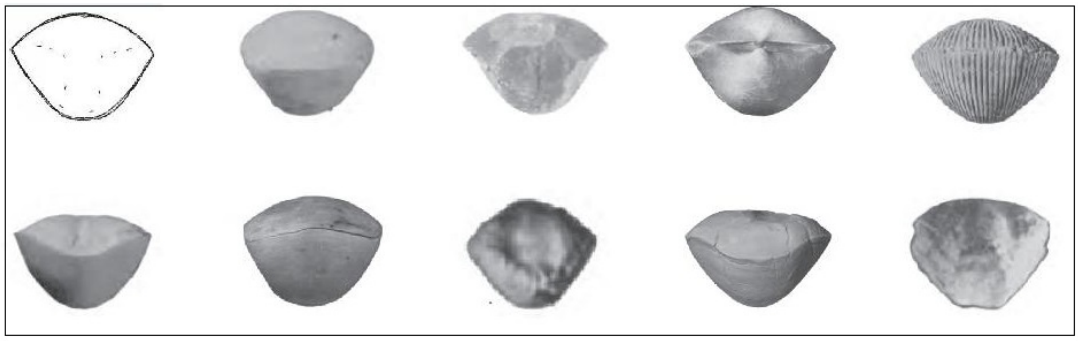


Figure 4.5: Top 3 results of ontology search using “shell”-“small and rounded to subtriangular”

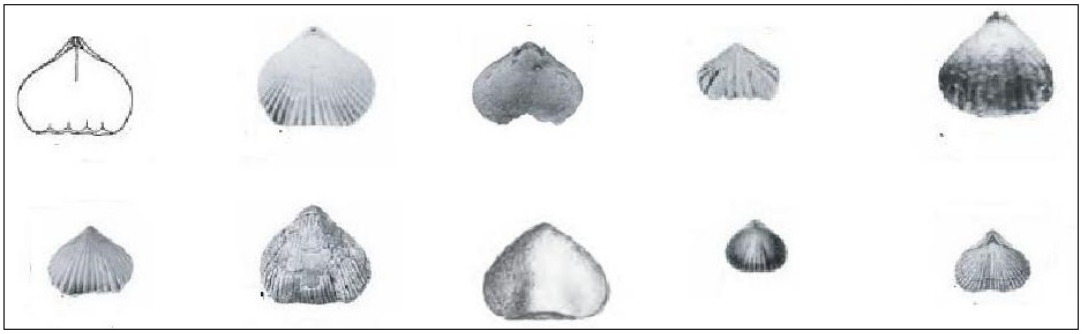
4.2 Image based Search

As we mentioned previously, users of the IPKB system usually cannot describe a fossil with words correctly and completely. At this time, a photo of the fossil, or even a simple hand drawn

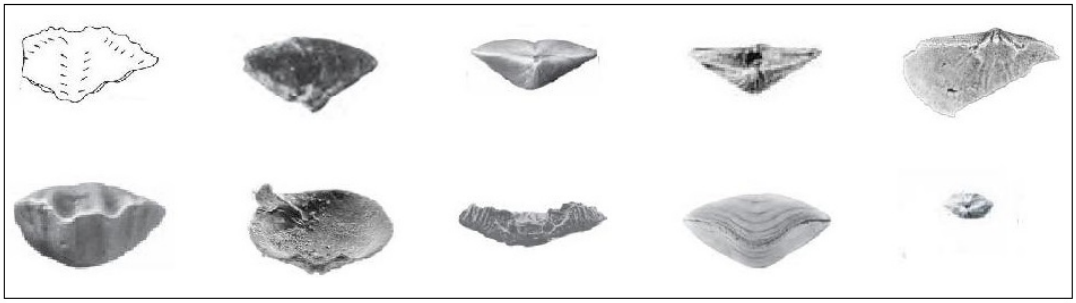
sketches can help to improve the search results. In this section we will firstly show several image retrieval results using both photos and hand drawn sketches. The features used for comparison are iRST-SHELO features with length 206. The distances between features are evalulated using Euclidean distances.



(a)



(b)

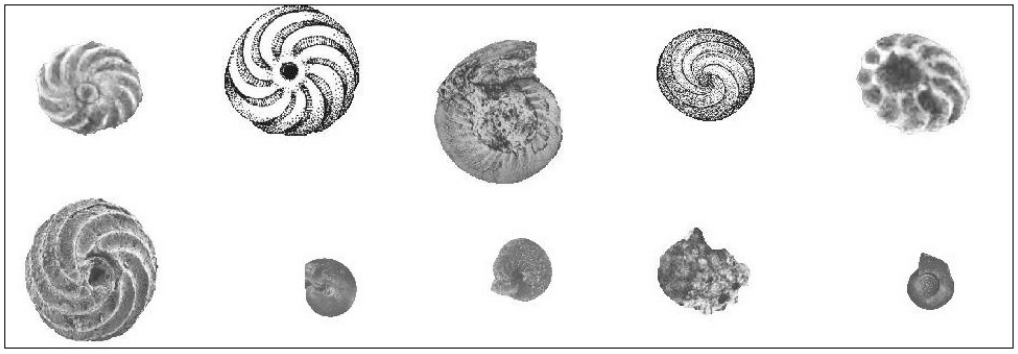


(c)

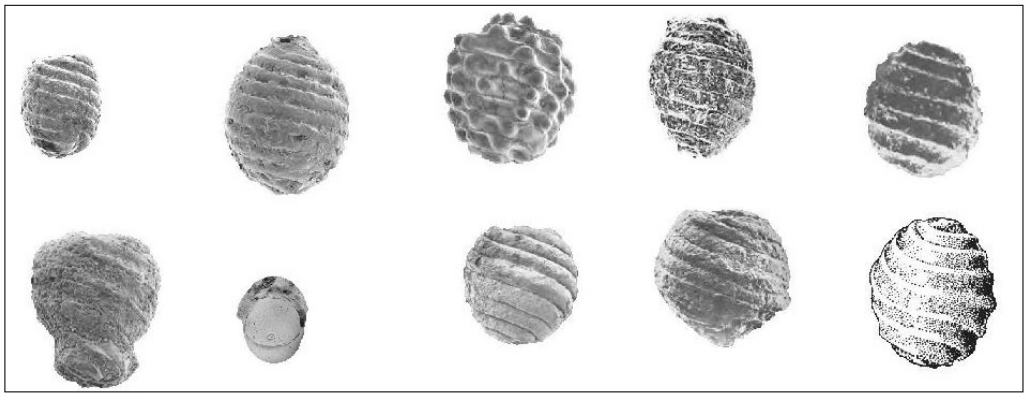
Figure 4.6: Image retrieval results using hand drawn sketches as queries

Figure 4.6 shows three groups image retrieval results using iRST-SHELO features. In each

group, the query image is located at the up left corner. The other 9 images in each group are ranked based their similarities to the query sketch. When query images are photos, iRST-SHELO features and Euclidean distances still work very well in our cases (see Figure 4.7). The photo at the up left corner of each group.



(a)



(b)

(c)

Figure 4.7: Image retrieval results using photos as queries

The results shown in Figure 4.6 and 4.7 demonstrate that iRST-SHELO features provide an effective and efficient way to represent images and also rank them. In this experiment we only use one query image each time, however, we look for similar items by going through all fossil images (18822 images in total) in our image data set. In real content-based search page, we actually do not compare the query image with all images since we use textual ontology based search first to narrow down the search scope for next step.

4.3 Text-Image based Search

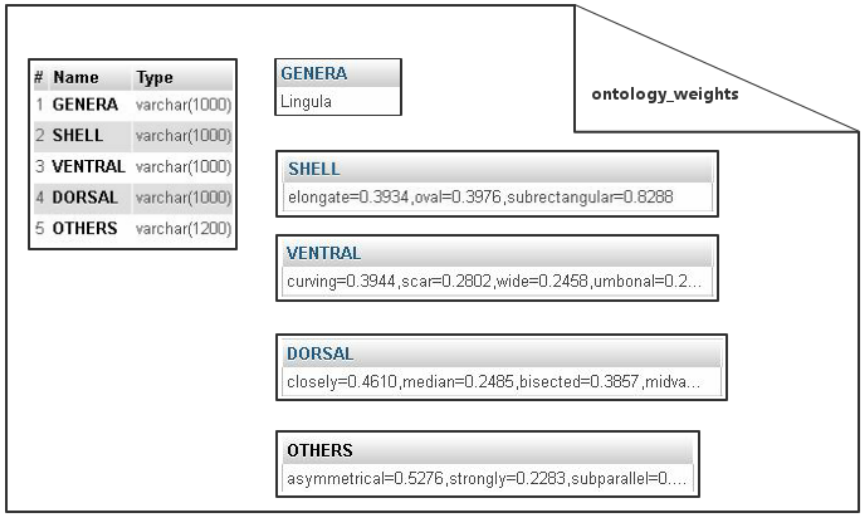
Before text-image based search, as we discussed above, we first use textual ontology based approach to reduce the number of returned results. Upon these results, we use approaches presented in chapter 3 to execute text-image based search.

4.3.1 Experimental Settings

In order to facilitate text-image based search combining textual ontology and image information together, several tables for PTFIDF features and an iRST-SHELO table are built offline in our database. PTFIDF features are stored in four tables:

- Table **ontology_weights**: stores term frequency weights in category for each genera, there are totally 3829 entries in this table, which is exactly the number of genera used for textual ontology construction.
- Table **shell_df**: stores document frequencies of terms in morphological descriptions which directly describing the shell. There are 993 entries in this table, which means there are 993 unique terms used for such a kind of description (stop words eliminated).
- Table **dorsal_df**, **ventral_df** and **other_df**: stores document frequencies of terms in morphological descriptions on dorsal regions or sub-regions (1339 entries), ventral regions or sub-regions (1567 entries) and other regions (2480 entries) respectively.

Figure 4.8 shows structures of these tables. Figure 4.8(a) shows details of **ontology_weights** with some examples. Noticing the term-value pair stored the SHELL, VENTRAL, DORSAL and OTHERS attributes. The value is calculated by dividing the TFIDF weighting by the length of TFIDF feature vector. Figure 4.8(b) shows the structure for other three tables.



(a)

TERM	DF
elliptical	7
divides	1
anteriorly	284
marginal	17
hard	1
ridge	74
chilidium	8
perforating	1
quarters	1
articulatory	1
better	1

(b)

Figure 4.8: Structure of tables for PTFIDF features (a) ontology_weights. (b) Others.

Except for the above four tables which stores PTFIDF features, there is another important table called **irstshelo_features** which stores iRST-SHELO features of all images in the preprocessed image data set. There are 18822 entries in this table and its structure can be seen in Figure 4.9. The length for an iRST-SHELO feature is 206 thus there are 206 numbers saved in the attribute FEATURES in the table.

IMAGE	GENERA	FEATURES
volb_F44_1a.jpg	Moellerina	2.862293 3.021271 -0.882540 3.638897 1.666107 3.00...
volb_F44_1b.jpg	Moellerina	4.866003 3.356774 -0.901122 5.634988 0.201278 -0.6...
volb_F44_1c.jpg	Moellerina	1.782579 3.762333 -0.091893 5.024413 0.938144 1.73...
volb_F44_1d.jpg	Moellerina	4.042194 2.261297 -1.423136 6.596042 0.088624 -0.7...
volb_F44_1e.jpg	Moellerina	3.322828 2.319325 -1.368620 5.828199 1.582553 -2.1...
volb_F44_2a.jpg	Gemmichara	2.556573 2.921962 -0.094953 5.801504 0.582255 2.16...
volb_F44_2b.jpg	Gemmichara	4.319142 0.236954 -0.561031 6.042650 -0.044059 -0....
volb_F44_2c.jpg	Gemmichara	3.858283 3.105913 -0.629876 5.520691 0.008147 -1.4...
volb_F44_3.jpg	Primochara	2.014035 3.818589 -2.519082 4.343688 -1.010868 1.0...
volb_F44_4a.jpg	Pseudomoellerina	2.067958 6.004908 -1.068007 3.334443 0.071590 0.86...
volb_F44_4b.jpg	Pseudomoellerina	3.352609 4.131793 -0.653299 6.654825 0.199507 -1.3...

Figure 4.9: Structure of table irstshelo_features.

The value of α in the integration formula is set to 1.2 in our experiment. It can be set to other values to balance the weight between textual ontology information and image information.

4.3.2 Results and Analysis

In this experiment, we allow users to input textual ontology descriptions as well as uploading an image as the queries. We still type in “shell” and “small and rounded to subtriangular” as a textual ontology pair, in addition, we upload a photo of a fossil as in Figure 4.10.

Figure 4.10: The query Image

In order to show the benefit brought by importing image information, we compare the results

of textual ontology based search and text-image based search using the same queries as stated above. Figure 4.11 shows the results of text-image based search. Textual ontology information is still reflected by the top three results. Moreover, in the first and third genus on the list, we found images of fossils which are pretty similar to the query image.

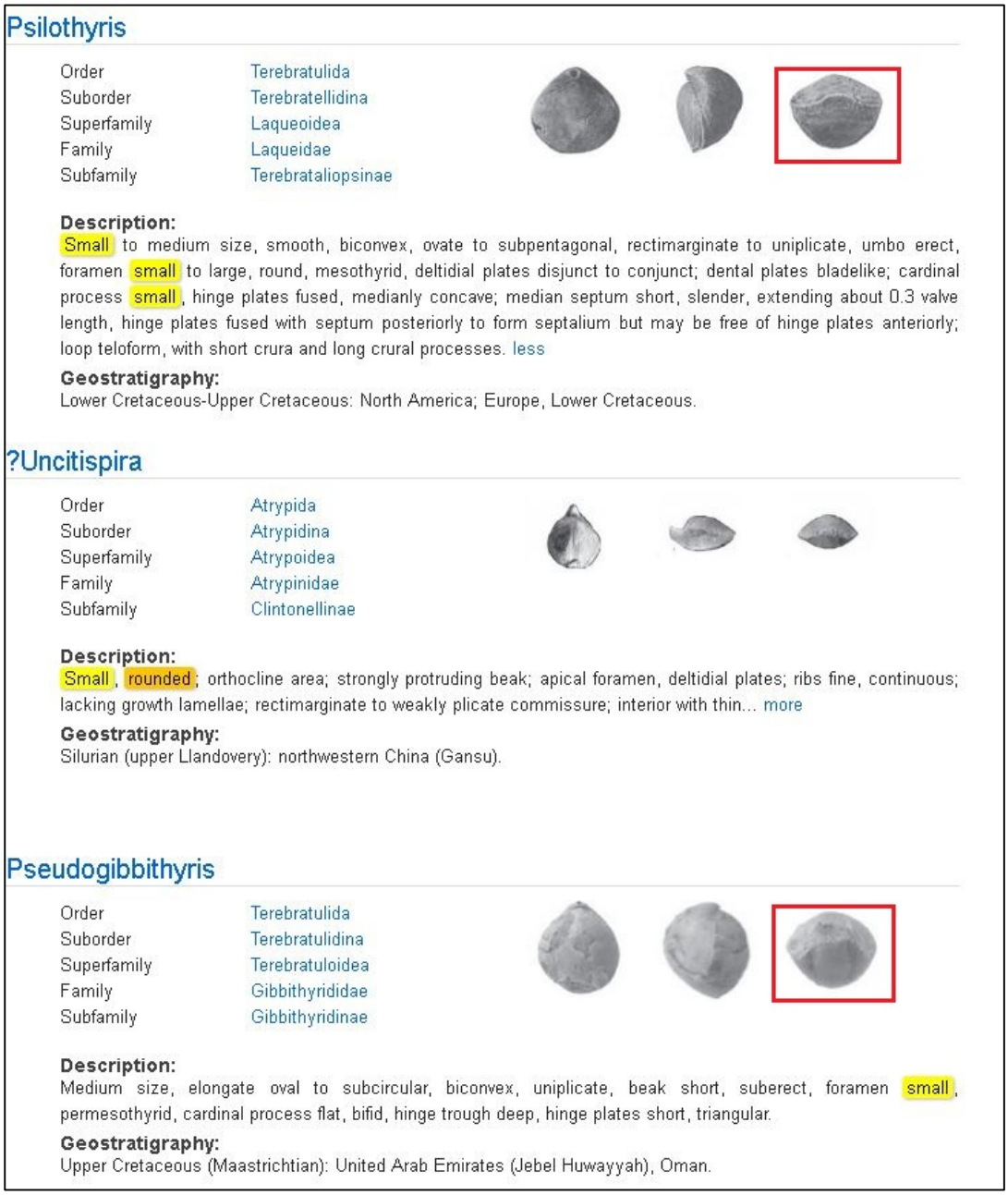


Figure 4.11: Top 3 results of text-image based search

Notice the first genus Psilothyris in the list in Figure 4.11. Where will it be when we execute pure textual ontology based search? The answer is in Figure 4.12. Psilothyris is ranked at the last position of the first page, which is far away from the first genus in Figure 4.5.

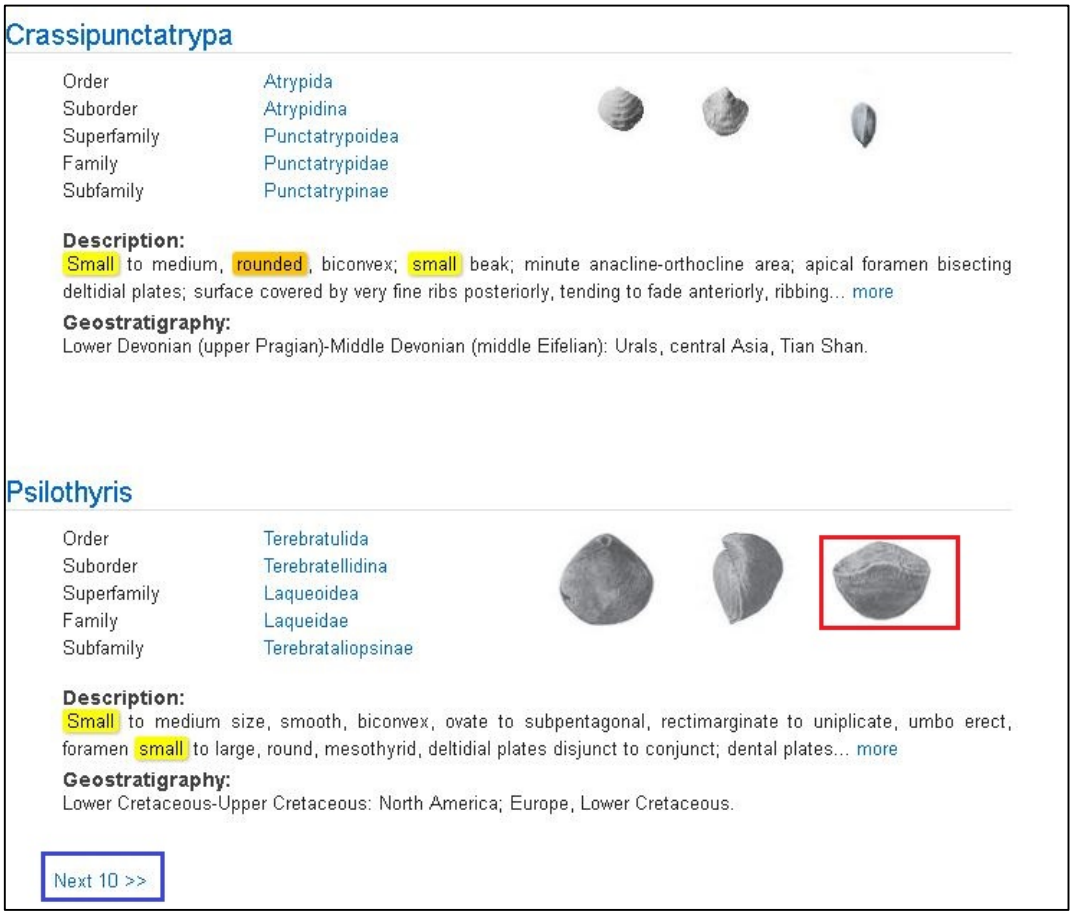


Figure 4.12: The last two results on the first page in textual ontology based search

4.3.3 Summary

The quick search and advanced search are designed only to compare term similarities between certain attributes such as taxon name, morphological description, time period, etc. In many cases in cannot meet users' requirement. The first experiment which comparing the results of advance search and textual ontology based search demonstrate this statement. From the experimental results of pure textual ontology based search, we can conclude that textual ontology

information can help us locate genera with specific morphological properties on certain regions. However, when the user cannot give out correct and rich enough textual ontology information, the searching may be confusing.

In the final experiment, we use both textual ontology information extracted from genera descriptions and visual features generated by fossil images. Both of these two features are reflected dramatically in the search results. On top of that, with the help of images, users don't need to input lengthy and complicated technical terms to describe the appearance of a fossil. Therefore, the user experience is largely increased by this content based search.

Chapter 5. Conclusion and Future Work

Invertebrate Paleontology Knowledgebase (IPKB) is a digital library constructed for the Treatise of Invertebrate Paleontology. It does not only provide a way for users to browse genera records and upper layer taxa records, but also allow users to input keywords to search for certain genera. In the old IPKB system, there are only genera records extracted from Part H of the Treatise. In order to enlarge the range of genera for searching and browsing, we extract and parse text and image data from Part B, Part L and Part O thus enlarge the size of IPKB's database. Text and images from different parts of the Treatise may have different attributes, although these differences are not large enough to be paid much attention, we still make some trivial modifications on previous approach to parse data.

Two different types of search functions: quick search and advanced search are provided by the old IPKB system. Results for these two search functions are picked out and ranked based on the term occurrences in attributes such as taxon name, morphological description, etc. In many situations, this cannot satisfy user requirements. When users want to describe a certain region of the fossil, quick search cannot give out reasonable results. In addition, the expected users of IPKB system can be non-professional in the paleontology field. He or she may not be able to use correct and enough technique terms to describe a fossil, which makes search functions not available for these users.

In order to overcome these challenges during searching and improve user experience to a certain extent, we propose a novel search function called content based search. Content based search contains three major components: textual ontology based search, which reorganizes morphological descriptions into structured textual ontologies; image based search, which uses iRST-SHELO features to represent each fossil images and a group comparison approach to compare similarities between groups of images; text-image based search, which uses a weighting technique to integrating textual ontology and visual features to form a complete content based

search module. Through experiments, we demonstrate the availability and advantages of the textual ontology based search over the previous pure text based search, and also the largely increased user experience by integrating visual information to the content based search.

After adding this novel search function, IPKB system becomes much more powerful and intelligent than previous version, however, there are still a lot of work to be done to complete this digital library system. First, the database is not complete thus we only import fossil data from Part B, Part H, Part L and Part O. Second, the visual features can be integrated with search functions not limited to only textual ontology based search. Third, there still a large portion of the web interface to be improved. Although IPKB system is not perfect and complete at this time, the techniques adopted in its construction may be used in other research area.

References

[1] http://paleo.ku.edu/

[2] https://en.wikipedia.org/wiki/Invertebrate_paleontology

[3] Patric Denton, Yuxin Chen, Bo Luo, Paul Selden and Xue-wen Chen (2012). IPKB: A Digital Library for Invertebrate Paleontology. JCDL ’12 Proceedings of the 12th ACM/IEEE-CS joint conference on digital libraries, (pp. 101-110)

[4] Bell, Suzanne (2015). Librarian's Guide to Online Searching: Cultivating Database Skills for Research and Instruction, 4th Edition: Cultivating Database Skills for Research and Instruction. p. 69. ISBN 1610699998.

[5] Pomerantz, Jeffrey, & Marchionini, Gary. "The Digital Library as Place," Journal of Documentation, 63(4), 2007, pp. 505-533.

[6] http://dl.acm.org/

[7] S. Shneiderman, M. Turin, and the Digital Himalaya Project Team. Digital himalaya: an ethnographic archive in the digital age. European Bulletin of Himalayan Research, 20(1):136--141, 2002.

[8] Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp.63-70.

[9] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

[10] [Yonghao He, Jian Wang, Shiming Xiang, Chunhong Pan] Image Annotation via Learning the Image-Label Interrelations. *IEEE* International Conference on Image Processing (ICIP), 27-30 Oct. 2014 (pp. 3102 - 3106)

[11] [Minmin Chen, Alice Zheng, Kilian Q. Weinberger] Fast Image Tagging. Proceedings of the 30th International Conference on Machine Learning (ICML).

[12] [Matthieu Guillaumin, Thomas Mensink; Jakob Verbeek; Cordelia Schmid] TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. 2009 IEEE 12th International Conference on Computer Vision. (pp. 309 - 316).

[13] [Aditya Khosla, Jianxiong Xiao, Antonio Torralba, Aude Oliva] Memorability of Image Regions. NIPS, 2012.

[14] Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. pp. 1150–1157.

[15] [Aditya Khosla, Jianxiong Xiao, Antonio Torralba, Aude Oliva] Memorability of Image Regions. NIPS, 2012.

[16] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2005.

[17] Saavedra J M. RST-SHELO: sketch-based image retrieval using sketch tokens and square root normalization [J]. Multimedia Tools and Applications, 2015: 1-21.

[18] Jose M. Saavedra. Sketch-based image retrieval using a soft computation of the histogram of edge local orientations (s-helo). In International Conference on Image Processing, ICIP'2014(To appear), 2014.

[19] Yuxin Chen; Sampathkumar, H.; Bo Luo; Xue-wen Chen, "iLike: Bridging the Semantic Gap in Vertical Image Search by Integrating Text and Visual Features," Knowledge and Data Engineering, IEEE Transactions on , vol.25, no.10, pp.2257,2270, Oct. 2013.

[20] A. Sharma, A. Kumar, H. Daume and D.W. Jacobs, "Generalized Multiview Analysis: A Discriminative Latent Space", Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012.

[21] J. Song , Y. Yang , Z. Huang , H. T. Shen , R. Hong, Multiple feature hashing for real-time large scale near-duplicate video retrieval, Proceedings of the 19th ACM international conference on Multimedia, November 28-December 01, 2011, Scottsdale, Arizona, USA.

[22] X. Shen , F. Shen , Q. Sun , Y. Yuan, Multi-view Latent Hashing for Efficient Multimedia Search, Proceedings of the 23rd ACM international conference on Multimedia, October 26-30, 2015, Brisbane, Australia.

[23] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Transactions on Image Processing, 2(2):176--201, April 1993.

[24] Fernand Meyer, Topographic distance and watershed lines, Signal Processing, v.38 n.1, p.113-125, July 1994.

[25] [Kristina Toutanova and Christopher D. Manning. 2000] Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.

[26] Jiangsheng Yu, Yuanliang Meng, Junyan Li, Bo Luo, and Xue-wen Chen, "Automatic Extraction of Non-independent Terms using Hypothesis Testing and Likelihood Interval Methods," Information and Telecommunication Technology Center, the University of Kansas, Lawrence.

[27] Abdi. H., & Williams, L.J. (2010), "Principal component analysis". Wiley Interdisciplinary Reviews: Computational Statistics 2 (4): 433–459.

[28] Arandjelovic R, Zisserman A, "Three things everyone should know to improve object retrieval". CVPR, pp 2911–2918, 2012.

[29] B. On, N. Koudas, D. Lee, and D. Srivastava, "Group linkage," ICDE, pp. 496-505, 2007.