A Multi-Level Table Evaluation Method For Plain Text Documents

Vanessa Long, Steve Cassidy and Robert Dale
Centre for Language Technology
Division of Information and Communication Sciences
Macquarie University
Sydney, Australia
{vanessa, cassidy, rdale}@ics.mq.edu.au

Abstract

Due to their expressive powers, tables are popularly used in documents. The presence of tables in documents creates needs for table-processing algorithms. Over the years, many systems were developed , and their performance, usually in terms of recalls and precisions, were reported. According to current practice, systems are tested using some propriety data. This raises some important questions: how should we interpret the results? how should we compare different systems objectively? how do we know that the high recall and precision rates do not come at the expense of making mistakes in other parts of the table-processing task. In this paper, we would like to address these problems by proposing a multi-level table evaluation method.

1. Introduction

Due to their expressive powers, tables are popularly used in documents. Tables have been receiving much research attention since the early dates of document processing. Over the years, many systems were developed to address various issues in table processing, and their performance results, usually in terms of recalls and precisions, have been reported. We cannot judge the real performance of the systems by directly compare the reported recalls and precisions for the following reasons.

1. different systems might be designed for addressing different research issues.
2. different systems might be tested using some proprietary data with different levels of complexity.
3. the reported results do not contain information about the overall performance of the systems. High recalls and precisions for specific tasks do not necessary imply good overall performance, as the results might come at the expense of making mistakes in other parts of the table-processing task.

This raises an important question: how should we interpret the results, especially when the systems are designed for different purposes and are tested using different data? Despite effort had been made to standardise the evaluation process [1, 2, 3, 4, 5], getting a common agreed standard in place is still an outstanding problem, partly due to the complex structures of tables.
Table-processing is a complex problem with many subproblems: some tasks focus on detecting table locations in the documents (such as table boundary identification); some tasks interest in the inner structures of tables (such as table structure recognition); some tasks want to find out the functional structure of table components (such as table interpretation). According to our observation, table-processing tasks can be classified into the following categories.

1. structural tasks: these are the tasks that involve determining table locations and table structures. Examples of these tasks are listed below.
   (a) table boundary identification
   (b) cell segmentation
   (c) column-row structure identification
   (d) multi-line cell identification
   (e) spanned cell identification
2. interpretation tasks: these are the tasks that build on top of the basic tasks. Examples of these tasks are listed below.
   (a) header identification (for both row headers and column headers)
   (b) header type identification (e.g. the relationships between headers and the member cells)

(c) cell content type identification (e.g. the cell function in the table)

Evaluations of structural tasks are different from the evaluations of the interpretation tasks: the former requires information contained in the input document whereas the latter requires semantic interpretation from the evaluators. Since reporting recalls and precisions does not provide sufficient information for comparing systems, this paper explores an alternative evaluation method that provides complete views of the performance for the structural tasks.

2. Terminology

In this paper, we will frequently refer to the following terminology, which was introduced in [6].

1. A *line-art line* is a line whose purpose is to serve as a vertical delimiter. Line-art lines typically consist only of punctuation characters, with the hyphen and underscore being very common, and the plus sign being used to indicate column boundaries.
2. A *column delimiter* is a sequence of characters that separate table columns.
3. A *cell segment*, marked by one or two column delimiters, is a sequence of text tokens embedded in the same row-line.
4. A *cell* is the most basic, semantically complete unit in a table. A simple table cell consists of just one cell segment, but a multi-line cell may contain two or more cell segments from adjacent row-lines.
5. A *row* is one or more row-lines that contain cells that are horizontally aligned. Sometimes a row can be just a row-line, but very often a row consists of multiple row-lines.

3. The Markup Specification

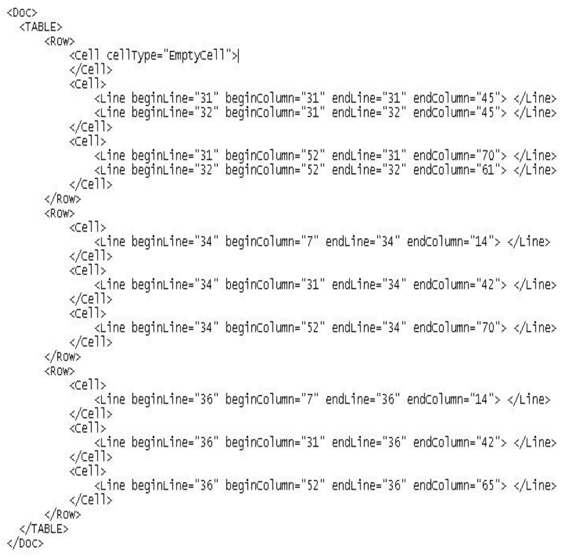
One of the prerequisites for table evaluation is the unambiguous markup of the experimental and ground-truth data. The following markup guidelines are proposed to support the evaluation process.

1. a character in a document is referenced by its line index and column index. A string is referenced by the positions of its first and last characters.

2. Only those strings that contribute to the table content should be included in the ground-truth markup. Blank lines,line-art lines, column and row delimiters, and leading and trailing white spaces should be excluded from the markup.

3. Empty cells are all markup by the string of $\langle CellcellType = "emptyCell" \rangle$ $\langle /Cell \rangle$
4. Non-spanned content cells may contain one or more cell segments. Each cell segment is specified by the positions of its first and last characters in a pair of $\langle Line \rangle$ tags.
5. The markup for spanned cells is very similar to the markup for non-spanned cells, except that additional spanning properties are specified within the $\langle Cell \rangle$ elements. If a cell is horizontally spanned, then both the *SpanColStartIndex* and *SpanColEndIndex* properties must specified. If a cell is vertically spanned, then both the *SpanRowStartIndex* and *SpanRowEndIndex* properties must specified.
6. The markup document follows the XML syntax: at the highest level, a markup document contains a pair of $\langle Doc \rangle$ tags. The $\langle Doc \rangle$ element contains any number of $\langle Table \rangle$ elements, which contain one or more rows. At the lowest level, the markup document contains the $\langle Line \rangle$ elements.

Figure 1 shows an example of a markup document.



```
<Doc>
  <TABLE>
      <Row>
          <Cell cellType="EmptyCell">
          </Cell>
          <Cell>
              <Line beginLine="31" beginColumn="31" endLine="31" endColumn="45"> </Line>
              <Line beginLine="32" beginColumn="31" endLine="32" endColumn="45"> </Line>
          </Cell>
          <Cell>
              <Line beginLine="31" beginColumn="52" endLine="31" endColumn="70"> </Line>
              <Line beginLine="32" beginColumn="52" endLine="32" endColumn="61"> </Line>
          </Cell>
      </Row>
      <Row>
          <Cell>
              <Line beginLine="34" beginColumn="7" endLine="34" endColumn="14"> </Line>
          </Cell>
          <Cell>
              <Line beginLine="34" beginColumn="31" endLine="34" endColumn="42"> </Line>
          </Cell>
          <Cell>
              <Line beginLine="34" beginColumn="52" endLine="34" endColumn="70"> </Line>
          </Cell>
      </Row>
      <Row>
          <Cell>
              <Line beginLine="36" beginColumn="7" endLine="36" endColumn="14"> </Line>
          </Cell>
          <Cell>
              <Line beginLine="36" beginColumn="31" endLine="36" endColumn="42"> </Line>
          </Cell>
          <Cell>
              <Line beginLine="36" beginColumn="52" endLine="36" endColumn="65"> </Line>
          </Cell>
      </Row>
  </TABLE>
</Doc>
```

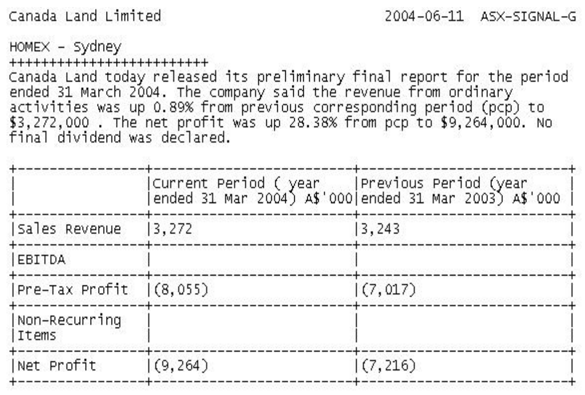
Table 1. An example of a markup document

4. Evaluation and Error Analysis

The goal of evaluation is to find out the differences, in terms of the number of errors and the type of errors, between experimental results and ground-truth data. The table representation section (4.1) defines '*what*' will be compared, and the evaluation steps define '*how*' to compare the target objects, and the error analysis section (4.3) gives feedback of the evaluation result in an informative format.

4.1. Table Representation

Tables in documents can be represented by rectangles with holes: the table boundary defines the rectangle boundary, and the holes inside the rectangle represent the locations of the table content. Figures 2 and 3 show a document and its table representation using rectangles. In theory, the evalu-



```
Canada Land Limited                               2004-06-11  ASX-SIGNAL-G

HOMEX - Sydney
+++++++++++++++++++++++++
Canada Land today released its preliminary final report for the period
ended 31 March 2004. The company said the revenue from ordinary
activities was up 0.89% from previous corresponding period (pcp) to
$3,272,000 . The net profit was up 28.38% from pcp to $9,264,000. No
final dividend was declared.

+----------------+-------------------------+--------------------------+
|                |Current Period ( year    |Previous Period (year     |
|                |ended 31 Mar 2004) A$'000|ended 31 Mar 2003) A$'000 |
+----------------+-------------------------+--------------------------+
|Sales Revenue   |3,272                    |3,243                     |
+----------------+-------------------------+--------------------------+
|EBITDA          |                         |                          |
+----------------+-------------------------+--------------------------+
|Pre-Tax Profit  |(8,055)                  |(7,017)                   |
+----------------+-------------------------+--------------------------+
|Non-Recurring   |                         |                          |
|Items           |                         |                          |
+----------------+-------------------------+--------------------------+
|Net Profit      |(9,264)                  |(7,216)                   |
+----------------+-------------------------+--------------------------+
```

Table 2. An example of an input document

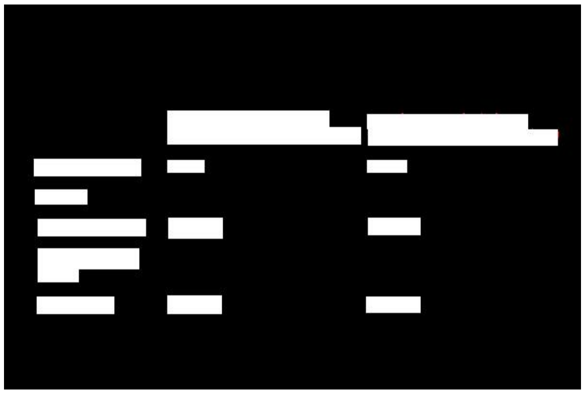


Table 3. The table representation of the input document using a rectangle containing holes

ations for the structural tasks can be achieved by judging the intersecting patterns between the rectangle representing the experimental result and the rectangle representing the ground truth result. The experimental result and the ground truth result are exactly the same if and only if their corresponding rectangles are exactly matched. In practice, comparing the intersecting patterns between two rectangles with arbitrary number of holes is very hard. To overcome this problem, we propose to use bounding boxes (solid rectangles) to represent table components, and we measure the overlapping patterns of the bounding boxes at three levels: table-level, row-level and cell-level.

4.2. Evaluation Steps

When evaluating table extraction results, we would like to determine the types of errors and the sizes of errors simultaneously. To reflect the fact that errors can occur at the table, row/column and cell levels, the proposed evaluation method solve the problem through a multi-level evaluation: table-level evaluation, row-level evaluation, and cell-level evaluation. Table structures can be complicated by the existence of empty cells and spanned cells. Empty cells and spanned cells will be evaluated separately at cell-level, and they are excluded from other levels of evaluation. Given an experimental result and the ground-truth result for a document, the evaluation steps can be summarised as the following.

1. Pre-processing. This step is to verify that both the experimental results and the ground-truth results meet the markup standard. For example, it makes sure that only legal markup tags are sued, and it checks that the same piece of information only appear once in the markup document.

2. Region extraction at table-level. This step constructs bounding boxes for all tables for both the experimental markup document and the ground-truth document according to the region definition in section 4.1.

3. Region extraction at row-level. Discarding the table boundaries, this step constructs bounding boxes for all rows in all tables for both the experimental markup document and the ground-truth document.

4. Region extraction at cell-level. Discarding both the table and row boundaries, this step separates spanned cells and empty cells from the table before it constructs the bounding boxes for all non-spanned content cells for both the experimental markup document and the ground-truth document.

5. Region comparison. This step is to compare the corresponding region lists for the experimental markup and ground-truth markup, and report the error types, which is discussed in section 4.3.

4.3. Error Analysis

Five error types have been identified in [2]: insertion error, deletion errors, merging errors, splitting error and partial overlapping errors. The proposed error analysis method reports all these error types at three levels: the table level, the row level and the cell level.

Given two rectangles $R1$ and $R2$, there are five possible intersecting types: exactly matched, $R1$ fully contains $R2$, $R2$ fully contains $R1$, $R1$ overlaps with $R2$ and $R1$ does not overlap with $R2$. Based on their intersecting pattern, the following error types can be derived.

1. $R1$ and $R2$ are exactly the same. This intersecting pattern corresponds to the exact match type, where the an element appearing in the ground-truth is also appearing in the experimental result as a whole element.

2. $R1$ contains $R2$. This intersecting pattern corresponds to the merging error type, where the an element appearing in the ground-truth is fully contained within an element in the experimental result.

3. $R2$ contains $R1$. This intersecting pattern corresponds to the splitting error type, where an element appearing in the experimental result is fully contained within an element in the ground-truth data.

4. $R1$ and $R2$ are partially overlapped. This intersecting pattern corresponds to the partial overlapping error type, where an element in the ground-truth data partially overlap with an element in the experimental result.

5. $R1$ and $R2$ are not overlapped at all. This intersecting pattern corresponds to two error types: the insertion error and the deletion error. An insertion error occurs when an element appears in the experimental result, and does not appear in the ground-truth markup. An deletion error occurs when an element appears in the ground-truth, and does not appear in the experimental result.

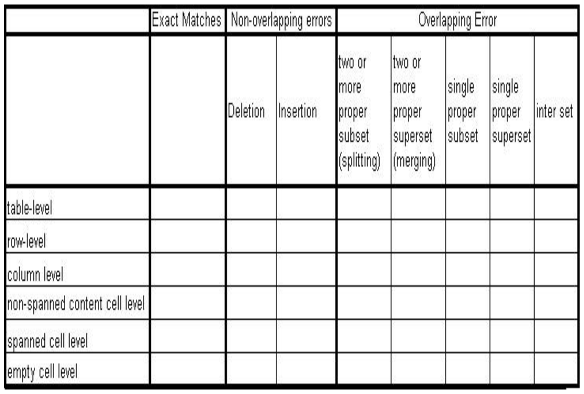
After the region comparison step, an error analysis result, such as the one in figure 4, is produced.

5. Future Work

We have proposed a table evaluation method and a markup strategy for plain text documents. Although bitmap documents do not fall into our topic domain, the proposed markup strategy and evaluation method could also work for bitmap documents with minor modifications.
We would like to extend our evaluation method to cover embedded tables in the future. When evaluating tables with embedded tables, four possible outcomes must be considered.

1. both the surrounding table and the embedded table are identified correctly.

2. only the surrounding table is identified correctly; the embedded table is incorrectly identified.



	Exact Matches	Non-overlapping errors		Overlapping Error				
		Deletion	Insertion	two or more proper subset (splitting)	two or more proper superset (merging)	single proper subset	single proper superset	inter set
table-level								
row-level								
column level								
non-spanned content cell level								
spanned cell level								
empty cell level								

Table 4. Multi-level evaluation and error analysis

3. only the embedded table is identified correctly; the surrounding table is incorrectly identified.

4. neither the surrounding table nor the embedded table is identified correctly.

References

[1] J. Hu, R. Kashi, D. Lopresti, G. Nagy, and G. Wilfong. Why table ground-truthing is hard. In *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, pages 129–133, Seattle, WA, USA, 2001.

[2] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Table structure recognition and its evaluation. In *Document Recognition and Retrieval VIII*, pages 44–55, 2001.

[3] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Evaluating the performance of table processing algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 4(3):140–153, 2002.

[4] F. Kboubi, A. H. Chabi, and M. B. Ahmed. Table recognition evaluation and combination method. In *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1237–1241, Seoul, Korea, 2005.

[5] T. Kieninger and A. Dengel. An approach towards benchmarking of table structure recognition results. In *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1232–1236, Seoul, Korea, 2005.

[6] V. Long, R. Dale, and S. Cassidy. A model for detecting and merging vertically spanned table cells in plain text documents. In *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1242–1246, Seoul, Korea, 2005.