Final Assignment - Social Web

Arthur-Ervin Avramiea
2517642
a.e.avramiea@student.vu.nl

Mihnea Dobrescu-Balaur
2549278
mihnea@linux.com

Zilvinas Kucinskas
2547940
zil.kucinskas@gmail.com

1. INTRODUCTION

With the rise in popularity of the Web, everybody around the world produces content, from local bloggers to global news agencies. This means that the online medium is now full of content, and that is great. The Web embodies diversity and pluralism in opinions, making sure that anybody can find useful content. However, with this much content available, it can be difficult for people to keep track of the items that interest them the most. And because of the decentralised nature of the Web, it can be difficult to find relevant articles.

In order to solve these problems, websites started providing RSS[1] feeds and more advanced users have started following them. There are even online aggregators, like the recently closed Google Reader[2], in which an user could add multiple RSS feeds that he or she is interested in, and then the website will keep track of new articles and display them in managed lists.

The problem with RSS and aggregators like Google Reader is that they are not so user friendly, having a dense interface, similar to a full webmail inbox. Also, they display the articles in a plain way, with lots of text. This all adds up to an overload of information to the user. More recent applications, like Feedly[3] and Flipboard[4], have taken a fresh approach of the problem. They create rich, magazine-like layouts for the articles, and they also help with content discovery, having predefined and curated âĂIJfeedsâĂİ that the users can subscribe to.

We believe that we can take this idea a step further, and create a rich visual timeline, in which the media content (images and videos) take precedence over the text, removing everything but the headline. This timeline helps the users quickly get up to speed with the latest events and news from around the world, and when they find an article that is interesting, they can read it from the original source, with the original layout, in just one click.

Since there is no such thing as one size fits all, our application allows users to select what regions they are interested in, as well as what domains. So, for example, one user might be interested in Politics around the US, while another user might be interested in Tech news from Asia. Personalising a timeline is easy, and a user can store multiple timelines on his or her account.

2. STRUCTURED DATA

Our application has to include articles from all over the world, on all possible topics, and to cover this demand we decided to use, first and foremost, the RSS feeds of the main news agencies in the world. Wikipedia provides a list[5] of them, grouped by country. Besides news agencies, we also include information from prominent newsletters like the New York Times[6] and online publications like The Verge[7].

Besides XML data (via RSS), we also use the Twitter API in order to get JSON data of the tweets related to any given article.

To enrich the userâĂŹs visual experience, besides the media from the original article, we use the Bing search API to find relevant images and videos, that we then later embed in the rendered story.

All the mentioned data sources and others (detailed in the Analysis section) get mixed in a pipeline that builds a JSON object representing the visual summary of the story that we want to render for the user. Then, using

[1] http://en.wikipedia.org/wiki/RSS
[2] http://www.google.com/reader/about/
[3] http://feedly.com/
[4] https://flipboard.com/
[5] http://en.wikipedia.org/wiki/List_of_news_agencies/
[6] http://www.nytimes.com/
[7] http://www.theverge.com

Web technologies we fetch the corresponding JSON files in the frontend application and render them, building the timeline.

3. DATA ANALYSIS

In order to give the user an approximation about the impact of the story they are skimming, we use Sentiment140[8] to perform sentiment analysis on the tweets that we found for that story. Since location is important for our application and for our users, we cluster the sentiment results by region. Figuring out where do the tweet authors live exactly is not trivial, since the majority of tweets does not contain location information. To solve this problem, we rely on the information that users share on their profile page. However, since that data is not structured at all, we have to reason about its text value and decide what region it represents. We do this using the Google Geocoding API[9].

4. REFERENCES

[8] http://www.sentiment140.com/

[9] https://developers.google.com/maps/documentation/geocoding/

5. APPENDIX