Kai Pak
November 15, 2020
Fall 2020 CS410 – Text Information Systems
Technology Review

Tika: A Powerful Content Detection and Extraction Platform with Text Analysis Tools

Consider the show Star Trek where humanity has made contact with millions of intelligent species across the universe. Despite the thousands if not millions of languages and dialects each of these species must in turn have on their home worlds, it is rather astounding that the characters on the show are easily able to communicate with each thanks to a technically convenient explanation called 'The Universal Translator'. Amazingly, this technology seamlessly—and in real-time—translates everyone's spoken word into a common tongue.

Text data back here on Earth shares a similar, if not, somewhat less daunting problem of many differing data formats and standards that do not have a common method of formatting, tagging, compression, metadata, and extraction methods. This is essentially the job of the Apache Tika framework. It is a powerful framework used to extract data from the thousands of different data format types through a common API. It also includes tools for text analysis and data mining tasks. While the native applications themselves of course know how to extract this data, this is not a scalable solution that can work against the volume and diversity of the Internet or even much simpler corporate networks.

Like in many other data science areas of specialization, text analysis and natural language processing (NLP) literature often focuses on the elaborate machine learning models used to achieve useful incredible results. However, as is commonly the case, often the most challenging parts of these text data focused projects lies with consistent and robust data extraction and evaluation. As any experienced machine learning practitioner knows, successful projects arise from well-designed and implemented data extraction pipelines.

In the case of NLP, this is made much more challenging by the numerous differing types of file formats that exist on the Internet: by far, the largest corpus of human text in existence. In addition to the ubiquity of markup (Markdown, HTML, JSON, YAML, etc.) languages and data formats, there are common binary formats such as Microsoft Word and its variants, PDFs, and others. Add to the mix other formats that can contain text along with images and videos (PowerPoint, Keynote, Excel, etc.), plus the numerous versions and dialects they each have and it quickly becomes clear that creating individual extractors can become an overwhelming task for any enterprise scale text processing project.

Architecture Overview[1]

Tika is designed around a consistent framework that relies on several key components. For identification, it relies on the MIME (Multipurpose Internet Mail Extensions) standard so it

[1] Zitting, Jukka and Mattmann, Chriss. *Tika in Action.* Manning Publications, 2011.

can determine what type of file it is dealing with. The details of MIME detection is beyond the scope of this paper, however, Tika is able to use MIME detection patterns to identify the data type and relevant associated patterns to extract the data. Then, there is the parser component which is the part of Tika that does the actual text and metadata extraction. Finally, there is the language identifier portion which detects the language that the stream of input data is composed in. The Tika framework is organized around a Tika façade class that acts as the frontend API to all of Tika's capabilities. Around this class are the libraries that run the core competencies. Although this is a relatively simple set of repositories, its design expresses elegance and belies the complex logic and tools it uses to do its task.

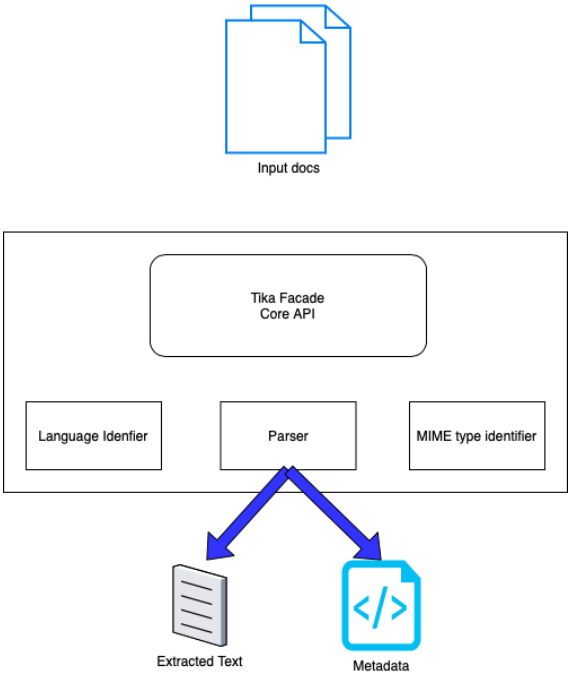

Figure 1: Hi level Tika architecture.

Tika is primarily written in Java although APIs exist in Python, R and other languages commonly used in NLP applications. In Python, the API launches a Java VM which allows REST calls to a Tika Server. Extracting metadata and text from a document could not be more trivial. Here's Tika pulling metadata from this document I am currently writing.

```
[2]: import tika
     from tika import parser

[5]: tika.initVM()

     parsed = parser.from_file('/Users/kai/Documents/Documents - zen/education/UIUC/cs410/technology_review.docx')
     parsed['metadata']

[5]: {'Application-Name': 'Microsoft Office Word',
      'Application-Version': '16.0000',
      'Author': 'Pak, Kai Young',
      'Character Count': '3032',
      'Character-Count-With-Spaces': '3556',
```

Conclusion

The task of text data collection and extraction can quickly overwhelm NLP or text data mining projects. Tika provides a powerful framework to simplify this core need.