At A Glance News

Greg Garrett

Justin Ratcliffe

Introduction

Our goal for the *At A Glance* project was to create an on-line service that provides users with summaries of articles from the news source of their choosing. As daily consumers of news, we knew that little was currently being offered for customization of news detail level. RSS feed aggregators allow readers to get news from multiple sources and customize by particular topic. Yet, time constraints and the prodigious amount of content mean that few have time to read all the news they would like. Some publishers have approached consumers' expeditious desires but have not put control into the hands of the public. One example is *NPR*: they provides five minute news casts at the top of every hour, summarizing five of the latest top stories. Another is *Newser* [1] which provides summaries of news articles averaging around one hundred words each; these articles are written by staff writers and the content is typically resemblant of tabloids. We want readers to get the news they want, from the source they want, at the level of detail they want.

A key motivation behind our project is that we wish to provide journalistic value by making the significant interesting and relevant. By summarizing articles from the news source of the user's choosing, we hope to alleviate the information overload that may cause them to lose interest in a story. Whereas beforehand the reader may have gotten bogged down in the details of the story introduction, with *At A Glance* they will be able to read a brief summary, allowing them to consume the salient points of the story first, at which point they can decide if they want more detail. In this way, users can at least superficially engage with a story, giving them a base knowledge of the latest news.

In our preparatory research for the project, we analyzed *NewsInEssence*, an interactive, multi-source news summarization system developed by the University of Michigan [2]. We found that while *NewsInEssence* shared the same general goal as *At A Glance*, that of providing readers with news summaries, their method was quite different from our own. They aimed at taking articles from multiple news sources and grouping them by topic into clusters. These clusters would then be used to create summary of the news item, drawing the most relevant sentences from the different articles.

At A Glance takes a simpler approach to generating news summaries. Instead of drawing articles from multiple new sources, we focus on just one. At regular five minute intervals, our system will download the latest articles from a news source's RSS feed. Important metadata about each article is extracted from the XML-structured RSS feed, such as title, publication date, URL for the full article, and the article text. Once this has been done, summaries of three different lengths are generated for each article: a short summary of 5 sentences; a medium length summary of 10 sentences; and a long summary of 20 sentences. Finally, HTML is generated for these summaries, and links to these files are provided on the *At A Glance* front page.

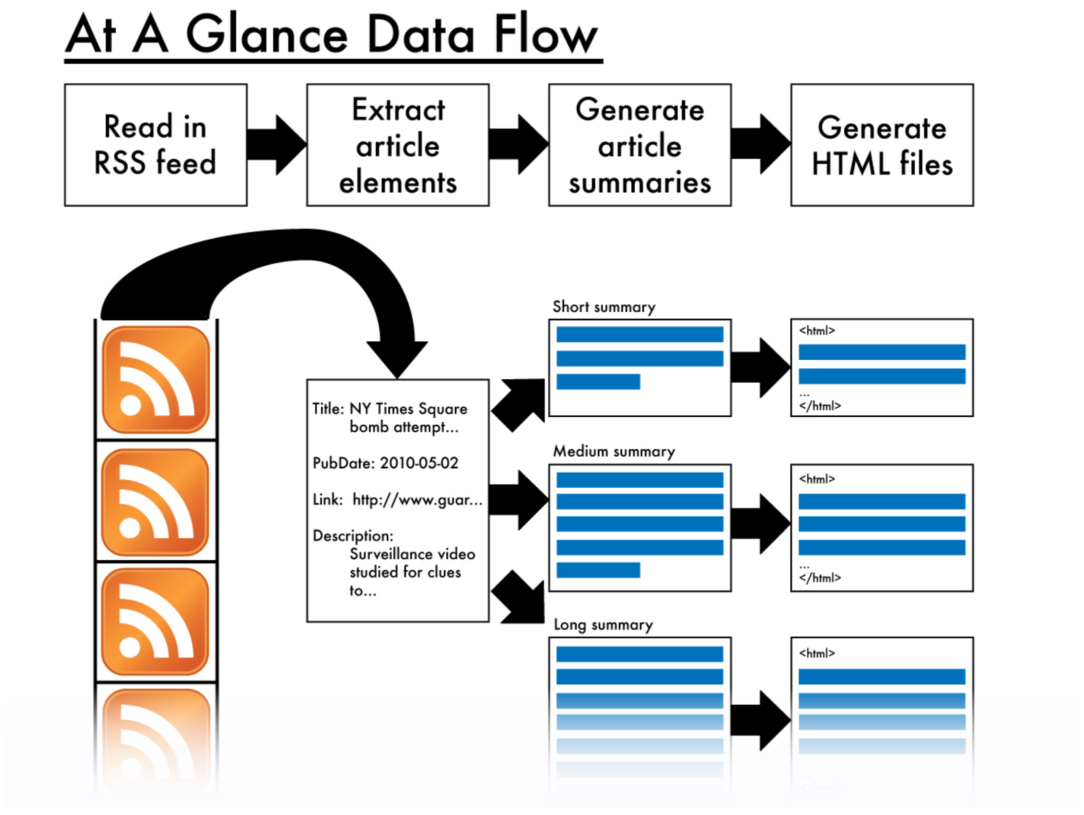

Figure 1: The *At A Glance* data flow

In this current incarnation, *At A Glance* is implemented as a Java service running on a web host. At set intervals (five minutes), it runs the RSS feed reader, which uses the RSS Utilities library [3] to download and parse the XML file from the target RSS feed. Once the relevant article metadata has been extracted, the Classifier4J library [4] is run on the article text to create 5, 20, and 100 sentence long summaries. Then, HTML and CSS are used to generate the necessary web pages, with JavaScript being used to create a more dynamic user experience by overlaying the web pages for the individual article summaries, meaning the user does not have to navigate to a new window or browser tab.

Achievements

As stated, *At A Glance* is a stand alone Java application running on servers at regular intervals. It takes the most recent articles from *The Guardian* and creates summaries at three levels of detail: 5 sentences, 10 sentences, and 20 sentences. It groups each article into its appropriate category to help the reader distinguish topics and shows them the title of the article as well as the hook. This should give the reader enough context to determine the level of detail they wish to have regarding that topic. If they are uninterested in the story, the user may simply move on to another title. If they are intrigued by the hook, they may go on to click the links under it entitled *SHORT*, *MEDIUM*, *LONG*, and *FULL ARTICLE*. The latter link opens a new window with the publisher's page displaying the article. The other links display the respective summary in an overlay with large, serif font making it easy to read. To close the overlay, the user may click the 'X' found in the upper right, click outside the overlay, or hit escape leaving them right where they left off.

What we were unable to implement was a text scraper; this meant we could only include publishers which included their full article text in their RSS feed. This was a large disappointment for us because a major feature of our system was giving the reader the ability to choose their news source. We know how we would implement this, but time constraints prevented us from doing so. Once this is created, we should be able to make quite an extensible product in regard to adding publishers.

Many small tests were performed during development to ensure a reliable product. During testing of the feed, we used the headline RSS feed from *The Guardian* as data. We tested the back end to ensure it reliably produced summaries with as little "junk" content as possible; occasionally, the summarizer has issues with some characters and puts question marks in the middle of sentences. We also performed cross browser testing to make sure the HTML/CSS and Javascript worked correctly on multiple browsers. It is not perfect on all platforms and browsers but it is usable on all.

We were quite pleased with the results of our design and development. We created a stable, and attractive product in only a single iteration of development (granted, there were many small micro-iterations). We get most excited when thinking about what is in store for the next version of *At A Glance*.

Future Work

We recognize that despite how much we were able to accomplish this semester, the *At A Glance* system as it is stands is still a rough prototype. It lacks some of the functionality we originally designed into the system, and certain features need to be improved before our service would be acceptable to consumers. Listed below are the features that will be the focus of future development:

Improve the quality of the article summaries

The Classifier4J library leaves much to be desired when it comes to providing high quality text summaries. While better than some of the other free text summarization services we encountered, we found that rather than creating summaries over the entire article, often it would just summarize the beginning of a story. A specific case involved an article that listed the top ten activities to do in a city. For the short article summary, none of these items were mentioned; even in the medium length summary only the first two were. A number of commercial text summarizing services exist which we could explore to determine if they provide more accurate summaries.

Allow readers to choose their news sources

At A Glance is designed to allow the reader to choose their own news sources, addressing a shortcoming we saw in similar services like NewsInEssence. Due to time constraints we were only able to implement the system so that it read from a fixed RSS feed, that of the Guardian. Future iterations of our system would include this functionality, giving users increased flexibility in choosing the news source from which the articles to summarize are drawn.

Include multimedia

Some news source RSS feeds include links to image files, and we feel that having a preview image for an article on the main page would enhance the user experience. This is a feature we liked about the *Newser* homepage [1]. However, the RSS Utilities library [3] does not provide functionality for extracting these image links, so we were unable to implement this feature. Future iterations of *At A Glance* would investigate how this functionality could be added.

Include related articles

Another feature we feel would enhance the user experience would be the linking of related articles from other news sources. If the user is sufficiently interested in a story, they may wish to see what other news sources are saying as well. By providing such links, we can minimize the amount of searching the user would have to perform, keeping them engaged in the story.

Implement as a browser extension

A possible future direction for our project would be to implement it as a browser extension, allowing users increased flexibility as they obtain article summaries on the fly from whichever news source they are currently visiting. Browser extensions can be quite powerful, as plug-ins like TBUZZ, which gathers the most recent tweets about a story, demonstrate. We feel that *At A Glance* could likewise be successfully implemented as a plug-in.

Demonstration

The demonstration we presented in class was our functional prototype for *At A Glance*. The news articles were pulled from the Guardian RSS feed, and all summaries were pre-generated from the article text. For the demonstration, our prototype was running on an external server [5], but the system can be run locally. To run the demonstration, you will need the latest version of the Java Runtime Environment. Simply execute the makefile, which will automatically build the system, then execute the run script AtAGlance with the following parameters: the URL of the target RSS feed (for the demonstration we used http://feeds.guardian.co.uk/theguardian/rss); the file path for the target main page file including file name (for the demo, index.html); and the file path of the folder in which to write the article summary files (for the demo, summaries/). Open the generated main page file in a web browser and start reading news summaries!

So, if in a shell or Terminal:

```
make <enter>
./AtAGlance http://feeds.guardian.co.uk/theguardian/rss index.html summaries/ <enter>
```

References:

[1] Newser, http://www.newser.com
[2] News In Essence, http://lada.si.umich.edu:8080/clair/nie1/nie.cgi
[3] RSS Utilities, http://java.sun.com/developer/technicalArticles/javaserverpages/rss_utilities/
[4] Classifier4J, http://classifier4j.sourceforge.net/
[5] *At A Glance* Demo, http://www.justinratcliffe.com/glance/