TikaEvalAndStructuralComponents

Evaluating Structural Components in Extracted Content with the tika-eval Module

NOTE: This page assumes basic knowledge of the tika-eval workflow. Please see TikaEval and make sure that you understand how the tika-eval modules works on text before considering structure.

Background

File formats often contain structural or stylistic elements, and Apache Tika attempts to normalize and represent some of these features in its XHTML output. As of Tika 1.20, users can get counts of common XHTML tags (in Profile mode) and/or comparison counts of common XHTML tags (in Compare mode). Users can also count "tag exceptions" – cases where the structure tags violate XML/XHTML requirements, e.g. `<b><i></b></i>`.

Known Limitations

1) Simply counting structure tags offers only a rudimentary insight into the structure of a single extract or as a comparison between two extracts of the same source file. One might want to apply a more advanced tree-based similarity/distance metric between two extracts – our JIRA is open and committers are standing by.

2) If one one tool's extracts have more <p> elements than do another tool's that doesn't necessarily tell you that one extract is better than another.
For example, one tool – Tool A – might add <p> elements for every new line in a PDF:

```
<p>The quick brown fox</p>

<p>jumped over the lazy dog</p>
```

Another tool – Tool B – might apply heuristics to reconstruct logical paragraphs, such as
`<p>The quick brown fox jumped over the lazy dog. </p>`

3) The tika-eval module is currently using TagSoup on *.html files. TagSoup is designed to silently fix structural problems in the HTML so no exceptions will likely be thrown for non-compliant HTML. We may alter this behavior in the future.

Tool A would have more <p> tags, but Tool B is probably capturing better information about the structure of the document.

Intended Uses and Scope

While there are obvious limitations, the tag capability will allow users to identify:

- If one tool is completely skipping or capturing fewer structural elements. One could imagine a regression in Tika that dropped footnotes in DOCX, and this would quickly show that there were fewer `<div>` elements in the output from the more recent version; or one could quickly see that one parser is extracting URLs (`<a>`) elements from PDFs, but another isn't.
- Structural problems in Tika's XHTML output – there is an XML exception flag in the database if an error was thrown parsing Tika's output

How to Count Structural Components

If you are using Tika to generate .json files, follow the directions on TikaEval for how to create a directory of extracts, **but** don't include the `-t` option: `java -jar tika-app.X.Y.jar -J -i input_dir -o extracts`. This has the effect of storing the content that is extracted as XHTML, and it sets a metadata value of `ToXMLContentHandler` for the key `X-TIKA:content_handler`. When tika-eval finds that value set in the metadata, it parses the XHTML with a SAXParser to count the structure tags and extract the text.

Included Tags

As of Tika 1.21, the tika-eval module includes the following structural components:

- <a>
- <b>
-
- <i>
- <img>
-
-
- <p>
- <table>
- <td>
- <title>
- <tr>

- <u>
-

NOTE: For most parsers, aside from the HTML parser, Tika injects the file's "title" metadata component into the `<title>` tag. For example, if the "title" metadata element in a PDF is "Tika 2.0 – NextGen Extraction", that will be injected into the XHTML as both a metadata item `dc:title`, and in the content within `<title>` tags. The HTML parser passes through the source HTML page's `<title>` contents.

Given how much Tika relies on `<div>` tags, we may at some point want to include the `class` information so that we can tell instantly if there's a change in e.g. `<div class="slide-notes">` elements.

For now, we've chosen not to include the `<h*>` tags because, while the HTML parser passes those through, the other parsers do not tend to generate header tags.

Reports

While we added structural component extraction in Tika 1.20, we only added report writing in 1.21-SNAPSHOT. Users of Tika 1.20 can use the H2 database directly for reporting, generate their own reports via a custom XML reporting config file or use the updated report configuration files from 1.21-SNAPSHOT available: for profile mode and for comparison mode.

Profile Mode

As of Tika 1.21-SNAPSHOT, there are reports for:

- Tags by mime – each row is a mime type, and there's a single column for the sum of the tags of each tag type
- Tag exceptions by mime – each row is a mime type, and there's a count for the number of XHTML extract files that triggered a parse exception in tika-eval
- Tag exception details – this lists every XHTML extract file that triggered a parse exception in tika-eval

Comparison Mode

As of Tika 1.21-SNAPSHOT, there are reports for:

- Tag comparisons by mime – each row is a mime type pair (mime type as identified by tool A and mime type as identified by tool B), and there are pairs of columns for each tag, e.g. `tags_div_a` contains the sum of the `div` tags in extracts from tool A and `tags_div_b` contains the sum of the div tags in extracts from tool B.
- Tag exceptions by mime pair – each row is a mime type pair (`mime_type_a` and `mime_type_b`) and there's a count for the number of XHTML extract files that triggered a parse exception in tika-eval
- Tag exception details A – this lists every XHTML extract file from tool A that triggered a parse exception in tika-eval
- Tag exception details B – this lists every XHTML extract file from tool B that triggered a parse exception in tika-eval