

Comparing Record Boundary- and Automatic Segmentation Algorithms

Both papers represent an unsupervised approach to the record segmentation problem. The techniques do neither rely on training examples nor user inputs and can work automatically.

The one striking difference starts with the input to both algorithms:

While the Record Boundary Algorithm works on **plain lists** in single html-documents and analyses the **tag sequence** of the page, the Automatic Segmentation Algorithm doesn't. It needs a **list and a details-view** and uses structures in content to do the segmentation. The authors state that most parts of the hidden web are of list/detail - structure, which we both doubt. On the other hand: It's hidden, so how should we know?

The Automatic Segmentation Algorithm is very nice for crawlers. It tries to work with html-forms to receive its input. The Record Boundary Algorithm on the other hand isn't that intelligent. It needs the ready-served list-page to proceed.

Also note, that the Automatic Segmentation Algorithm works on multiple pages, and "subtracts" them from each other to get a **page-template** in return. The page-template can be safely ignored. The Record Boundary Algorithm ignores the template completely. Templated stuff would drop out of the heuristics – if there are enough records – anyway.

The clear structure of the Record Boundary Algorithm and the possibility to easily add or change heuristics can be seen as an advantage compared to the adjustment of the HMM, which is rather sophisticated.

On the other Hand, the HMM is the only approach that allows assumptions on the record attributes (**columns**) and the database that exists in the background. Another advantage of the HMM is its tolerance for inconsistencies in content.

We think that the presented algorithms can coexist well. Their goals hardly overlap, and although their domain-independency is stated in the papers, their successful application depends on the kind and structure of the sites they are used on.

Comparison of the Segmentation Algorithms:

Algorithm	<i>un-supervised</i>	<i>domain-independent</i>	<i>needs list-and detail-pages</i>	<i>infers Columns</i>	<i>uses tag-tree</i>	<i>allows inconsistencies in content</i>	<i>uses template finding</i>
5 Heuristics (#5)	X	1,2,3,4			X		
CSP (#6)	X	X	X				X
HMM (#6)	X	X	X	X		X	X