

Critical Comparison

1 Introduction

“Mining Data Records” (Liu, Grossmann, Zhai), or MDR, and “Automatic Data Extraction from Lists and Tables in Web Sources” (Lerman, Knoblock, Minton), or ADE, both present algorithms for extracting data from web pages.

2 Feature comparison

MDR focuses on pages whose structure is determined solely by form- and table-related HTML tags. There is no learning step required before the algorithm is able to mine a page, since the system is based on a constant set of string comparisons.

In contrast to this, ADE's goal is to build a wrapper for a set of similar structured pages which can then be used to process any number of pages from the same source without requiring further structure analysis. This makes it more adaptable to different kinds of pages than a completely static system like MDR.

A consequence of this fundamental difference is that MDR extracts data from all table-based structures on a page, whereas ADE recognises the regions which contain the actual data by comparing the set of example pages and mining only those sections which differ on these pages. This easily leads to MDR results containing superfluous information, like static content of a page which only use table-related tags for design purposes.

On the other hand, MDR correctly handles non-contiguous data records, like continuous information spread over multiple rows and/or columns, and even recognises interleaved records (e.g. spreadsheets). ADE does not correctly deal with such pages.

ADE's two-step approach should also lead to quicker execution times once the wrapper is built, since the actual data extraction step does not have to analyse the page structure any more. This might become relevant when data has to be extracted from large sets of similar pages. However, unlike MDR, the ADE paper does not provide any information about operating times. In fact, although the paper lists experiment results, there is no information about the chosen implementation at all.

3 Technical details

Critical parts of the ADE approach are directly or indirectly based on previously existing algorithms (DataPro, AutoClass and ALERGIA). Most of these algorithms are only superficially described in the paper and require closer examination if the

ADE system is to be implemented. MDR, on the other hand, is almost a standalone system, only using an existing string matching algorithm (edit distance string-matching algorithm).

4 Example cases

Since we could find an existing implementation only for the MDR algorithm, it was not possible to directly compare actual results. Instead, the examples presented here are in part based on results from the MDR program and in part theoretical conclusions based on the algorithm descriptions in the papers.

Our first example page was a list of products from the Geizhals website¹. MDR correctly identified all data regions and their data records and extracted their contents. This included not only the product list but also static page regions like menu bars etc. For the same page, ADE would discard these static regions and instead only extract the product list information, even further splitting the record description (e.g. “Intel Pentium-M 1.1GHz(Centrino) * 10.4" TFT (1024x768) * 512MB DDR * 40GB * ext. DVD/CDRW * ...”) into separate columns, recognising the asterisk character as a column separator. MDR extracted the entire description as one single cell.

For our next example, we chose the popular web shop Amazon². The existing MDR implementation, for no apparent reason, failed to produce any data output for a product search result page. According to the MDR paper, it should correctly handle two-level tables like the one in this case. ADE, as far as we can tell, would successfully locate the product list on the page, split its contents into relevant columns (title, author, price etc.) and then extract all information from them.

A third example page, the laptop product comparison on Dell's website³, contains a column-based table layout and therefore was an ideal example to test MDR's capability of handling non-contiguous data records. Yet again, the example implementation did not meet our expectations: No data was extracted from this page. The ADE system, while certainly able to correctly locate the data region on this page, would totally fail to identify the relations between the table cells, due to the fact that it cannot handle non-contiguous data layouts.

5 Conclusion

The fact that we were able to find a demo implementation for only one of our two

1 <http://www.geizhals.at>

2 <http://www.amazon.at>

3 <http://www.dell.at>

systems, which did not seem to completely fulfil MDR's requirements, made a thorough comparison rather difficult. Our results therefore mostly represent our own theories of how these two systems should work. Nevertheless, it seems obvious that ADE's strength lies in detecting structures even on pages not featuring an explicit formal structure, whereas MDR excels in handling non-contiguous layouts. For pages using only completely formal, table-based HTML structures for presenting their data, both approaches should work correctly. Still, even in this case ADE would spend a certain amount of computation time learning the page structure and building a wrapper, whereas MDR simply takes the straightforward approach of directly extracting the data.

Assuming the web will never be “clean” (presenting its data in a completely formalised manner), MDR is useless if a general system for arbitrary pages is required.