CSE 705 - SEMINAR REPORT
WEBTABLES: EXPLORING THE POWER OF TABLES ON THE WEB

Kevin Quadros
kquadros@buffalo.edu

OVERVIEW

The Internet is a vast collection of unstructured data. The English language crawl indexes of Google were used as a starting point for all analysis. This data set was sent through an HTML parser to detect pure HTML tables. The output was of the magnitude of about 14.1 billion web tables. These tables contained both tables used for purely relational data as well as those used for other non-relational purposes like layout, calendars, and HTML forms and so on. We are only concerned with the about 154M tables used to depict relational data. Each such table contains a header row that contains the attributes. We further consider every such table to be a relational database containing a single table with the schema defined by the attribute labels.

The actual extraction process is much more involved. We make use of machine learning classifiers that are trained using supervised learning. We use two such classifiers, one that would rate a table's relational quality while the other would detect the presence of the attribute labels that are used as metadata. The relational filter is set for high recall and lower precision. The performance of the metadata detector can be improved by making use of statistics from the attribute correlation statistics database.

This dataset is over 5 times larger than any other dataset used in previous research. Using this corpus, we look to solve the fundamental problem of making search of such structured data more efficient at a search engine level. We also suggest the creation of a novel attribute correlation statistics database. The ranking of relational data must take into consideration the special features of tables. We present two new algorithms to rank relational tables. The first one is FeatureRank that uses a linear regression estimator over query independent and dependent features. The next algorithm is SchemaRank that uses average point wise mutual information scores over all pairs of attributes to generate the ranking score.

We could use these statistics to create tools that could aid database designers. The applications include a schema auto-completer that would take input some context attributes and predict pairs of attributes that would be relevant to the schema. It can be implemented using a greedy algorithm that would guess the next-most-probable attribute. For weakly coherent context attributes, we suggest a domain based suggestion tool. From the experimental results, we note that the system performs better if allowed to make up to 3 tries. The next application is an attribute synonym finder. It calculates degree of synonymy using a function that considers various facts of synonymous attributes like they occur in similar contexts, but never in the same schema. In the results, we found that for smaller values of k, the accuracy of the system is good. The final application is a join graph traversal system. It makes use of a neighborSim method. The

algorithm takes as input a context and draws links to clusters of schemas for each attribute. The clusters are tightly coherent and all share the common attribute. The experimental results show that the join graph is useful and the clusters are almost always coherent.

DETAILED COMMENTS

1. The strengths of this paper are that it uses a corpus that is 5 times larger than any other corpus previously used. There are around 30 million queries made daily whose results are in this corpus. Thus suggesting a new search methodology for this data set would be highly beneficial. Also the applications suggested by this paper show very good results in the experiment.

2. The weakness of the paper is that this corpus is not readily available to the research community. In the join graph traversal application, the algorithm does not make use of the neighborSim function. The idea proposed for synthesizing attribute names from table data is not backed up with some published work, nor is any algorithm/method suggested. The paper also does not provide run-time performance results of the system.

3. The paper is technically well-balanced. The motivation behind the system is well justified. A good deal of background information is provided. For the various ranking schemes, a progression in terms of level of sophistication is given. The matters pertaining to Machine Learning and Information Extraction are only introduced but dealt with in related papers.

4. The paper is technically sound. It provides a premise, an architecture, algorithms and accuracy values for experimental results. There are large numbers of examples provided which make the paper more coherent.

5. The scale at which the information extraction has been done in this paper is far greater than that of other related work. The work done by Gatterbauer, et al. for extracting tables from an HTML page without parsing it rather using cues like on-screen data placement would be a good table extractor for the WebTables system. The attribute synonym finding tool is novel. Similar synonym finders in other papers make use of metrics like co-occurrence, proximity statistics or machine-learning based corpus-wide linguistics based systems. None of them use an ACSDb-like system that employs probabilistic methods to detect synonyms.

6. Some of the discussions were on the machine-learning based classifiers. The purpose for inclusion of table features with the training set was discussed in terms of supervised learning. Questions were also raised about the various parts of the relational search interface. Students commented on the feasibility of human judges in creating the various training sets used for supervised learning and for comparison of performance of the system.