Answer Retrieval From Extracted Tables

Xing Wei, Bruce Croft, David Pinto
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01002
{xwei,croft,pinto}@cs.umass.edu

ABSTRACT

Question answering (QA) on table data, which contains densely packed information in two-dimensional form, is a challenging information retrieval task. Data can be placed at a distance from the metadata describing it. The metadata itself can be difficult to identify given the layout of a particular table. This paper describes a QA system for tables created with both machine learning and heuristic table extraction methods. Our approach creates a cell document for each table cell. A probabilistic language model selects the most likely cell documents for the information need. The performance of the system is tested with government statistical data, and errors are analyzed in order to improve the system. We also apply these improvements on another type of table data set and show the experimental results.

Keywords

Question answering, tables, information extraction, metadata, conditional random fields.

1. INTRODUCTION

Question answering (QA) is a discipline of information retrieval (IR) that attempts to find specific answers in documents, relieving the user of having to scan retrieved documents for a desired information need. One source of these answers is data tables. Tables present an interesting problem for question answering. QA systems look for potential answer entities in a close relationship with query terms. However, in a data table, the answer may be rows and columns away from the text that could contain the query terms; the row names, column headers, titles and captions. Text tables in particular make this difficult, since the layout of these tables is as varied as their composers.

Another type of table, found in web pages, uses a markup language such as HTML or XML to designate the position of cells and presents a somewhat easier problem than text tables. However, HTML/XML tables are often used to format documents instead of presenting data. A content judgment must be made to see if the table contains useful data. In addition, the use of mark up is not consistent, so header lines and data cells still need to be identified. We focus on answer retrieval from text tables in this paper since it is the harder problem, and it is our belief that the techniques presented here will also be applicable to HTML/XML tables.

Tables provide a visual way to link metadata (headers, titles) with cell data (question answers). Linking cell data and metadata, especially from text tables formatted by humans, is a difficult task given the variations in layout among table authors. Those links are vital, however, in being able to match questions with their answers.

Extracting and associating data and metadata requires a series of accurate decisions. The table itself must be identified within the text document and data and header lines separated. Within the set of header rows, an algorithm must recognize the difference between titles and column headers and determine the span of each individual column header. Finally, row headers are identified. With this information in place, the data is combined creating a cell document for each table cell.

There are a number of approaches to these decisions. Previously, a heuristic system [10] and a machine learning system [11] using conditional random fields (CRF) decided on labels for individual lines of text files. The CRF uses finer labels and produces higher overall accuracy at labeling individual lines. These results are presented in section 4.1.1. However, accuracy on certain elements of the tables (especially headers) was lower than desired.

This paper describes solutions to the header identification problem and the effect of those solutions on answer retrieval. The addition of new features, an enlarged training set and better delineation of tables from text are explored as means to better QA performance. The results are compared to extraction using the heuristic and CRF extractors previously available.

In section 2 we present an overview about related work and key components, followed by a description of methods in the experiments, including the evaluation, in section 3. In section 4 we develop the table extraction and QA retrieval on government statistical data which contains mostly complex tables, and test the development on Wall Street Journal data that contains simpler tables in section 5. Section 6 contains a discussion of these results, and section 7 points out directions for future research.

2. OVERVIEW

2.1 Related Work

Question answering systems have many features in common [14]. The first step is traditional document retrieval. The second step is a search of those documents for an entity containing the answer. Often, a question classifier is used to determine an entity type (number, name, country) that is employed in the search for an answer.

Pinto et. al. [10] created a system, QuASM, based on this model. Retrieved documents were searched for answers based on finding named entities matching the class of the question. Part of this system involved the transformation of table data into cell documents. Based on the work of Pyreddy and Croft [13], a character alignment graph (CAG) was employed to find text tables in documents. Heuristic algorithms then extracted the cell data and matched it with its metadata. Results concluded that accurate tagging of table lines is important in building the representation used for information retrieval. However, since this system does not finely distinguish between types of header rows, it tends to draw in extraneous metadata. A study of the errors made by the QA system showed that the extraneous metadata is a leading cause of failure to retrieve the appropriate answers.

Better table extraction is needed. Matthew Hurst [2, 3, 4, 5] describes the problem of information extraction from tables as one of both layout and language: the elements of tables are potentially ambiguous; cells may span multiple columns or multiple lines; header information may lay across multiple cells; there may be a lack of continuity between table lines. While systems can be based on either layout or language, the combination of the two is necessary to resolve the ambiguities in the data of tables. Given a table, Hurst's model breaks tabular text into blocks, then determines what the blocks represent---using generative language models in both stages.

Pinto et al. [11] developed the use of conditional random fields to tag lines of documents, incorporating both language and layout. To better describe the layout of tables, twelve tags were used to describe the lines of tables versus four in the heuristic system. The features incorporated into the CRF deal with layout (spaces between cells) and language (keywords in header lines). The CRF achieved significantly higher accuracy in labeling individual lines of a document.

2.2 Table Extraction and Answer Retrieval

Table extraction and answer retrieval are the two main parts of the question answering system. Extraction transforms each data cell of a table into an individual document, which is the cell document, consisting of the cell data and the metadata drawn from titles, headers, etc. Answer retrieval finds the answer from the cell documents created during table extraction. It ranks the cell documents using a language modeling approach.

Table extraction has three key elements. The first is locating a table in a document. The second is determining the structures within the table, such as headers, footnotes, etc. The third is associating the various elements of a table with their related data cells to create a cell document that works as an answer passage. It is our hypothesis that improvements in the extraction algorithm will lead to improvements in answer retrieval.

3. METHODS

3.1 Table Extraction

Table extraction is at the heart of our QA system. It provides the information for the answers to our questions. We used two methods to extract our data; one based on heuristics and one based on CRFs.

3.1.1 Heuristic

The first step is to locate the tables in the file. The text is converted to a character alignment graph. The CAG shows the characters and spaces, and alignments of those two entities show structures that are table like. Each line of the CAG is given one of four labels, identifying Titles, Captions, Data and Non-table lines.

The second step is to match data cells with their appropriate headers. A number of heuristics were used to find the rows that contain header information and set a row header. The cells will be extracted with their row headers and column headers into metadata documents.

Here are some examples of the heuristics we used:

- A row with more than two gaps may be a table row. Gaps are large areas of white space in a row, and may indicate column structure.
- Rows at the beginning of the table are likely to be less regular than the rest of the table.
- Rows in a table with a similar number of cells are likely data rows.
- Cells in the first column are used as row headers.

See Pinto et. al. [10] for a complete description of the features.

3.1.2 Conditional Random Fields

Conditional Random Fields are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes. They are discriminative models. CRFs support the use of many rich and overlapping layout and language features, which are what we need in the application. See [7] for a complete description.

To perform table extraction, we set the lines of text as the data input sequence. Features are calculated for each line. We used thirteen features in three sets, white space features, text features and separator features. These features are based on the heuristic features in section 3.1.1, with some modifications. In addition we used feature conjunctions of the current line with the previous line, the current line with the following line and the two following lines together. The CRF uses Viterbi decoding to assign one of twelve labels to each line. A complete description of the labels and features is in Pinto et. al. [11]. We implemented our CRF extractor using Mallet [8] – a machine learning for language toolkit. After CRFs label lines, heuristic are applied to associate cell data with metadata.

3.2 Database Building

We built the database of cell documents using Lemur [9] – a language modeling and information retrieval toolkit. Databases were built with various combinations of stemming and stopping to

see how these variables affected performance. We use the Krovetz stemmer [6] and the default stopper of Lemur.

3.3 Answer Retrieval

Answer retrieval attempts to find the cells containing the answer. The cells were put into cell documents with metadata, and can be pulled in as answers when the documents are retrieved. So answer retrieval here is actually cell document retrieval.

The cell documents are ranked using language modeling techniques [1, 12]. The basic approach for using language models for IR assumes that the user generate a query as text that is representative of the "ideal" document. The task is then to estimate, for each of the documents in the database, which is most likely to be the ideal document. Thus, we rank documents by

$$P(D \mid Q) \propto \prod_{i=1}^{n} P(q_i \mid D) \qquad (1)$$

where D is an cell document, Q is a query and q_i is a query term in Q. We used interpolation with a collection model and Dirichlet smoothing to smooth probabilities [15]. The collection probabilities were estimated using the entire collection.

3.4 Evaluation Methods

3.4.1 Data Set

3.4.1.1 FedStats Data Set

A crawl of www.FedStats.gov performed in June 2001 gathered a large set of documents, many containing examples of text tables generated by government agencies. A heuristic chose a sub-set of these documents likely to contain tables. The data set for experiments was chosen randomly from these documents, which might or might not contain tables. Each line of these documents was labeled by means of a simple heuristic program. The machine labels were reviewed and corrected by a human reader.

A set of documents, 62 in all, containing 26,947 lines that have 5,916 table lines, was selected to be test data. For the training data, the original set has 52 documents, including 31,915 lines and 5,764 of them are table lines. To improve the system performance, we labeled another 225 documents, containing 244,965 lines with 61,576 table lines to enlarge the training set. So the new enlarged training set has 277 documents, with 276,880 lines and 67,340 are table lines.

3.4.1.2 Wall Street Journal Data Set

The tables in the FedStats data set are often complex, containing multi-level headers, super headers (spanning multiple columns), and a variety of sub-headers and section headers. Another data set taken from the Wall Street Journal (WSJ) has a different style and contains mostly simple tables. We used the Wall Street Journal set as a test set to confirm our models and improvements to them. We labeled 25 documents, with 2,296 lines and 449 of them are table lines to be test data; and 98 documents, with 5,683 lines that have 1,401 table lines, to be training data.

3.4.2 Table Extraction Evaluation

In the table extraction process, there are two parts—line labeling and cell association. Line labeling determines each line's label to show whether it is in a table, and if it is then what is its function in a table. Cell association is to associate a table cell with its title, headers and other metadata.

To evaluate line labeling, we have three measures: line accuracy, recall and precision. These three measures focus on different parts of line labeling performance. Line accuracy is the percentage of correct line labels over all lines. It shows the overall accuracy and general performance. Recall is the percentage of correct table labels over all table lines in the documents. Precision is the percentage of the table labels that the program labeled correctly. Precision and recall show the system performance on a certain set of labels relevant to QA performance. We also apply precision and recall to the set of header lines. In order to make the CRF results comparable with the heuristic results, multiple CRF labels are combined to make a single heuristic label. This is reasonable because it is obviously much more acceptable to label a <TABLEHEADER> as a <SUPERHEADER> than a <NONTABLE>.

To evaluate cell association, we have the system associate each cell with its relevant information based on correct line labels. Then we compute the precision and recall of the items that were associated correctly. From our experiments, cell association heuristics work very well, thus we have focused on the problem of line labeling with machine learning techniques in this discussion.

3.4.3 QA Retrieval Evaluation

To evaluate the QA retrieval results, fifty questions for the tables in the FedStats data set and fifty-three questions for the tables in the Wall Street Journal data set were generated by hand. The questions were generated from the random set of documents that was selected from each data set. A question asks about a cell in a table. For example, the answer to the question 'What percent of the public thinks chief executives are not ethical?' corresponds to the cell containing '58' in the table in figure 1. A full list of questions is given in appendix. The content in the answer cell is used to check if the retrieval documents have the right answer. Then the Mean Reciprocal Rank (MRR) is computed when 1, 5 or 100 documents are retrieved. The MRR of each individual query is the reciprocal of the rank at which the first correct response was returned in the first N responses or 0 if none of them contained a correct answer. The score for a sequence of queries is the mean of the individual query's reciprocal ranks. MRR is often used as a question answering performance measure.

	More Ethical	Less Ethical	The same	Not Sure/ No Answer
CEO's opinion	71%	10%	14%	5%
Public poll	23	58	9	10

Figure 1. An example table used as a question source

4. ANSWER RETRIEVAL FROM FEDSTATS TABLES

Previous work by Pinto et. al. [11] show that CRFs label documents accurately overall, but did not perform well on header lines specifically. Since recognizing headers is essential to creating correct cell documents, experiments were conducted to see if header accuracy could be improved.

4.1 Baselines

The results from the original system are set as baselines. The original training data set has 52 documents selected from the crawl of FedStats. These documents were randomly selected from a subset of the crawl likely to contain tables, based on the output of the heuristic table extractor. These documents contained 31,915 text lines and 5,764 table lines. Thirteen features, including white space features, text features and separator features were used in the original CRF program.

4.1.1 Table Extraction

Table 1 shows the performance of the original CRF. As a reference, Table 2 presents the results for the heuristic table extractor.

Table 1. CRF table extraction performance baselines

Line Accuracy		Precision	Recall
93.5%	Table lines	96.9%	83.9%
	Header lines	50.4%	57.4%

Table 2. Heuristic table extraction performance

Line Accuracy		Precision	Recall
80.4%	Table lines	56.1%	75.8%
	Header lines	11.1%	67.9%

These tables point out the weakness of both systems. The heuristic extractor has low precision, so it pulls in much extraneous metadata. The CRF extractor has high overall table precision, but low recall on pulling in headers, the metadata needed for the cell documents.

4.1.2 QA Retrieval

The MRR for QA retrieval from the original CRF system and the heuristic system are in Table 3.

Table 3. MRR for QA retrieval baselines

	MRR at	Original CRF	Heuristic
No stemming/ stopping	1	0.14	0.14
	5	0.178	0.171
	100	0.194	0.187
With stemming/ stopping	1	0.1	0.14
	5	0.149	0.168
	100	0.171	0.187

From the above results we can see that the heuristic method works almost the same or even better than CRF extraction on QA retrieval, although CRF extraction has higher line accuracy, precision and recall on table lines. It should be noted that performance is low on all measures, pointing to the difficulty in answering questions from tables compared to retrieving answers from text. An analysis of the errors indicated that the main problem is the low recall of header line labeling, as shown in Table 1 and Table 2. Table header lines are very important. They are the link that connects table cells with a query. Without headers as part of the cell document, a table cell will not be retrieved as an answer to a question.

The example in Figure 2 shows how CRF extraction missed the headers. The labels in the beginning of the lines are the labels that the CRF got for the following text. For the table in this figure CRF extraction mislabeled the beginning part of the table, including the title and the header, as <NONTABLE>. These lines do not contain the features which are indicative of titles and headers to this CRF model. The header is missing separator characters that usually indicate column names, and leading white space is absent from the title. Tables like this led to the development of new features to better capture this type of layout.

<BLANKLINE>
<NONTABLE>Table 2.1. The 1994 National Solid Waste Stream
<BLANKLINE>
<BLANKLINE>
<SEPARATOR>----------
<BLANKLINE>
<DATAROW>General Waste Residential CII Waste All Waste
<BLANKLINE>
<DATAROW>Category Waste Generated Generated Generated
<BLANKLINE>
<DATAROW> (Million Tons) (Million Tons) (Million Tons)
<BLANKLINE>
<SEPARATOR>----------
<BLANKLINE>
<DATAROW>Paper and
<BLANKLINE>
<DATAROW>paperboard 36.4 44.9 81.3
<BLANKLINE>
<DATAROW>Glass 10.7 2.5 13.2
…

Figure 2. An example of mislabeled headers

4.2 Improvements

4.2.1 New Features

In order to improve the CRF labeling results, we considered adding new features. A study of the errors made with the current set of features led to the creation of two new feature groups. One describes the layout of characters, especially the alignment of the current line with its neighboring lines. When a human looks at a table, there is clear overlap between the space and non-space characters in data rows and headers. Measurements of this overlap are integrated as new features.

Another group of features describes the language used in titles and headers. Our observation is that certain key words appear in the text. For example, the title may contain 'Table'.

Altogether, we developed tens of new features and tested them individually. Finally we selected 20 of them that have better results and are representative, and put these 20 features into the new system. Table 4 lists some of the new features and the percentage improvement obtained with that feature.

4.2.2 Increased Training Data

Each individual new feature had a small effect on precision or recall as shown in Table 4. Given that this was not sufficient improvement, the effect of the size of the training set was explored. To test the hypothesis that more training data would lead to better header labeling, a ten-fold cross-validation experiment was performed. The data in the test set and the original training set were pooled and split into ten similar size

parts, each containing the same number of documents. We tested the CRF performance for header lines on these ten small sets respectively with the other nine as a training set. The results were obtained without the new features detailed in 4.2.1 and are shown in Table 5.

Table 4. New features with their improvements

Features	Improvements
Percentage of duplicated blank/nonblank characters of this line with the nth above/next non-blank lines.	7% on recall
Percentage over all gaps of duplicated gaps with near borders of this line with the nth above/next non-blanklines.	8% on recall
Percentage of words beginning with capital letters	4% on recall
Contain 'table' or 'figure' at line beginning	7% on recall and 5% on precision
Contain formatting tags "< ... >"	5% on recall

Table 5. Cross-validation experiments, header accuracy

Test set	Original CRF		Heuristic	
	Precision	Recall	Precision	Recall
1	100%	81%	0.8%	69%
2	96%	88%	10%	70%
3	95%	99.9%	56%	80%
4	99%	98%	61%	80%
5	96%	66%	46%	98%
6	92%	86%	41%	89%
7	66%	17%	9%	44%
8	80%	92%	2%	77%
9	93%	59%	15%	78%
10	94%	89%	31%	75%
Overall	95%	86%	26%	78%

The results suggest that increasing the size of the training set will improve the recall and precision on headers. Therefore, we added the other 225 documents discussed in section 3.4.1 into the training set. In order to show the relation between the size of training data set and table extraction results, we split this enlarged training data set into four similar-size subsets and did four tests, in each of which we put one more subset from the training set.

The results are shown in Table 6. From these results we can see that the recall, especially the crucial header line recall, improved when the training set was enlarged.

Table 6. CRF table extraction performance by the size of training data set

Training Data Sets	Line Accuracy		Precision	Recall
1	90.9%	Table lines	98.0%	75.1%
		Header lines	59.7%	42.7%
2	91.2%	Table lines	91.6%	81.4%
		Header lines	78.9%	55.3%
3	90.2%	Table lines	97.3%	83.5%
		Header lines	67.2%	62.0%
4	92.2%	Table lines	97.2%	84.3%
		Header lines	62.9%	65.4%

4.2.3 Improved Header Tolerance

As discussed in section 3.4.2, performance when associating table cells with headers is good on perfectly labeled data. Unfortunately, at our level of header recall, we are still mislabeling header lines. The algorithm discussed in section 3.4.2 used a very strict cutoff to determine the boundaries of tables. For example, some <TABLEHEADER> lines in Figure 2 were labeled correctly after the CRF was improved. The new labels are in Figure 3. However, the highlighted line labeled <NONTABLE> caused the extraction program to miss the headers for this table.

```
<BLANKLINE>
<NONTABLE>  Table 2.1. The 1994 National Solid Waste Stream
<BLANKLINE>
<BLANKLINE>
<SEPARATOR>  ------------------------------------------------------------
<BLANKLINE>
<TABLEHEADER> General Waste  Residential   CII Waste     All Waste
<BLANKLINE>
<TABLEHEADER> Category       Waste Generated  Generated     Generated
<BLANKLINE>
<NONTABLE>                   (Million Tons)  (Million Tons)  (Million Tons)
<BLANKLINE>
<SEPARATOR>  ------------------------------------------------------------
<BLANKLINE>
<DATAROW>          Paper and
<BLANKLINE>
<DATAROW>          paperboard     36.4        44.9        81.3
<BLANKLINE>
<DATAROW>            Glass        10.7        2.5         13.2
…
```

Figure 3. An example of missed headers caused by <NONTABLE>

We changed the rules in the program and the main flow of the rules is the following:

- When the system encounters a super header line, like <TITLE>, <TABLEHEADER>, or <SUPERHEADER>, if the tolerance is 0 then give the tolerance a value, which is 7 in our current system; if the tolerance is more than 0 then the table ends in the above line.
- When the system encounters a sub header line, like <SUBHEADER> or <SECTIONHEADER>, give the tolerance a smaller value. We use 5 in our program.
- When the system gets a <BLANKLINE> or a <NONTABLE>, reduce the tolerance.
- When the tolerance gets 0, the table ends.

See section 4.3 for a discussion of experiments using this new algorithm.

4.3 Performance on Answer Retrieval

The improvements discussed in section 4.2 were evaluated in QA experiments. We test these improvements in two increments. The first step tested the new CRF extraction trained with more features and data; the second step added the algorithm that is more tolerant of non-table lines in the headers. Table 7 contains the results of

these experiments, along with the heuristic results and the original CRF results for comparison.

Table 7. MRR for QA retrieval

	MRR at	Heuristic	Original CRF	Step 1	Final
No stemming/ stopping	1	0.14	0.14	0.16	0.2
	5	0.171	0.178	0.187	0.24
	100	0.187	0.194	0.198	0.256
With stemming/ stopping	1	0.14	0.1	0.16	0.2
	5	0.168	0.149	0.193	0.248
	100	0.187	0.171	0.210	0.265

Table 8 shows the improvements in MRR in percentage terms. Results are shown for both the improvement over the heuristic extraction method and extraction using the original CRF.

Table 8. Percentage Improvements

	MRR at	Vs. Heuristic		Vs. Original CRF	
		Step 1	Final	Step 1	Final
No stemming/ stopping	1	14%	43%	14%	43%
	5	9%	40%	5%	35%
	100	6%	37%	2%	32%
With stemming/ stopping	1	14%	43%	60%	100%
	5	15%	48%	30%	66%
	100	12%	42%	23%	55%

5. ANSWER RETRIEVAL FROM WALL STREET JOURNAL TABLES

Section 4 showed improved CRF extraction for answer retrieval on FedStats data. However, after the extensive use of FedStats to develop the CRF and extraction algorithms, an unbiased dataset was needed to confirm these improvements. Wall Street Journal data, from TREC volumes one and two also contains many text tables, although in simpler forms than FedStats. It was decided to test the system on this dataset as well.

5.1 Table Extraction

Table 9 shows the results of labeling experiments using the heuristic and CRF. The improved CRF, which applies the new features described in section 4.2.1, does much better than heuristic and original CRF.

Table 9. CRF table extraction performance from WSJ data

	Line Accuracy		Precision	Recall
Heuristic	85.3%	Table lines	58.6%	69.0%
		Header lines	14.8%	61.4%
Original CRF	95.6%	Table lines	97.8%	79.1%
		Header lines	45.8%	73.2%
Improved CRF	97.8%	Table lines	98.3%	91.3%
		Header lines	74.7%	82.9%

5.2 QA Retrieval

The results for answer retrieval from WSJ data are in Table 10. The final results are the ones after the same 2-step improvements as for FedStats data except the training set was not enlarged.

Table 10. MRR for QA retrieval from WSJ data

	MRR at	Heuristic	Original CRF	Final
No stemming/ stopping	1	0.145	0.301	0.377
	5	0.211	0.391	0.438
	100	0.219	0.400	0.449
With stemming/ stopping	1	0.127	0.245	0.340
	5	0.188	0.326	0.391
	100	0.200	0.335	0.401

The QA retrieval results from the simpler WSJ tables extracted by CRFs are also improved considerably by the same ideas that worked for the more complex tables of FedStats. In addition, the results from WSJ tables are much better than FedStats tables. The WSJ tables are editorially different than the FedStats tables. Titles in these tables are often not included, or are formatted in such a way that they appear as the last paragraph of a story. This question-answering system with CRF table extractor works considerably better on these simple tables because the extraction quality is higher.

6. DISCUSSION

In this paper we attempted to improve the performance of question answering on table data with the CRF extraction model. The CRF labeler shows excellent line accuracy, but this does not always translate to high effectiveness in retrieving answers. An analysis of the errors resulted in a number of improvements that directly impacted retrieval performance.

6.1 CRF vs. Heuristic

Initially, the QA retrieval results of the heuristic extraction were better than extraction via CRF. Although the CRF has better accuracy labeling lines, it did not bring in as many headers as the heuristic. But the cross-validation experiments show that with more training data, CRF performance on header lines is improved. After incorporating various improvements, the cell documents generated by the CRF extractor were superior in answering questions.

The experiments with the WSJ dataset indicate that the heuristic approach is stable across many types of tables. However, header line precision is poor, and the heuristics do not finely distinguish between types of header rows. These limit its overall performance on QA. On the contrary, the CRF is more adaptable. Its QA retrieval results were improved considerably by the methods we applied in these experiments. In particular, increasing the amount of training data had a positive effect on header labeling, and the inclusion of more varied features also lead to increased performance.

6.2 QA Retrieval Error Analysis

Table extraction and answer retrieval are the two main parts of the QA system. Errors in QA can be assigned to one of these two parts. A question may fail to retrieve a document containing the answer, or a document with the correct answer maybe ranked very low by the answer retrieval algorithm. By classifying and examining the errors, further improvements can be designed.

Table extraction is the foundation of QA retrieval. Better table extraction will generate better QA retrieval results. The line accuracy, precision and recall of table lines and header lines are all very important, especially header line recall, which shows a dominant role in the experiments. The question, “How many

thousands of pounds of carpet class fibers were consumed in spinning in 1991?" shows how the answer was missed by poor extraction. In the table containing the answer, the keywords for this query, including "fibers", "1991", "carpet class" and "thousands of pounds" are in the headers of this table. But the line-labeling program mislabeled all the header lines. After extraction, all the column headers and section headers are missed in the metadata document of the answer cell. Only a "carpet class' was extracted with this answer cell number as a row header. This metadata document was not ranked in the top 100 retrieved documents, and then the answer was totally missed. This points out the need to continue working on improving CRF line labeling, especially the recall of header lines to solve such problems.

Answer retrieval is also important to the system performance. The question, "What was the bearing acreage for tart cherries in 1995?" is an example of how a cell document was poorly ranked by the answer retrieval algorithm. The cell was correctly extracted from its table. There is, however, more than one table on this question topic in the dataset. Some of these tables present the bearing acreage for tart cherries for US states. The table containing the answer also has the numbers listed with state numbers, but has a total line at the end. The tables are extracted correctly, but from the questions and the current QA retrieval methods we fail to locate the answer due to the lack of the query term "total" in the question. The cell documents for the other cells that are in similar tables contain all the keywords of this question, but the correct document contains the word 'Total' instead of a state name. Since an important query term is missing, the correct document is given a low rank on this question.

Other improvements require increased synergies between table extraction and answer retrieval algorithms. The table containing the answer to the question, "What was the value of inventories in June 1995?" sheds light on this. The title of this table is long and content rich. The title contains a detailed keyword list for the whole table and also has a time stamp, "January 1992 through March 2001", which may confuse the document retrieval component of the QA system. Also, there are many other tables that have titles with the same rich content, and may even match query terms better than the real answer cell. Although the answer cell has a "June" as the column header and a "1995" as the row header, the "June" and "1995" in the metadata document have the same weight as the time words in the title, like "January" "1992" "March" "2001" in this example. An answer retrieval algorithm that puts more weight on row and column metadata, for example, might be the type of synergy which would lead to a higher ranking of the correct answer on this type of question.

7. CONCLUSIONS AND FUTURE WORK

Question answering from table data remains a difficult task. The heuristic extractor is consistent across databases but performance is low. The CRF extractor performs better, but it is sensitive to the data and works considerably better on simple tables.

Future work will concentrate on raising the recall of table header lines and improving the models for answer retrieval. It is likely that our training set is still too small and not diverse enough to produce a generic table extractor. Including WSJ documents along with the FedStats documents would be a good starting point for expanding the training data.

Smoothing could be a source of improvement for the retrieval models. One method would be to incorporate information from the whole table, or nearby paragraphs into the language model for the cells. Another proposal would build many different models for the cell. In addition to the cell document we now create, various models for the text around the cell would also be generated, e.g. the text above the cell. In this way, metadata missed by the original extraction may be captured in one of the other models.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the National Science Foundation under NSF grant #EIA-9983215. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

9. REFERENCES

[1] Berger, A., and Lafferty J. Information Retrieval as statistical translation. In *Proceedings of ACM SIGIR 1999*, 222-229, 1999.

[2] Hurst, M. *The Interpretation of Tables in Texts.* PhD thesis, University of Edinburgh, School of Cognitive Science, Informatics, 2000.

[3] Hurst, M. Layout and language: An efficient algorithm for text block detection based on spatial and linguistic evidence. In *Proc. Document Recognition and Retrieval VIII,* 56–67, 2001.

[4] Hurst, M. Layout and language: An efficient algorithm for text block detection based on spatial and linguistic evidence. In *Proceedings of the 18th International Conference on Computational Linguistics. ICCL,* July 2000.

[5] Hurst, M. and Nasukawa, T. Layout and language: Integrating spatial and linguistic knowledge for layout understanding tasks. In *Proceeding of the 18th International Conference on Computational Linguistics. (COLING 2000),* 2000.

[6] Krovetz, R. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, 191-202. ACM Press, 1993.

[7] Lafferty, J., McCallum, A. and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML,* 2001.

[8] McCallum, A. Mallet: A machine learning for language toolkit.

http://mallet.cs.umass.edu/

[9] Ogilvie, P. and Callan, J. Experiments using the Lemur toolkit. In *Proceedings of the 2001 Text REtrieval Conference (TREC 2001),* 103-108, National Institute of Standards and Technology, special publication 500-250, 2002.

http://www-2.cs.cmu.edu/~lemur/

[10] Pinto, D., Croft, W. B., Branstein, M., Coleman, R., King, M., Li, W. and Wei, X. Quasm: A system for question answering using semi-structured data. In *Proceedings of the*

JCDL 2002 Joint Conference on Digital Libraries, 46–55, 2002.

[11] Pinto, D., McCallum, A., Wei, X. and Croft, W. B., Table Extraction Using Conditional Random Fields, In *Proceedings of SIGIR '03 Conference*, 235-242, 2003.

[12] Ponte, J. and Croft, W.B., A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR 1998*, 275-281, 1998.

[13] Pyreddy, P. and Croft, W. B. Tintin: A system for retrieval in text tables. In *Proceedings of the Second International Conference on Digital Libraries*, 193–200, 1997.

[14] Voorhees, E. Overview of the Trec-9 question answering track. In Proceedings of the Ninth Text Retrieval Conference(TREC-9), 2000.

[15] Zhai, C. and Lafferty, J., A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of ACM SIGIR 2001, 334-342, 2001.