Newsblaster Article Text and Title Extraction

Michael Wojcieszek

Abstract

Although the Internet is slowly evolving into a more interconnected, semantic web, a significant portion of the online content today still exists in unstructured documents, making the task of extracting informative textual content a challenge. My motivation mainly resides in boosting the performance of article extraction in Newsblaster but the implications have far-reaching consequences in other applications of information retrieval, news aggregation and summarization, event and entity tracking, and even financial trading based on new sentiment analysis.

Introduction

The main challenge in text extraction lies in distinguishing parts of an HTML document which represent article text apart from other non-informative web components such as navigation panels, menus, sidebars, advertisements and more. It requires parsing of the HTML document into a DOM (Document Object Model) Tree and searching through the nodes of the tree for the relevant news content. Furthermore, it must reliably distinguish these nodes from other non-informative textual noise.

This task is further complicated by the dynamic nature of the web. With the rise of the Social Web in the last decade, websites have evolved from static pages to more dynamic, interactive viewing experiences. The widespread success of online advertising has further cluttered this space with all forms of tracking software to make the online experience more personalized. This has resulted in article text nodes being embedded in a sea of irrelevant information. To add to the complexity, the Internet is still young and, as such, definitive standards for markup language practices have yet to experience widespread use.

Related work

Extracting information from semi-structured documents has been well-studied and several algorithms have been proposed to accomplish this task. By far the most successful, boilerplate detection computes both textual and structural features on parts of a document and leverages machine learning techniques to build a classifier. The set of features includes text frequency in the entire corpus, indicators for the presence of particular tags that enclose a block of text, shallow text features, statistics about the local context of text, and other heuristic features. The approach explores using decision trees along with SVM classifiers trained on a set of labeled documents.

Newsblaster

The Article Extractor module of Newsblaster consists of a series of heuristics that rely on punctuation frequencies and ratios to make a decision. Surprisingly, the comma ratio produces pretty good results for predicting whether a particular node contains news content or not.

However, it is completely ineffective in extracting titles from the bodies of HTML documents. By standard practice, the article title should be correctly defined in the title field (marked by '<title>' and '</title>' tags) by authors, but according to the CleanEval challenge, about 33.5% of title fields in news pages contain junk.

My approach

I propose a novel solution that focuses on first identifying a bounded rectangular region around the relevant textual content of a news web page that includes article text, the title, corresponding images and captions, details about authorship and time/date of creation. The task then reduces to finding a minimum subtree of the DOM Tree that contains all relevant information associated with a news article. This is marked by the dashed rectangle in the figure below.

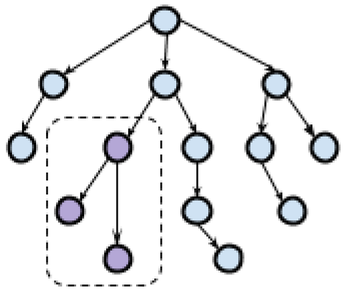

Figure 1: DOM Tree of HTML Document

Once the simplified tree has been determined, the task shifts to pruning the nodes that are noisy until we are left with a skeleton that represents the core elements of the news article. This can be done by computing features for each of the nodes and iteratively stripping nodes that do not pass a set of tests or weak classifiers, which become stricter with each iteration. The final product is a thinned tree where each of the nodes has reasonable probability of containing some sort of relevant textual content.

The set of features useful for pruning:

- textual frequency in the entire corpus to obtain a vocabulary of words used in useless parts of the document
- punctuation frequency and ratios
- information contained in style sheets corresponding to the format, layout or style of a text node
- information contained in the attribute fields of tags that might hint at the type of content present in the node
- local context and relative position of text
- heuristic features

The final step would involve matching nodes to their respective categories (i.e. article text, title, image caption, author, date/time, etc.).

Newsblaster Article Text and Title Extraction

Michael Wojcieszek

Progress-to-date

So far I have devised an algorithm that, given a raw HTML document, parses the document into a DOM Tree, traverses the tree to identify nodes of potential value, and using a set of heuristics, it analyzes statistics about the various paths within the tree and selects a node that lies within the most likely path and represents the minimal subtree containing the article content. Then it saves various versions of the HTML document, each containing an injected JavaScript script that highlights an unique piece of text within the document that corresponds to a node in that subtree. The process is illustrated in the figure below.

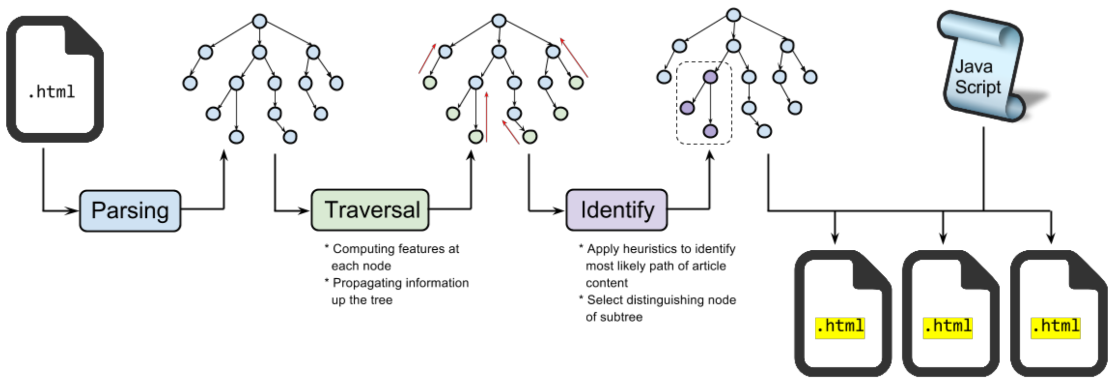
.html
Parsing
Traversal
* Computing features at each node
* Propagating information up the tree
Identify
* Apply heuristics to identify most likely path of article content
* Select distinguishing node of subtree
Java Script
.html
.html
.html

Figure 2: Article Extractor 2.0 Pipeline

Future work

These various versions of the HTML document will then be pipelined into a crowd-sourced service like Mechanical Turk or CrowdFlower, where human review will determine whether the highlighted text in the document is relevant article content or noise. Such a service will provide the necessary annotated corpus for me to test out my hypothesis for article text and title extraction.