A System for Understanding Imaged Infographics and Its Applications

Weihua Huang
School of Computing, National University of Singapore
3 Science Drive 2
Singapore 117543
+65-65162784
huangwh@comp.nus.edu.sg

Chew Lim Tan
School of Computing, National University of Singapore
3 Science Drive 2
Singapore 117543
+65-65162900
tancl@comp.nus.edu.sg

ABSTRACT
Information graphics, or infographics, are visual representations of information, data or knowledge. Understanding of infographics in documents is a relatively new research problem, which becomes more challenging when infographics appear as raster images. This paper describes technical details and practical applications of the system we built for recognizing and understanding imaged infographics located in document pages. To recognize infographics in raster form, both graphical symbol extraction and text recognition need to be performed. The two kinds of information are then auto-associated to capture and store the semantic information carried by the infographics. Two practical applications of the system are introduced in this paper, including supplement to traditional optical character recognition (OCR) system and providing enriched information for question answering (QA). To test the performance of our system, we conducted experiments using a collection of downloaded and scanned infographic images. Another set of scanned document pages from the University of Washington document image database were used to demonstrate how the system output can be used by other applications. The results obtained confirm the practical value of the system.

Categories and Subject Descriptors
I.7.5 [Document and Text Processing]: Document analysis, Graphics recognition and interpretation.

General Terms
Design, Experimentation.

Keywords
Infographics, association of text and graphics, document image understanding, applications.

DocEng'07, August 28–31, 2007, Winnipeg, Manitoba, Canada.

1. INTRODUCTION
Automatic recognition and understanding of document content has been a popular research area because it allows efficient information retrieval and extraction from documents. Traditionally, information retrieval and extraction from documents focused on documents with single information modality. For text-based documents, techniques such as summarization, categorization and question answering (QA) etc. have been studied widely. When dealing with graphical documents, different techniques have been explored, such as graphical symbol segmentation, graphical model fitting and graphics parsing etc. Another type of documents that is widely studied is text-based documents which are converted to the raster image format. For this type of document, layout analysis, image segmentation and optical character recognition (OCR) are common techniques involved. In real life, a common situation is that a document is comprised of a mixture of information from multiple modalities such as text and image. In this case, we are facing the problem of integrating information extracted from each modality for capturing more complete content of the document. A typical example is the infographics inserted in documents such as scientific papers or new articles. Information graphics, or infographics, is an effective visual representation that explains information simply and quickly using combination of text and graphical symbols. William S. Cleveland studied the elements in infographics, the principles and the graphing tools for designing infographics to represent data [5, 6]. One frequently used type of infographics is the scientific charts (bar chart, line chart, pie chart etc.) that are often used for displaying data values [21] or delivering intended messages through data illustrations [3]. As Carberry et al [4] suggested, understanding infographics itself can be treated as a discourse problem in the sense that it requires assimilating information from multiple knowledge sources such as text and graphics. Furthermore, most infographics appearing in documents are reproduced as raster images where everything becomes pixels. Thus recognition and understanding of imaged infographics is a more challenging problem than the understanding of electronic infographics that were directly generated through graphical packages such as Microsoft Excel etc. Figure 1 shows an example of scanned document page containing an infographic.

We are developing a system that recognizes and interprets imaged infographics. The input of the system is a document containing imaged infographics. The system generates description of the

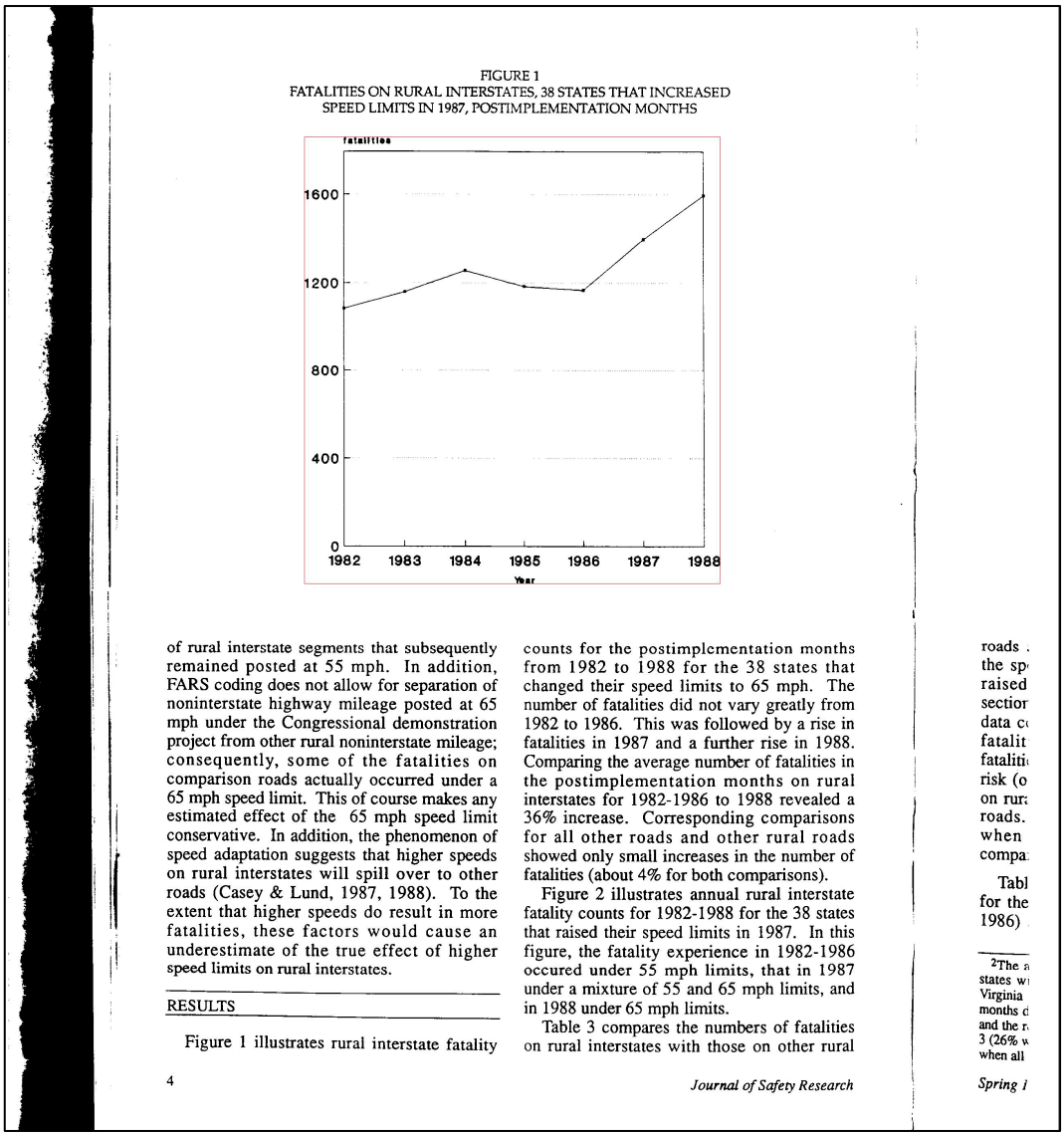

Figure 1. A sample scanned document containing an infographic. The boxed area is the infographic identified by the system.

infographics in both XML format and natural language form, which can be used for other applications. Such applications include providing a supplement for optical character recognition (OCR) systems as existing OCR systems are not able to recognize infographics in document pages. Also, the information obtained from the infographics, together with other textual information of the document, can be used for information extraction applications such as question answering etc. There are many kinds of infographics in the real life. At the moment, we take charts as a representative of infographic, mainly because they are commonly used and the way of data representation in charts gives us chances to explore data interpretation better than dealing with other kinds of infographics such as maps etc. The scheme proposed here, however, can be extended to other kinds of infographics, because most techniques involved are general methods.

The remaining sections of the paper will be as follows. Section 2 reviews related works in infographics recognition and understanding. Section 3 introduces the details of the major modules of our system. Section 4 presents two sample applications, namely OCR supplement and question answering. Section 5 shows experiments conducted and the results. Finally, section 6 concludes the paper and suggests future works.

2. RELATED WORKS

Recognition and understanding of infographics is a relatively new research field. In recent years, more and more works in this field have been reported. As an early attempt, Futrelle et al presented a diagram understanding system based on graphics constraint grammars to recognize x-y data graphs and gene diagrams [9]. In the computational linguistic community, Carberry et al proposed a scheme for summarizing the intended message in electronic

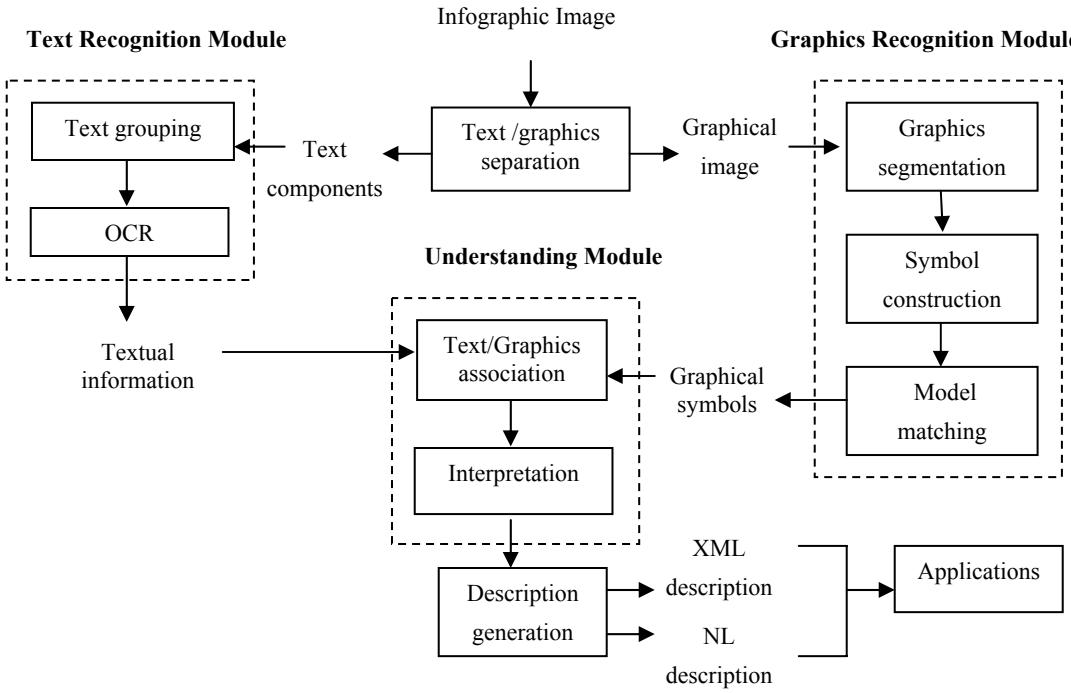

Figure 2. Overview of the system architecture.

infographics [3] and they modeled the infographics understanding problem as a discourse problem [4]. In document image analysis community, researchers are more interested in dealing with imaged infographics. Yokokura et al proposed a schema-based framework to graphically describe the layout relationship information of the bar chart images [20] based on vertical and horizontal projections. Zhou et al applied Hough-based techniques to achieve bar chart detection and segmentation [24]. Later they also proposed a learning-based chart recognition paradigm using Hidden Markov Models [25]. We proposed a model-based approach for recognizing several commonly used types of chart image [11] and a method for classifying chart images based on shape features [12].

Most of these previous works studied the commonly used infographics which are charts (bar charts, pie charts or line charts). These works focused on recognizing and understanding individual infographic images. None of them attempted to integrate the information extracted from infographics with other textual information in the document that contains the infographics. We believe that such integration makes better use of the information extracted from infographics for the purpose of information extraction etc. Thus in the system we developed, the input to the system is the whole document page containing infographic images instead of a single infographic image. There are also some problems not clearly addressed by the previous works. For example, to achieve understanding of infographics, both textual and graphical information contained in an infographic need to be extracted and associated together. In our system, association of text and graphics is an important step and the details will be presented in the following section.

3. DETAILS OF THE SYSTEM

Figure 2 shows the major modules in our system. The main idea is to separate text and graphics and pass them to the corresponding modules to be recognized. The extracted information from the two modalities is then associated and interpreted to form a complete description of the infographic. This description is stored both in XML format and natural language form, to be used by other applications that will be discussed in section 4.

One extra step is not shown in the diagram in Figure 2 because it is optional. The step is to find out the location of the infographics from an input document. If a document is a mixture of text and imaged figures (including infographics), then text and images are directly separated by checking the layout information available in the document structure. An example of such documents is the web-pages written in HTML. In this case, the system focuses on extracting information from the images and combines the extracted information with the textual part of the document. On the other hand, if the whole document itself is a raster image, such as a scanned document page, then this extra step in the system is performed to locate the figures and identifies infographics from them. This task is done through logical layout analysis [1] followed by image classification [18]. After the infographics are identified, all the steps in the remaining modules are carried out to perform recognition and interpretation to capture their content. Details of the major modules in the system are presented in the following sub-sections.

3.1 Preprocessings and Text/Graphics Separation

Several preprocessing steps are performed before graphical and textual information can be extracted from an imaged infographic:

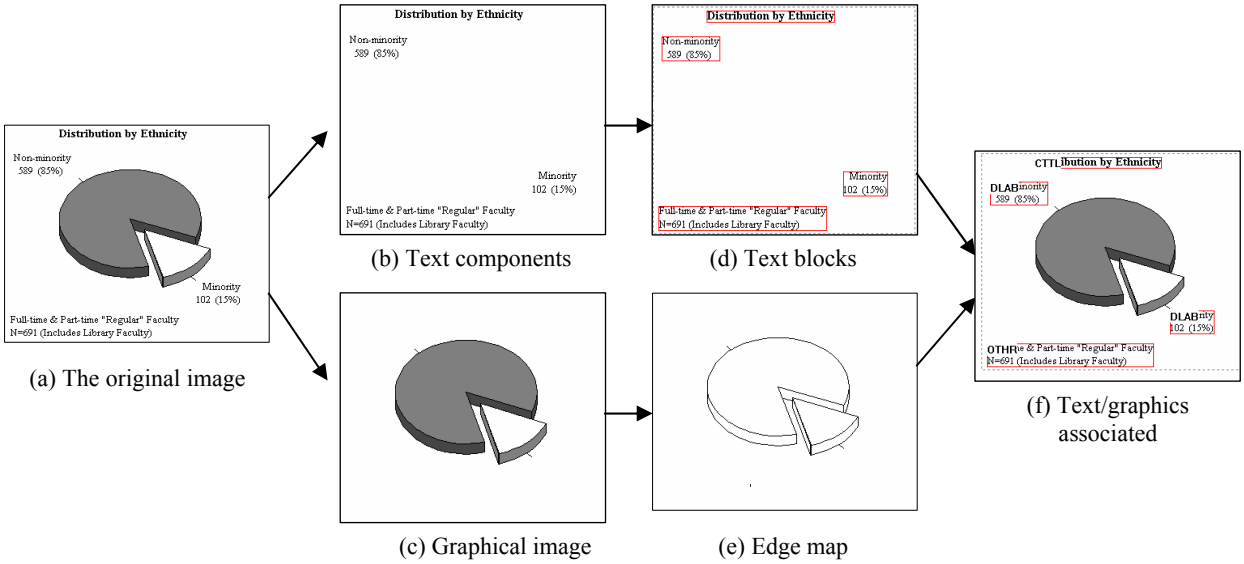

Figure 3. Example of imaged infographic understanding (before description generation).

- If an infographic is an image with RGB colors, it is converted to a grayscale image. The purpose of doing so is to increase the efficiency of the following steps. The original image is not overwritten by the new grayscale image, so that the color information is still accessible in later stages.
- From the grayscale image, connected component analysis [17] is performed within the imaged infographic area to separate text from graphics. A series of filters are applied to classify the connected components into textual components and graphical components. These filters take into consideration the size, the height and width, the height/width ratio and the black pixel density of each connected component constructed. In this way, the connected components are classified into three types: graphical component, textual component and noise. Textual components and graphical components are stored separately, to be processed by corresponding recognition units.
- Edge map is obtained for the graphical components. This is done through edge detection. Commonly used edge detector such as the Canny Edge Detector is applied here. After this step, the image containing graphical components becomes a line drawing. This step is necessary for the vectorization step which is carried out before graphical symbol construction.

Figure 3 (a) shows a sample infographic. Figure 3(b) shows the text components classified. Figure 3(c) shows the image with all text components removed. Figure 3(e) is the edge map obtained.

3.2 Graphical Symbol Extraction

A graphical symbol is defined as the smallest graphical unit that carries meaningful information. In our system, construction of graphical symbol is a bottom-up process in which primitive graphical entities are put together to form more complex symbols. Primitive graphical entities include straight line segments and arcs formed through vectorization process. The set of graphical symbols to be formed depends on the type of infographics being studied, or in other words depends on the domain knowledge attached to each infographic type. For infographics that are charts or data plots, the most important graphical symbols to obtain are the coordinate lines and the symbols that represents data.

Before graphical symbols are constructed, vectorization is performed to convert the line drawing obtained during preprocessing to a set of lines and arcs in the vector form. An algorithm with a data structure called Directional Single-Connected Chain (DSCC) [23] is applied to break the edges into small chains that are straight segments. A curve fitting algorithm based on [8] is then applied to the endpoints of these chains to further construct straight lines, circular arcs and elliptic arcs. The vector information of these primitive graphical entities includes the endpoints of each line, the thickness of each line, the center point for all arcs, the radius for circular arcs and the maximum and minimum radii for elliptic arcs.

The first graphical symbols to be located are the coordinate lines. Most charts, or data plots, contain coordinate lines. Some previous works [20, 26] proposed methods to detect these coordinate lines. However, we found the previous methods not robust enough to handle skew angles, customized coordinate system, and false positives such as frames or border lines. Thus some extra heuristics are used in our system for coordinate line detection, including: the coordinate lines must be perpendicular (in 2D case); there should be small text blocks along the line; the area bounded by the candidate coordinate lines must contain most of the graphical objects.

To identify graphical symbols representing data for each chart type, domain knowledge is used here. There are two levels of domain knowledge. At the top level, it specifies what kind of graphical symbols are expected to appear in each chart type. At the bottom level, the set of primitive graphical primitives and the constraints among them further determines how each graphical

symbol is formed. Part of the domain knowledge of several commonly used chart types are shown below:

- Bar chart:

  BarChart = {*x-axis*, *y-axis*, *BarSet*}, where

  BarSet = {*Bar*}, where number of elements ≥ 2 and

  Bar = {l_1, l_2, l_3 | $l_1 \perp l_3$, $l_2 \perp l_3$, $l_3 \parallel$ *x-axis*, CE(l_1, l_3), CE(l_2, l_3), EL(l_1, *x-axis*), EL(l_2, *x-axis*)}

- Pie chart:

  PieChart = {*Pie*}, where number of elements ≥ 2 and

  Pie = {l_1, l_2, a_1 | CE(l_1, l_2), EL(l_1, a_1), EL(l_2, a_1)}

- Line chart

  LineChart = {*x-axis*, *y-axis*, *Polyline*}, where

  Polyline = {l_i | CE(l_i, l_{i+1}), for i = 1, …, n-1} where n ≥ 2

Here, l_i refers to a line and a_i refers to an arc, where i = 1, 2, 3 … n. Some binary constraints between two primitive graphical entities *a* and *b* are defined:

- $a \parallel b$: line a is parallel to line *b*.
- $a \perp b$: line a is perpendicular to *b*.
- CE(*a*, *b*): shape *a* and *b* share one common endpoint.
- EL(*a*, *b*): one end point of shape *a* lies on shape *b*.

There are also global constraints for some chart models. For example, in a bar chart model, all bars must have similar width. And in a pie chart, the summation of the angles from all wedges must be 2π. These global constraints are hard to express using simple symbols thus they are not shown above.

The actual symbol construction process is done through extraction of all possible shapes or polylines specified in the domain knowledge of all chart models available. Based on the vectorized lines and arcs, a graph G(V, E) is formed where V is the set of intersection points among the lines and arcs, and E is the set of segments (either straight line segment or arc segment) between intersection points. Considering all primitive graphical entities and their intersections, construction of symbols is done using constrained graph search on G.

After all possible symbols are constructed, the system then checks them against every chart model available in the domain knowledge to calculate the likelihood that the given infographic image being one of the chart types. The chart type returning the highest likelihood is deemed to be the type of the input infographic image. Once the chart type is determined, all relevant graphical components are passed to the understanding module to be associated with the textual information.

3.3 Textual Information Extraction

To obtain textual information in an infographic, two steps are carried out. First of all, text components are grouped accordingly to form text blocks. Each text block contains a sentence, a phrase or a number. For text grouping, the method proposed by Yuan et al [22] is used. The grouping function is defined as:

$$f(s_1, s_2) = \sqrt{\frac{k s_1 s_2}{s_1 + s_2}} \qquad (1)$$

where s_1 and s_2 are the sizes of the two components and k is an adjustable parameter that is used to determine the grouping level. The size of a component is in terms of the number of black pixels belonging to the component. If the calculated *f* is smaller than the distance between the two components, the components are considered belonging to the same text block. The advantage of this method is that all values can be obtained easily and it is rotation invariant. In our case, the value of k is set to 10. An example is given in Figure 3(d).

OCR is then applied to each text block to recognize its content. We use the OCR functions provided in the Scansoft Omnipage Capture SDK package. The error rate of character recognition process heavily depends on the quality of the infographics. Since the OCR accuracy is not the main focus of our work here, we choose to manually correct the OCR errors.

After text grouping and OCR, two types of textual information are obtained. Firstly, the bounding box of each text block indicates locational information of the text block. Secondly, the OCR result provides the electronic text corresponding to each text block.

Table 1. Major roles of text in infographics

Block label	Role in infographics	Type of role
CTTL	Title of the infographic	Descriptive
XTTL	Title of x-axis	Descriptive
XLAB	Label along x-axis	Implicative
XUNT	Unit of x-axis	Implicative
YTTL	Title of y-axis	Descriptive
YLAB	Label along y-axis	Implicative
YUNT	Unit of y-axis	Implicative
DVAL	Data value	Descriptive
DLAB	Data label	Descriptive
LGND	Legend name	Descriptive
OTHR	Other description	Descriptive

3.4 Text/Graphics Association Unit

The task of associating text blocks with graphical symbols is modeled as a problem of classifying text blocks into different roles in an infographic. In other words, we need to find out which text corresponds to which graphical symbol and what kind of role does the text play. After going through a collection of infographics, we have identified 11 text roles in infographics, which are summarized in Table 1. From the table, it can be seen that most text blocks are attached to certain graphical symbols, such as the x-axis label etc., while some text blocks play a global role, such as the title of the whole infographic etc.

Furthermore, the 11 roles of text are put into two major types, as shown in the last column in Table 1. First of all, a text block can be used to describe a graphical symbol or the whole infographic. Such text blocks are called *descriptive* blocks. On the other hand, some text blocks do not provide any description. Rather, they join

with the graphical symbols to imply information that is not directly presented. Such text blocks are called *implicative* blocks.

Based on the information about the text blocks and graphical symbols, five features are defined for training the association rules. Three of these features capture explicit locational relationship between text blocks and graphical symbols, one feature makes use of the implicit characteristics of a text block itself, and the last one represents the global positional information. We hypothesize that the text block and graphical symbol to be associated together are nearest neighbor of each other. This hypothesis is based on the argument raised by Larkin and Simon [14] stating that information in diagrammatic representation is organized by location. This hypothesis can significantly reduce the amount of information to be processed by the classifier since only the nearest symbol is examined. The details of the features are as below.

- **Distance between a text block and the nearest graphical symbol.** The value of this feature is real. The distance between a text block T_i and a graphical symbol G_j is defined as:

$$D(T_i, G_j) = \min D(E_{il}, E_{jk}), \forall E_{il} \in T_i \forall E_{jk} \in G_j \qquad (2)$$

  where E_{il} is one of the four edges of the bounding box of T_i, and E_{jk} is one of the edges of graphical symbol G_j. Every graphical symbol G_j can be treated as a composition of edges. Function $D(E_{il}, E_{jk})$ is a basic geometric function that calculates the shortest distance between two edges. There is a special case where the text box is inside the graphical symbol (a 2D shape in this case), and the distance is set to zero. This is detected by checking whether the center of bounding box is inside G_j. By calculating the distance between a text block and all existing graphical symbols, the symbol that is nearest to the text block is identified. Only the distance between the text block to the nearest graphical symbol is stored.

  If we directly use absolute distance as the value of this feature, then it may be affected by the size of the image because larger images tend to result in longer distances. To remove this effect, we normalize the coordinates of every point p(x, y) by dividing x by image width W and dividing y by image height H. After normalization, both x and y falls into the interval [0, 1). Then the distance calculated is independent of the image size.

- **Type of the graphical symbol that is nearest to a given text block.** This feature is nominal. Although the set of graphical symbols depends on the type of infographics, there are some commonly used symbols. These symbols form the universal set of graphical symbols and each element in the set is given a label. For example, graphical symbols may be labeled as "X-AXIS", "Y-AXIS", "BAR", "WEDGE" etc. The value of this feature is actually determined together with the first feature. If the text box is inside a graphical symbol, then that symbol is treated as the nearest to the text box.

- **Relative position between a text block and a graphical symbol.** This feature is nominal. If we use the center C of a graphical symbol as the reference point, the angle between the center of the text bounding box and the center of the symbol can be any value from [0, 2π). In our work, we quantize the angles into 8 intervals. Then the relative position between a text box and a graphical symbol can be represented as one of the 8 labels (T = top, B = bottom, L = left, R = right, TL = top-left, TR = top-right, BL = bottom-left, BR = bottom-right). Again this feature is only calculated between the text block and its nearest neighbor. If the text box is inside the graphical symbol, then the relative position is NIL.

- **String checks.** We also use a string parser to check if a given text string can be interpreted as an integer or a floating point number. This is to find out whether the text directly represents values. The parser is written as a Boolean function *isNumber()* which returns true when the text string can be parsed to integer or floating point number, and false otherwise. This feature is calculated implicitly for any given text block.

- **Centricity of a text block.** This feature is calculated to measure how close a text block is to a bisector of the whole image. It is a global feature and its value is real. The value is normalized to the interval [0, 1], where 0 means the text block lies on the bisector and 1 means the text block is farthest from the bisector. The calculation is done for both the horizontal and vertical bisector, so there are actually two values. These two values are stored separately in the feature vector. This feature is useful to classify text blocks with global roles, such as chart title etc.

The next step is to use a machine learning algorithm to learn association rules from training examples. A set of training images are collected from the internet or scanned documents. Text blocks were extracted from these images and were manually assigned a block label. Every labeled text block forms a training example. The feature vector contains value of the five features defined and the correct class that the text block belongs to, for example <0.047420, Y_AXIS, L, true, 0.23, 0.78, YLAB>. The learning algorithm used by our system is C4.5 [15], a decision tree learner used in many machine learning tasks. C4.5 is the extension of the ID3 algorithm and it allows continuous attributes such as the distance feature used in our work.

During actually classification, the same set of features is calculated from the text blocks extracted from testing images. The trained classification rules are then applied to assign the corresponding label to each text block. Figure 3 (f) shows an example of association results. As the association is a two-way process, the content of a text block is also attached to the corresponding graphical symbol, for future interpretation purpose.

3.5 Interpretation and Description Generation

To achieve high-level understanding, the textual and graphical information are combined together to further imply information that is not directly presented. One such information is the data values. Infographics such as charts provide visualization of tabular data. The core information of the tabular data consists of data label plus data value. After associating text with graphics, the system further performs interpretation to obtain both the data

```
<infographic>
<type>2D Line Chart</type>
<title>-</title>
<x_axis><axis_title>Year</axis_title>
        <labels><label>1982</label>
                ……
                <label>1988</label></labels></x_axis>
<y_axis><axis_title>-</axis_title>
        <axis_unit>fatalities</axis_unit>
        <labels><label>0</label>
                ……
                <label>1600</label></labels></y_axis>
<data_set>
<data><label>1982</label><value>1080</value></data>
<data><label>1983</label><value>1156</value></data>
        ……
<data><label>1988</label><value>1590</value></data>
</data_set>
</infographic>
```

(a) An XML description for the infographic in Figure 1

- Figure <f_no> contains a <c_type> chart with the title <c_title>.
- The data contained are <x_title> versus <y_title>.
- Data entry <d_label> has a value of <d_value> <y_unit>.

(b) Sample templates for NL sentence generation

> Figure 1 contains a line chart with no title. The title of the x-axis is year. Data entry 1982 has a value of 1080 fatalities. Data entry 1983 has a value of 1156 fatalities. Data entry 1984 has a value of 1250 fatalities. Data entry 1985 has a value of 1181 fatalities. Data entry 1986 has a value of 1163 fatalities. Data entry 1987 has a value of 1393 fatalities. Data 1988 has value of 1590 fatalities.

(c) Sample NL sentences generation for the same infographic

Figure 4. Illustration of descriptions generation

label and the data value. There are two rules for finding out data labels.

Rule 1: If the data label directly appears as a *DLAB* text block in the infographic, then it is directly associated with the corresponding graphical symbol representing the data component using our method.

Rule 2: If no text block is classified as data label, then labels attached to the axis are aligned with data components in the x-y axis area to assign the data label. The alignment rule is based on the relative position between axis labels and data components.

To obtain value for each data entry, similar rules are applied:

Rule 1: If the data value appears as a *DVAL* text block, then it is directly associated with a data component using our method and the value is immediately available.

Rule 2: If no text block is classified as data value, then the value needs to be calculated. The calculation depends on the domain knowledge of each chart type about which attribute of a data component should be looked at. For example, in bar charts we look at the height of each bar shape but in pie charts we look at the angle of each wedge instead. Without the domain knowledge, the calculation cannot be carried out. The type of the value (integer or float) calculated agrees with the type of the axis label, if there is any.

Following the interpretation step, descriptions of the infographic is generated. To facilitate other applications that may require different representations of information, the system generates two different types of description: XML format description and natural language description.

The XML format description is used to represent the information contained in the given infographic image in a tabular form. The hierarchical XML format is defined as follows.

At the top level, the tag <infographic> is used. It contains the following parts:

- <type>: the type of the infographic.
- <title>: the title of the infographic if it exists.
- <x_axis> and <y_axis>: the existence of x-y axes depends on the type of the chart. If they do exist, <axis_title> shows title of each axis. <labels> contains a set of <label> attached to each axis. <unit> specifies unit used by each axis.
- <data_set>, the tabular data obtained from the infographic image. Each data entry has a <label> and a <value>.

Figure 4 (a) shows part of the XML format description generated for the infographic image in Figure 1. The XML description presents the chart information in a tabular form, which makes it convenient for some applications such as query processing and re-construction of the infographic etc. Due to space limitation, the Document Type Definition (DTD) part of the XML description is not shown. The DTD is type independent, thus some fields that are invalid for some chart types are given a value of "NIL" during generation.

Some application, on the other hand, works on text in the natural language form. An example is the question answering that is continuously being explored in the information retrieval (IR) and information extraction (IE) research community. To be able to apply existing question answering techniques, it is required that the result of infographic understanding should be presented in a natural language form as well. In other words, the description should consist of a set of natural language (NL) sentences. To do so, a set of templates is defined. The templates are type independent, and they cover all the components in the original infographic. Figure 4 (b) lists some of the templates defined. The detailed values used to fill in the templates are obtained through the text/graphics association process and the interpretation process. After all the sentences are generated, they are combined to form a single paragraph, such as the one in Figure 4 (c).

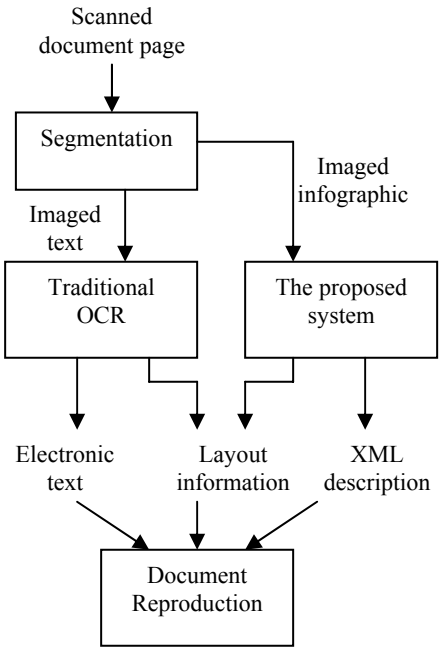

Figure 5. Joining the proposed system with traditional OCR system.

4. SAMPLE APPLICATIONS

4.1 Supplement to OCR System

Traditional OCR systems focus on recognizing the text from input document images. For other objects in the document, such as figures and other drawings, the OCR system will leave them untouched. Thus in the output file, the figures and drawings will remain as images. There are mainly two parts of output from a traditional OCR system: the recognized text in the electronic form and the layout information of the original document.

Although our system does not aim to provide a general solution to recognize all the objects untouched by OCR systems, it can be integrated with an OCR system to recognize the infographics in the documents. Furthermore, the information carried by the XML description can be used to easily re-construct the same infographic in an electronic form. The re-constructed infographic, together with the electronic text and the layout information, can be used to reproduce the original document, as shown in Figure 5.

A key issue is to find out the location in a document page for the re-constructed infographic to be inserted. This is important because one of the desired features of an OCR system is that both structural layout and logical layout of the original document should be preserved as much as possible. To handle this issue, there are two ways to do it. Typical OCR techniques define a document element as either a text paragraph or a figure, and determine the reading order among these elements [2]. If such reading order is available, then the elements before and after the infographic are known. The infographic is re-constructed and placed between the two elements. Otherwise, the system makes use of the figure caption below (or above in some cases) the infographic and search for its first appearance in the text. The paragraph containing such figure caption is considered as the element before the infographic, and the following paragraph will be treated as the element after the infographic. For example in Figure 1, the reading order specifies that the infographic is the first element in the document and the zoning information further suggests that it is placed horizontally at the center of the document. Thus in the reproduced document, the infographic is drawn first, in a similar position.

4.2 Enriching Information for Question Answering

So far, question answering techniques mainly focus on text-based documents, by extracting sentence level answers to factoid questions [16] or definitional questions [7, 19]. For a review of the question answering systems, please refer to [10]. For documents with infographics, the information contained in infographics may not be fully reflected in the text. Thus for questions related to the infographics, traditional question answering methods are not able to return an answer. In this situation, the result obtained by our system helps because it brings enriched information from the infographics.

Table 2. Set of basic queries to the infographics

Query Type	Return Type	Description
Max.	Value or label	Maximum among all existing values.
Min.	Value or label	Minimum among all values.
Avg.	Value or label	Average value of all values selected.
Between	Value or label	Between two values directly specified by user, or between two components specified by user using their labels.
Greater than	Value or label	Data components whose value is greater than the value directly specified by user or the value of the data component specified by user.
Less than	Value or label	Data components whose value is less than the value directly specified by user or the value of the data component specified by user.
Equal to	Value or label	Data components whose value is equal to the value directly specified by user or the value of the data component specified by user.

Based on our survey among human testers, some questions related to the infographics are factoid questions, which try to find out facts from the infographics. For example for the infographic in Figure 1, a question is: “How many fatalities were there in the year 1984?” For this type of questions, the original sentences in the text do not contain the require information. However, the natural language description generated by our system covers such

information. Thus it can be used to find out the answer. To facilitate the question answering processing, the NL description is inserted into the document as an additional part of the text. The insertion point is determined based on the idea described in the previous section. As the NL description contains natural language sentences, it can be handled by most traditional QA techniques.

Another type of questions purely refers to the infographics, and the answer cannot be obtained in a straightforward way. For example for the infographic in Figure 1, questions like "From when to when does the number of fatalities decrease?" or "What is the maximum number of fatalities among all years?" This type of questions is actually like data queries, except that the questions are in natural language form. Processing query-like questions is different from the two kinds of QA problems mentioned. It requires two major steps: translating a natural language question to query, and processing the query to generate the answer. For the translation from sentence to query, we used the method described in [13]. For query processing, we define a set of basic queries that are frequently raised by users. They are shown in Table 2. More complicated queries can be handled by transforming them into a combination of basic queries.

5. EXPERIMENTS AND DISCUSSIONS

200 imaged infographics were collected for training and testing the graphics recognition module and the text/graphics association module. Most of the images are scanned black-and-white images, while the rest are color images downloaded from the web. Out of these 200 imaged infographics, there are 80 2D bar charts, 60 2D line charts, and 60 pie charts that are 2D or 3D.

Table 3. Performance of graphics recognition

Chart type	Number of images	Correctly Recognized	Accuracy (%)
2D bar	80	75	93.75
2D pie	48	43	89.58
3D pie	12	11	91.67
Line	60	51	85.00
Overall	200	180	90.00

The performance of the graphics recognition module was evaluated by the result of the model matching. As mentioned earlier, the model matching uses all graphical symbols to compare against the domain knowledge for each chart type and calculates the likelihood between the given image and a chart type. The chart type returning highest likelihood value is deemed to be the type of the given image. Each of the 200 infographic images was processed by the system and was assigned a type. The number of images whose type was correctly recognized is shown in Table 3. Most of the errors are due to the failure to recognize certain graphical symbols (such as the axis lines) or the failure to satisfy certain global constraints (such as the summation of angles should be equal to 2π). The recognition accuracy of 3D pie chart is higher than that of the 2D pie charts in the table. However, this should not be a general conclusion. Instead, it is due to the relatively smaller number of 3D pie chart images tested and their relatively better image qualities.

To evaluate the performance of the text/graphics association module in the system, all text blocks were manually assigned the correct block label beforehand. There are 1222 text blocks in the bar charts, 789 text blocks in the line charts and 246 text blocks in the pie charts. All the images were fed into the system for extracting text blocks and graphical symbols automatically. Then feature vectors were calculated for each text block identified by the system and were used for training and testing. 10-fold cross validation was used to measure the accuracy of the classifier trained by the C4.5 decision tree learner. The average classification accuracy for each chart type is shown in Table 4. The reason that the accuracy for pie charts is relatively lower than the other two types is that bar charts and line charts have relatively more regular layout than pie charts.

Table 4. Average text block classification accuracy

Chart Type	Average Accuracy
Bar chart	88.22%
Line chart	91.00%
Pie chart	74.79%

Table 5. Comparison of question answering performance with and without information from infographics

Type of question	No. of questions	Correctness	
		Without infographic	With infographic
Factoid	50	38 (76%)	45 (92%)
Query-like	50	-	42 (84%)

We also conducted a small experiment to test how much the system output helps with the question answering process. 10 scanned document pages were selected from the University of Washington document database I as the testing images. Human testers were asked to give 5 factoid questions and 5 query-like questions to each document page. To involve the NL description of infographics in the question answer process, one of the 5 factoid questions was required to be related to the infographic. We used the question answering system developed by Cui et al [7] to process both the questions and the documents to find out the answer to each question. Cui's work uses probabilistic lexico-syntactic pattern matching, also known as *soft pattern matching*, and the result is a ranked list of sentences. In our experiment, only the sentence at the top of the ranked list (i.e. with the highest matching score) was considered as the correct answer. On the other hand, the query-like questions were also parsed based on the work in [13] and the resulting queries were processed by the system to generate answers. The performance evaluation of the question answering is shown in Table 5. From the table, it can be seen that the original textual information is not sufficient to handle all the factoid questions related to the infographics. With the NL description added, the performance of the question answering system is improved. The query-like questions are

handled equally well, with most of the errors due to the failure to parse a sentence into the correct query.

6. CONCLUSION

In this paper we introduce a system for recognizing and interpreting imaged infographics in the documents. The problem is hard because it involves text recognition, graphics recognition and associating the two sources of information together. The details of the major modules in the system are presented in the paper, including extraction of graphical and textual information from the input infographic, the association of the information from two modalities, the interpretation process and the generation of descriptions for the infographic. Two applications of the system outcome are also addressed here, namely the supplement to OCR system and the provision of enriched information for question answering methods. The system's performance was tested using a collection of imaged infographics downloaded from the internet or cut from scanned documents. Furthermore, scanned document pages from UW database I were also used to test how the system helps with the question answering application. At the moment, we focus on the scientific charts, a commonly used type of infographics. To further enhance the system's capability, handling of infographics with more complex layout and special effects such as gradient color or textures are to be studied in the future.

7. ACKNOWLEDGEMENTS

This research is supported by A*STAR grant 0421010085 and NUS URC grant R252-000-202-112.

8. REFERENCES

[1] S. Bergler, C.Y. Suen, C. Nadal, N. Nobile, B. Waked, and A. Bloch, Logical block labeling of diverse types of document images, *DLIA'99*, 1999, 4470-4475.

[2] T. M. Breuel, Layout Analysis based on Text Line Segment Hypotheses, *DLIA'03*, Edinburgh, Scotland, August, 2003.

[3] S. Carberry, S. Elzer, N. Green, K. McCoy and D. Chester, Extending Document Summarization to Information Graphics, *Proc. of the ACL Workshop on Text Summarization*, 2004.

[4] S. Carberry, S. Elzer, N. Green, K. McCoy, and D. Chester, Understanding Information Graphics: A Discourse-Level Problem, *Proc. of SigDial*, 2003, 1-12.

[5] W. S. Cleveland, *The elements of graphing data*, Chapman and Hall, New York, 1985, 1994.

[6] W. S. Cleveland, *Visualizing data*, Hobart Press, Summit, New Jersey, USA, 1993.

[7] H. Cui, M.-Y. Kan and T. S. Chua, Unsupervised Learning of Soft Patterns for Definitional Question Answering, *Proc. of the Thirteenth World Wide Web conference (WWW 2004)*, New York, May 17-22, 2004, 90-99.

[8] A. Fitzgibbon, M. Pilu, and R. B. Fisher, Directed Least Square Fitting of Ellipses, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21, 1999, 476-480.

[9] R. P. Futrelle, I. A. Kakadiaris, J. Alexander, C. M. Carriero, N. Nikolakis, J. M. Futrelle, Understanding diagrams in technical documents, *IEEE Computer*, Vol.25, 1992, 75-78.

[10] L. Gillard, P. Bellot, M. El-Bèze, Evaluations of Question Answering and Evaluations of the Evaluation, *The fifth Int. Conf. on Language Resources and Evaluation, LREC 2006*, Genoa, Italy, 24-26 May 2006.

[11] W. Huang, C. L. Tan and W. K. Leow, Model based chart image recognition, *6th Int. Workshop on Graphics Recognition*, *GREC'03*, 2003, 87-99.

[12] W. Huang, S. Zong and C. L. Tan, Chart image classification using multiple-instance learning, *WACV'07*, Feb 21st-22nd, 2007, Austin, Texas, USA.

[13] R. J. Kate and R. J. Mooney, Using String-Kernels for Learning Semantic Parsers, *Proc. of the Joint 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL-2006)*, Australia, July 2006, 913-920.

[14] J. H. Larkin and H. A. Simon. Why a Diagram is (sometimes) Worth Ten Thousand Words. In *Cognitive Science*, Vol. 11, No. 1, 1987, 65-100.

[15] R. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[16] D. Ravichandran and E. Hovy, Learning Surface Text Patterns for a Question Answering System, *Proc. of ACL'02, Philadelphia*, July 2002, 41-47.

[17] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy and P. Dosch. Text/Graphics Separation Revisited. In *5th Int. Workshop on DAS*, 2002, 200-211.

[18] F. Wang and M. Y. Kan, NPIC: Hierarchical synthetic image classification using image search and generic features, *Proc. of Conf. on Image and Video Retrieval*, 2006, 473-482.

[19] J. Xu, R. M. Weischedel and A. Licuanan, Evaluation of an extraction-based approach to answering definitional questions, *Proc. of SIGIR '04*, Sheffield, UK, 2004, 418-424.

[20] N. Yokokura and T. Watanabe, Layout-Based Approach for extracting constructive elements of bar-charts, *Graphics recognition: algorithms and systems, GREC*, 1997, 163-174.

[21] J. Yu, J. Hunter, E. Reiter and S. Sripada, Recognizing visual patterns to communicate gas turbine time-series data, *ES2002*, 2002, 105-118.

[22] B. Yuan and C. L. Tan. A Multi-level Component Grouping Algorithm and Its Applications. In *8th Int. Conf. on Doc. Analysis and Recognition, ICDAR'05*, 2005, 1178-1181.

[23] Y. Zheng, C. Liu, X. Ding and S. Pan, A Form Frame-Line Detection Algorithm Based on Directional Single-Connected Chain, *Journal of Software,* Vol. 13, 2002, 790-796.

[24] Y. Zhou and C. L. Tan, Hough-based Model for Recognizing Bar Charts in Document Images, *SPIE conference on Document image and retrieval*, 2001.

[25] Y. Zhou and C. L. Tan, Learning-based scientific chart recognition, *4th IAPR Int. Workshop on Graphics Recognition, GREC'01*, 2001, 482-492.

[26] Y. Zhou and C. L. Tan, Coordinate systems reconstruction for graphical documents by Hough feature clustering and geometric analysis, *Int. Conf. on Pattern Recognition, ICPR'04*, Cambridge, UK, 23-26 Aug 2004.