LUND INSTITUTE OF TECHNOLOGY
Lund University

MASTER THESIS

3D Visualization of News using Topic Detection and Personalization on Mobile Phones

Author:
Jia Cherng HO

Jia Wei HO

Advisor: Anders Ardö, Lund University
Advisor: Dan Gärdenfors, TAT
Examinor: Mats Cedervall, Lund University

August 19, 2010

Printed in Sweden
E-huset, Lund, 2010

Abstract

With the increasing use of mobile phones, there is an increasing need to access information everywhere. It's almost impossible to keep up to date with every news site and blog. This problem is commonly described as information overload. Our goal is to help users get an overview of his/her news more effectively when using a mobile phone. We investigated different visualizations and studied possible important parameters when browsing news. As a result we built a back-end for generating the necessary parameters that is used in the visualization of our front-end. The system was tested and run in a real environment using a server and a mobile phone that runs the Android OS. The visualization was successfully implemented using a 3D visualization which showed list of topics. These topics represented hot topics that is reported by various news sources. The use of topics helped the user get quicker information of what news to read and he/she would only have to flick through list of topics instead of tons of articles. 3D introduced several advantages for presenting text in a mobile phone such as hiding information and natural interaction.

Acknowledgement

We would like to thank our supervisors Dan and Anders for their patience, guidance, continuous feedbacks, ideas and suggestions throughout the thesis.

We would also like to thank Karl-Anders, Marcus and other employees at TAT, Sandeep, Shailesh and other thesis workers for their help when we encountered difficult problems. Thank you TAT for giving us the opportunity do our thesis in the company.

Lastly we would like to thank our family for their patience and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the increasing use of mobile phones, there is an increasing need to access information everywhere. The computational power and graphical capabilities of the mobile phone is also increasing rapidly which gives new exciting possibilities for presenting information to the user. There exists today a lot of alternatives for reading news. For example print newspaper, online newspaper website, blogs, twitter and RSS readers. One of the popular ones is RSS readers which let users subscribe to feeds. News sites such as Engadget[1] or Gizmodo[2] give users a large variety of news topics related to technology and subscribing to their feeds will give you many articles everyday in your RSS reader. Engadget gets an average of 33.1 objects per day [42] and more if a big event occurs. When you haven't used your RSS reader for one day or two, there will already be a large pile of unread items. And because of that it's not always so easy to keep yourself up to date with the latest topics on the web. If you have the time, going through the news in front of the computer might not be a big problem but what if you're on the go and want to be able to get a quick look at what's happening on the net and if any articles might be saying something that might be of interest for you. In a print newspaper the first page shows some of the biggest news. The second page usually shows articles that is of more interest at the moment for the readers. Existing mobile RSS readers doesn't show this, they only give you a list of articles from each subscribed feed. It happens a lot that a user going through articles in one subscription will read articles which has the same content in another subscription. Using a mobile phone means wanting to get a quick overview of your news. This is also a common problem of information overload. We chose to explore ways to get an insight on a large amount of news and solve the problem of information overload.

[1] http://www.engadget.com/
[2] http://www.gizmodo.com/

1.1 Report Structure

After giving a background to the whole project in section 2, which looks into areas of news reading on the web and information visualization, chapter 4 will show what studies has been made during the course of the project. Things also covered are our approaches to solving our problem and design of our visualization. Chapter 5 shows the result of our master thesis. In chapter 6 different observations are described based on the result. We draw conclusions about the research problem in chapter 7. In chapter 8 shows how our project can be developed further than the current functionalities and concepts applied.

Chapter 2

State of the Art

The World Wide Web gives people access to all kinds of news, whether it is politics, sports, technology or even video games related news. Beyond the bigger news sites, such as Reuters, NY Times, CNN, there are many different blogs that publish posts about various topics. Before we begin our analysis we will explain different ways to read news on the web.

2.1 Really Simple Syndication (RSS)

One way to keep up to date with the news a user is interested in, is to use RSS readers which let him/her subscribe to each news sites' and blogs' RSS feed. This makes news reading easier by collecting all the news to one place. An example of number of posts that is published everyday by certain news sites and blogs [42] is shown in Figure 2.1.

2.1.1 Aggregators

RSS readers are also called aggregators. They are used to collect RSS feeds and let users read them and subscribe to them easily in one application. Examples of popular aggregators are Google Reader[1] and FeedDemon[2]. This way users save time and effort because of not having to visit the websites to see if there are any new updates. Google Reader is an aggregator website. In Google Reader users can add subscriptions and manually assign them groups. The user can therefore select a group and read all articles within that group. There is also a mobile version of Google Reader which basically contains a plain list showing the titles of each item as opposed to its desktop version showing a short text of the content together with title and more in the list.

[1] http://www.google.com/reader
[2] http://www.feeddemon.com/

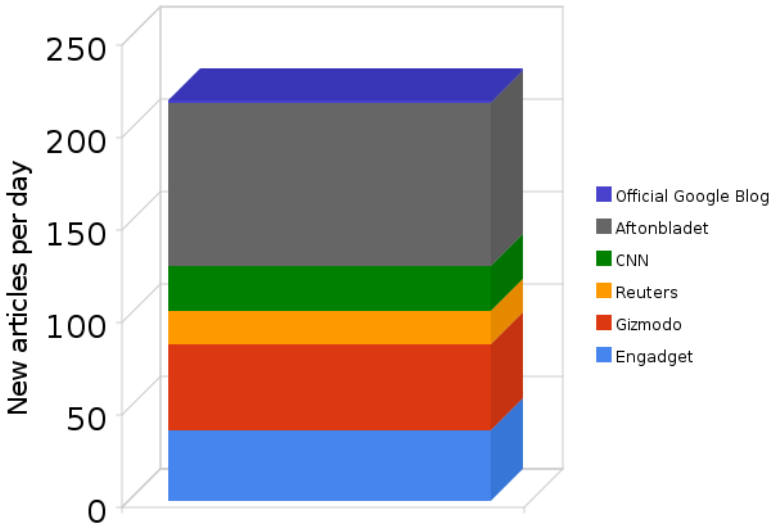

Figure 2.1: An example showing number of news articles for a user in a day using RSS reader to follow news

2.1.2 Format

At the bottom RSS is just an XML file [6]. It is a popular format which is used in blogs and news sites. Another name for RSS is feed, web feed or channel. It enables users to get incremental updates from blogs and news sites without visiting their homepage. The content of the text is most often a summary of the latest content with associated meta tags such as publication date, title, categories etc.

ATOM & RSS

There are mainly two formats of RSS which are used. You can see an example of both in section A.1. The interesting information in which we try to parse come from the following tags:

2.2 Other Alternatives

There are many existing solutions that collect and analyze news from various sources and show them to users without the need for a user to subscribe to feeds. Some of them are presented below.

RSS	ATOM	Description
<item>	<entry>	Surrounds all tags of an article
<link>	<link>	URL from an article
<description>	<content> & <summary>	Full content or a summary of an article
<pubdate>	<published>	Publication date
<category>	<category>	Manually assigned category
<feed>	<feed>	Name of a feed e.g. Engadget

Table 2.1: Comparison of RSS and ATOM

2.2.1 Scintilla

Scintilla [47] is a website which shows the top stories of science subjects within a day and personalized recommendations of articles. It uses data collected from hundreds of news sites, blogs, journals and databases. It's a way for users to sign up and get updates through the site. What is also interesting with this site is how it calculates topics that are hot topics. The site uses topic burst detection. It detects increasing size of clusters of documents that is interesting for the users. The topic burst detection uses a simplified version of Kleinberg's Burst Algorithm. Kleinberg's Burst Detection Algorithm [14] detects bursts in a document stream. The algorithm relies on the fact that certain relevant documents come more often when an appearance of a certain event begins to emerge in the document stream. This will trigger a new state called burst state as opposed to normal state. All the incoming documents are then in burst state.

2.2.2 Techmeme

The website [53] shows so called top stories of technology across different websites and blogs. It contains a list of top topics which are being written by multiple sources and the story put at the top should represent the most written story. Its goal is to collect the scattered news about the same story together without the user having to go through all the articles in different websites. Techmeme uses an algorithm for detecting the topics with human editorial input.

2.2.3 Google News

Google News [28] organises world news from various news sources in real time. All the news are categorized into World, Regional, Business, Sci/Tech, Entertainment, Sports and Health. News reporting about similar topics are

clustered together and presented in order to give the user a "a bird's eye view of what's being reported on virtually any topic" [3]. The first page shows the headlines from all the categories. It offers recommendation of articles based on the user's read history, regional editions for reading news related to a user's country, and specifying search terms in order to get news related to specific subjects. Google News is presented using only computer algorithms without any human intervention. The Swedish edition of Google News aggregates news from more than 100 news sources in order to find and present the biggest news [23].

2.2.4 BlogPulse

BlogPulse is a project that was developed and described in a paper. It crawled through blogs in order to discover trends [19]. BlogPulse also uses a set of data mining algorithms in order to find trends. The interesting part of this project is that it detects sudden phrase usage in the current day compared to an average usage of two weeks.

2.2.5 Living Stories

Google showed a new approach to presenting news online. Living Stories is a new format for presenting and consuming news online [30]. The idea is to collect all coverage of a particular story to a permanent URL. At one URL one will find all the news about a story and be able to follow its developments. One of the core principles of Living Stories is the story summary which helps users who are new to the story or have been loosely following the story. This gives the user a quick overview of what's happened. The contents of a story is organised by its developments and prioritized depending on how important it is. Another core principle is keeping track of what the user has read in order to highlight new updates to the story since the user's last visit. This brings a more personalized experience. An experiment of the Living Stories format was conducted from December 2009 to February 2010 by a partnership between Google, the New York Times and the Washington Post [31]. Since the package of Living Stories is open source, anyone can create their own Living Stories pages on their websites.

2.2.6 Zen News

Zen News is an iPhone application which displays a tag-cloud of keywords for discovering news headlines [36]. The larger a keyword is, the greater is the topic being discussed from various news sources. Clicking on a keyword will show additional keywords that are more specific, and another click on a keyword will reveal a list of stories related to it [54]. Clicking on a story

only show a summary of it but there is also the option of showing the story in either the embedded browser or in the Safari browser.

2.3 Layout

Newspapers and RSS readers use 2-dimensional layouts. For example Google Reader uses a list where the user only scrolls through a list of articles. Browsing by only scrolling up and down in a list is a common way to read news especially in a mobile application, such as NewsRob [34]. The reason for this is the limited size of the mobile screen. When the pile of news is starting to grow the user needs to scroll in order to get an insight into a large collection of news. This is a common problem in information seeking.

2.3.1 Visualization Techniques

Information visualization is an interesting field because it can show a lot of information in a space and ease the cognitive load of understanding the data presented [33]. Visualizing vast amount of data can be a challenging task as the visualization should show the connections of each of the information and also have to convey it easily approachable for the user to understand. Information visualizations can be very beautiful. There are various approaches of visualizing data such as tables, histograms and pie charts. Not all visualizations are appropriate for showing text data but in the next sections there will be some visualizations which shows different approaches of information visualizations.

2.3.2 Newsmap

This visualization [56] is a great example of a treemap which divides the set of information into distinguishable subsets. Colour is used for showing articles belonging to the same category, and size of cell and text show how many news sources cover the topic, see Figure 2.2. The tone of the colour indicates how new the topic is. It enables the user to easily distinguish the connections and importance within the collection of news. Newsmap visualizes the news aggregated from Google News [55]. Grouping articles together can help users to get a good overview of articles. It is developed to be used in web browsers on computers.

2.3.3 Voyage

This website [5] makes use of the layers in z-axis where the feed items further away from the screen are made not as visible by an effect of fog covering

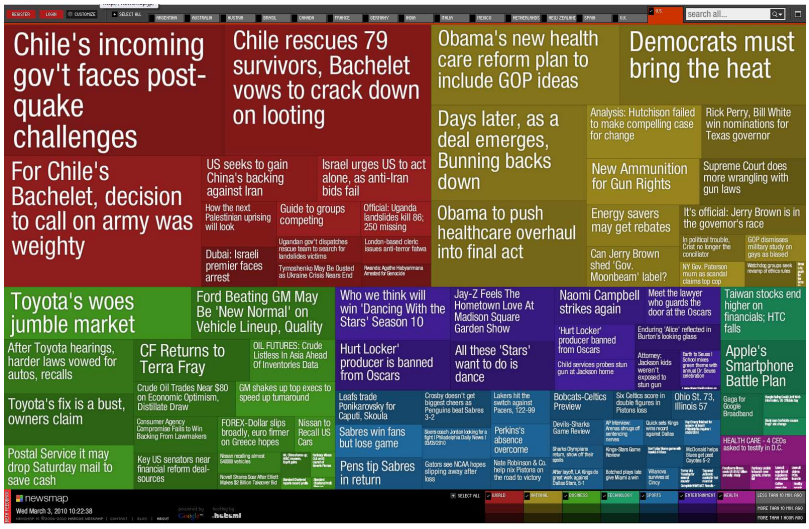

Figure 2.2: Newsmap - Visualizing news articles from Google News

them. As shown in Figure 2.3 this visualization is made of multiple layers. Each layer corresponds to a time and date. The further away a feed item is the older it is. Colours indicate the feed which the items comes from. It uses a pop up window to show the feed item when clicked. This allows the user to keep the context. This solution exist only for web browsers on computers.

2.3.4 Cooliris

Cooliris [11] shows a use of perspective in order to increase the visibility of number of items as seen in Figure 2.4. This is especially valid way of showing a large collection of items. This plug in, developed for web browsers on computers and mobile phones, lets the user search for pictures and videos. This is a good way of quickly revealing the amount of items left in the list when the mobile screen is too compact in x-axis.

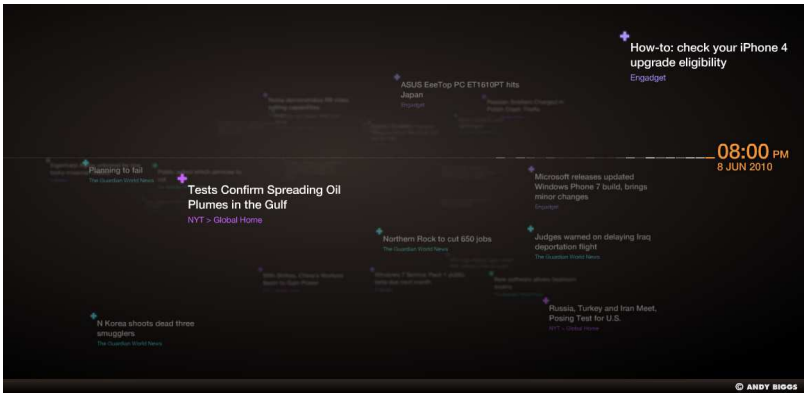

Figure 2.3: Voyage - Using depth to visualize feed items

Figure 2.4: Cooliris

Chapter 3
Research Problem

A huge problem with online news is the amount of news that gets published everyday. It's almost impossible to keep up to date with every news site and blog. This problem is commonly described as information overload. As mobile phone technology advances (larger screens, higher resolution, increasing computational power) as it does today, it will certainly be used more as a news reading device, whether it is when a user gets out of bed or during lunch breaks. Because a user usually doesn't spend a lot of time when using a mobile phone compared to a computer, it is important for a user to be able to get a quick overview of his/her news. The overview should present relevant news to a user so that he/she can more quickly read through news of interest. What we mean by relevant news is the following questions:

What has happened in the world?
Get an overview of news on a mobile phone.

Is there any big events that I should know of?
News covered by many sources.

Is there any interesting news?
News which a user prefers to read or follow.

In a RSS reader, such as Google Reader, it would take too much time to browse through large amount of news articles of each subscribed feed in order for a user to get a quick overview of what has happened. In this thesis we choose to focus on creating a prototype system that will help a user get an overview of his/her news. This is further described below. Another part of our thesis is to investigate how 3D can be used to visualize large amounts of information.

3.1 Use Case

Let us assume a user who likes to read technical blogs and news sites, but during an intensive work week where a project needs to be finished he doesn't really have the time and energy to read and follow the news. The news will pile up very quickly and the user will have to browse through all of them source by source, if he/she uses an RSS reader or a web browser. Since there are so many sources to keep track of it happens a lot that many articles from various sources contain similar content, it would take too much effort and time for a user on a mobile phone to get an overview of what has happened in the news, and will therefore lead to many users avoiding the RSS reader to read news on a mobile phone. The user will instead go to the computer.

Using our solution the user will not have to feel less comfortable to try to get an insight into the events occurred during the day or week. The user will browse the news using a more topic driven approach and get information quicker with the help of the visualization used in our solution. The user can browse articles that are associated with each other and therefore follow the progress occurred during an event. This will ease the burden for the user to browse news on his mobile phone. The user can more easily see what topics are big right now and be able to quickly know what to read.

3.2 Research Statement

This research investigates whether information retrieval algorithms can be applied such that it can aid the user into easier browsing of news articles. This master thesis also investigates if 3D visualization can help present news on a mobile phone. There will be an implementation of an IR back-end and a front-end which will show the result to users on a mobile phone. Because of the time frame performance and scalability won't be taken into account.

Chapter 4
Analysis

Our analysis is divided into two major sections. Section 4.1 is related to the layout and form of the visualization and section 4.2 is related to Information Retrieval.

4.1 Visualization System

The news presentation is also an important aspect of how the user gains knowledge of an event in the news. We investigate how visualization can help a user browse news more effectively.

4.1.1 Explore Visual Paradigms

Overview & Preview

The need for a way to scan through large amount of articles in a way that makes it easy for a user to quickly decide if a news group or news article is interesting or not is undeniably important. Here, we work with the concept of overview first and details on demand [48]. With overview we mean here by only revealing a minimal form, or surrogate [20], of news groups and news articles so that the user, with just enough information, can decide if he wants to read more of it or not. There are two kinds of surrogates:

- Overview, which is a representation of a collection of objects.
- Preview, which is a representation of a single object.

By constructing and revealing good previews and overviews for browsing news, it can aid the user in making relevance decisions and inform the user of scope, size and structure of large amount of news.

3D Visualization

The user will get a good overview if the use of perspective and positioning of the objects are good enough. It can help the user to see the natural interactions when 3D is used [21]. There has been a lot of studies in user interfaces using 3D. 3D interfaces can improve the cognitive and perceptual skills [46]. This is one of the main points of a 3D interface however other studies [10] might indicate that it can be higher cognitive load on the user if the design of the user interface is badly constructed. When 3D is used there are added complexity to the interface. As also depicted by [48] it puts more strains on the user to know where objects lie in the space. All the remarks so far indicate that 3D sets higher requirements on the design as the number of interactions and positions are increased. We still wanted to determine if the use of 3D might improve our design of the user interface. This involve the use of the z-axis and using perspective and zooming which can help the exploring of news.

Workshop

The collected feeds of news will be visualized in a mobile phone. The first approach to the problem was determining how articles could visually be grouped in such a way it's easy for the user to browse. Exploring different websites which used 3D visualization gave us an insight into different interactions and layouts. These showed different approaches to the use of a 3D space. These materials were shown in a workshop in order to generate ideas. The workshop consisted of 6 participants which were made into 3 groups. The following 3 ideas were the result of the workshop:

1. In Figure B.1 a list of topics, where each row shows tags related to its topic, is shown as the overview. When navigating into a topic the related news articles are revealed on another side of the list, by using spatial layout. The topics are still visible in order to not lose the overview.

2. Figure B.2 shows an overview showing different categories positioned in the corners and sides of the screen with related news articles floating nearby its own category. Each news article is connected to other news articles and it can also belong to more than one category. The news articles are visualized as bubbles where the title is shown. If zoomed in the bubbles can be rotated for revealing more details about the news article. Bubbles positioned further away mean older news articles.

3. A list of topics sorted by time, as shown in Figure B.3. Each row contains a topic and the number of news articles related to it. There

are two sliders, one at the top and the other at the bottom. The top slider is for adjusting the amount of topics shown to the user. If the slider is adjusted far to the right only topics with high amount of articles is shown to the user, the other topics are moved further away from the screen. The bottom slider is for adjusting the proximity of the news articles. If the slider is adjusted further to the right then more local news are shown and if further to the left then more global news. When navigating into a topic a grid of different news sources are shown together with the amount of news articles in it. There's also a recap of the topic at the top for informing the user about what the news articles are reporting about. The idea was to have an automatically generated recap for each topic. A list of news articles are shown when navigating to a news source. After choosing a news article, the whole news article is presented and a horizontal list of all news sources related to the topic is shown at the bottom.

Different layouts and ideas were presented which resulted in interesting ideas for our mockups, which are described later in section 4.1.2. From the workshop it was also pointed that we needed to narrow down to some possible use cases. We explored in what way a user want to acquire knowledge of news.

4.1.2 Mockups

Different ideas concerning our visualization were brainstormed, mockups and prototypes were created and implemented in order to see how it feels on a mobile phone.

Mapping of Parameters to 3D Space

The outline of an early use case we had was to enable users to quickly browse the news when taking an elevator and be able to quickly flick through interesting news. The key points of this idea are the following:

- Determine appropriate parameters which the result of this are described in section 4.1.3 **Choice of Parameters**.
- Mapping of a 3D space to parameters which the user would know when browsing in this space.
- Determine the use of colours.

Low-fidelity mockups was created in order to get a feel for the concept. Initial sketches were also done in order to know what possible parameters

could be mapped onto the 3D space. A brainstorming of parameters were done. This paper and pencil mockup shown in Figure B.4 and Figure B.5 gave us the insight of the concept and it showed that it required lots of scrolling through the space. The distance between the articles in z-axis was also adjusted in order to maximize the visibility. We still decided to implement a prototype using dummy data in order to know how it really felt on a mobile device. In Figure 4.1 various news articles are spread out on the screen and it's possible to move around the space by touching and dragging on the screen. News articles positioned more to the right mean higher user rank, which means other news articles positioned to the left are more general news. As for colours that is shown on each news article, more blue means higher user rank, and less blue gets more green which means more general news. The z-axis is used for the time parameter. Newer news articles are positioned closer to the screen. A slider is used to zoom in and out in order to read news articles further away. We have discovered that

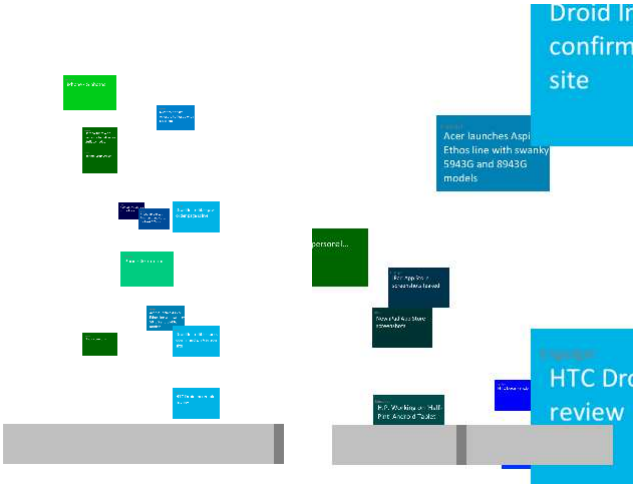

Figure 4.1: Two screenshots showing the mockup of mapping of parameters to 3D space

there were too many dimensions to browse which made it difficult for the user to know where the articles were positioned.

Zooming Interaction

As mentioned in section 2.3 list is a common way to browse on mobile phones and consists of one column. Web browsers for mobile phones also use zoom in and zoom out interaction. In our mockup when using the zooming interaction it can offer the user to get an overview when zoomed out and when wanting to get details on demand as described in [48] the

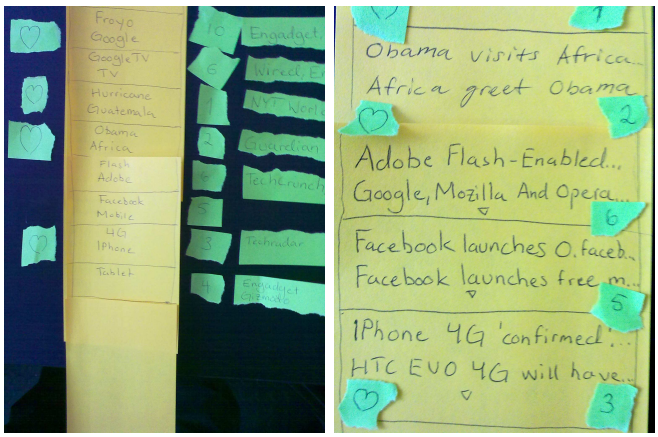

Figure 4.2: Two screenshots showing the zooming interaction

user will have to zoom in. In the case of our application we can use a list to display groups representing a topic. If zoomed out additional details can be shown and the titles of the articles in each group are replaced by the two most common used keyword in the articles within a trend. See Figure 4.2 to see how the zoomed out state looks like. When using 2D with zooming interaction it hides additional information which the user can access when in zoomed out state. This information lies hidden. Using mobile phone, not only does it require the user to scroll but also zoom in and out. This mockup was created using only paper and pencil. After evaluation of the mockup we decided that instead of hiding information and not be able to see the additional text when zoomed in it is better to make use of a perspective in a 3D space, as shown in the first idea in section 4.1.1 **Workshop**.

Combination of 3D Space and Zooming Interaction

Based on the previous mockup we created another mockup. This gave a good balance of 2D and 3D. As shown in Figure 4.3 the blue list is a list of topics where the first word is the word detected as the topic term. The other word is the word most commonly used among the news articles in the group. This approach applies the concept of dimensional congruence which as explained in [2] says that it is where spatial demands of a task is directly matched by the interaction technique that is used to execute it. The yellow column shows the additional information of which feeds news articles are in the corresponding group. The user can therefore swipe to the left in order to have that column in focus. Not only is the column visible partially for the user to peek at the feeds but also be able to get knowledge of this column. if it was only one column shown, the user might not have

Figure 4.3: Two screenshots showing use of perspective

seen this information and had to be notified in another way about it and have to guess to how to go back. The user swipes horizontally in order to see more of the yellow list. This type of interaction by rotating an object was also seen in the second idea in section 4.1.1 **Workshop**. This mockup shows only one possible case of additional information.

4.1.3 Design Decisions of Visualization

Using the result from the mockups and the studies from section 4.1.1, several decisions are made regarding the design of the visualization.

Choice of Parameters

In order to help users wanting to get a quick look at what's happening and find articles of interest we have used several parameters for visualizing news articles. We have chosen three parameters that are the main parameters affecting the visualization system:

- Trend rank which decides if the topic in a news article is hot at the moment.
- User rank which decides if a news article might be interesting for a user.
- Time, meaning the novelty of the news article.

These three parameters shows a possible pattern of reading news. The idea with trend has been demonstrated before but only visualized as a list of

articles in a table [19]. It is an interesting way to pick relevant news for a reader. It helps the user to easily discover news articles of interest in the user interface.

Overview

The overview shows the news groups/topics. In order to help a user in making a decision of the relevancy of a group/topic the surrogate of a group has been decided as the following attributes:

The trending topic term
It reveals the keyphrase which was detected as a topic term.

The most common word in the group/topic
This keyphrase complements the trending topic term for revealing more of the topic.

Title of news articles from the group/topic
It shows a preview of news articles related to the topic.

The number of articles which is related to the group/topic
It reveals the amount of coverage of the topic.

The trend rank of the group
The measurement of how hot the topic is.

The user rank of the group
The measurement of how interesting it is to the user.

The time of the latest news article in the group/topic
It informs about how new the topic is.

Groupview

The groupview shows the news articles related to a group/topic. We looked at the parameters title, URL, website title, content and date of a news article, and decided to use the following attributes:

Title
The title usually contain words that will be mentioned throughout the whole article content.

Website title
The website title can tell what kind of website that published it, thus revealing the kind of subject, for example if it was published by Joystiq (which is a video game blog) then it certainly is related to video games.

Image
An image, if the news article has any, can quickly reveal the content of the news article.

These parameters are easily extracted from RSS feeds retrieved from websites and blogs. It is reported from [35] that for users of Google News it's very common to scan only the title of the articles before visiting the real source if interesting enough. This can be compared to flicking through a large amount of news and only peek at the titles in an RSS reader.

4.2 Data Processing System

In order to have any visualization as described in previous section 4.1 the necessary parameters and attributes need to be generated. The outline of our analysis of this data processing system is covered by the following:

- **Collect**: It considers what type of text is collected and how it is collected.
- **Unify**: It covers all consideration of how news are represented. One such consideration is how groups of articles are created.
- **Personalize**: It considers relevancy of text meaning interesting articles based on user input.

In order to get an understanding of what is possible to do in achieving a better presentation of feed items we started exploring the area of Information Retrieval.

4.2.1 Theory

In order to start exploring we needed to study theories of Information Retrieval (IR). After gaining knowledge of the basics we moved on to identify and discover standard algorithms which could help in the process of determining the connections between news articles. The algorithms studied and given more weight are highlighted in this section.

Document Indexing

In IR systems a representation of the content in a document is created by going through a number of steps. Figure 4.4 shows the process of document indexing. When the document enters into an IR system it breaks the content into words. Then use a stoplist to remove words. This stoplist contains words that need to be filtered out, for example 'that', 'the', 'in', 'so' etc.

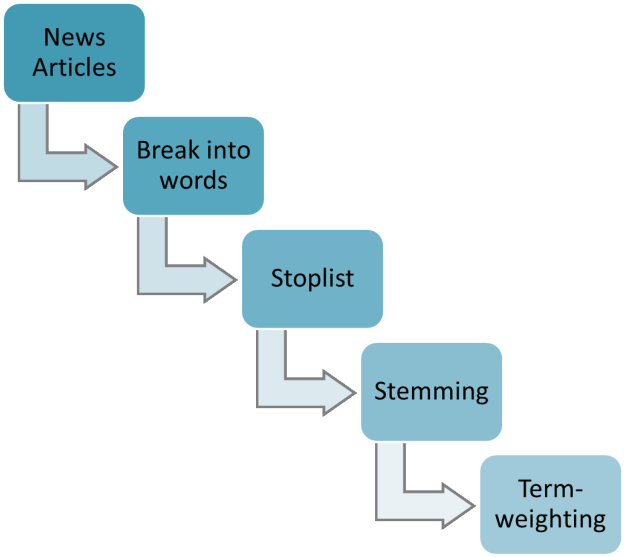

Figure 4.4: Steps of document indexing

In the next step stemming is carried out. The purpose of a stemming is to remove suffixes and transform it to its stem, thus words with the same stem will be treated as equivalent [57]. An example are the words 'upgrading' and 'upgrade' which have the same stem. This is advantageous because it will reduce the number of words that will be processed in an IR system. After stemming comes term weighting which is described below. After going through all the steps the result, which consist of terms with corresponding weights, is saved to a database.

Vector Space Model

In order to do calculations or analysis on text documents using keyphrases a mathematical representation of the articles is needed. We chose Vector Space Model [37] because of its simple model. Each document is represented by a vector of terms

$$d^T = [t_1, t_2, ..., t_i, ..., t_m]$$

where d is the vector and i has the range $[1, m]$. The terms, t_i, are grade values which are weights in the range between 0 and 1. Terms of a document are in our case keyphrases of an article. This weight is a value of how significant a term is for a document. By choosing only the most significant terms the size of the vector for each document can be reduced thus save space. Let collection c contain large enough documents to create a matrix

A

$$A = [d_1, d_2, ..., d_i, ..., d_n]$$

where a document d_i represents a vector v and $i \in [1, n]$. A is called the *term-document matrix* or *occurrence matrix*. Vector Space Model is a popular model in IR.

TF*IDF

How can weight for each term be calculated? The term weight of a term in a document is a degree of significance relative to the document. This value is calculated from TF*IDF [37]. The outline of this formula is that terms should have a higher weight value when terms occur relatively often within a document and not too much in other documents. TF is an abbreviation for term frequency which means the number of occurrences of a term within a document. IDF is called inverse document frequency and it decreases the term weight of frequent terms [37]. In most IR systems this is the final step of document indexing as described in section 4.2.1 **Document Indexing**.

Cosine Similarity

All the steps done previously are steps done to have a mathematical representation of all the documents in the collection. One of the more simple operations which uses the term-document matrix is cosine similarity. It takes two vectors v_1 and v_2 and basically calculates a cosine angle between the vectors. The similarity between two vectors is therefore calculated by the following [37]

$$d_1^T = [t_1, t_2, ..., t_i, ..., t_m]$$
$$d_2^T = [t_1, t_2, ..., t_i, ..., t_m]$$
$$cosine\ similarity = \cos(\theta) = \frac{d_1 \cdot d_2}{\|d_1\| \|d_2\|}$$

There are two ways to use this formula. This can be used to calculate the similarity between two documents. The vectors would then contain the terms of each document. The second use would be to instead calculate between a document vector and a vector of keyphrases from a query [49]. The similarity value is between 0 and 1, where 1 means identical and 0 nothing in common.

Latent Semantic Indexing (LSI)

One of the reasons for using LSI is improving the weights of the terms using co-occurrence. In other words this means that if term 1 in document A co-occur often with term 2 in other documents, LSI will give term 2 additional

weight for document A although it doesn't occur in document A [16].

It decomposes a matrix A using Singular Value Decomposition (SVD). SVD decomposes the matrix into three separate matrices [18]

$$A = USV^T$$

where A is in our case a term-document matrix. U is a matrix which consists of the eigenvectors computed from AA^T. V is a matrix which consists of eigenvectors of the matrix A^TA. The matrix S is a singular matrix consisting of singular values. Singular values are calculated by the following formula

$$s = c^{\frac{1}{2}}$$

where c is an eigenvalue. The singular values are placed in diagonal to form the matrix S.

SVD unveils hidden or "latent" data structure of the content than only counting the keyphrases [17]. By decomposing the term-document matrix it is also possible to reduce the dimensions. This is done for each of the three matrices [15]. Reducing dimensions to k dimensions mean generating U_k, S_k and V_k:

U_k = k columns of U

S_k = k rows and columns of S

V_k = k columns of V

where different values of k give different results. After reducing the dimensions, term weights can be improved [16]. LSI has several bottlenecks associated with storage considerations, difficulties in upgrading the underlying database and other speed and reliability considerations. LSI is great at addressing synonymy, which means different words with same meaning, but fail to address polysemy which means different meaning with same word [17].

4.2.2 Discovery of Software

Discovering softwares were also done in parallel with studying algorithms. In order to determine which softwares to use we collected a set of predetermined data for testing the software. It's predetermined feeds, which should contain articles that might interest our user from the use case described in section 3.1. The softwares used in our data processing system is highlighted in this section.

Keyphrase Extraction Algorithm (KEA)

We use the algorithm from KEA [39] for extracting the keyphrases of a news article.

At the start of the process, KEA cleans the input, which in our case is the content of a news article. This process involves splitting the document into tokens (sequences of letters, digits and internal periods). Punctuation marks, brackets, and numbers are replaced by phrase boundaries, apostrophes are removed, hyphenated words are split in two etc. This will result in a set of lines, where each is a sequence of tokens [13].

With these lines KEA tries to identify candidate phrases by using a set of rules [58], such as candidate phrases can't begin or end with a stopword. After obtaining a set of candidate phrases, these will be case-folded and stemmed. For each candidate phrase KEA computes 4 feature values where one of them is the TF*IDF value. The formula used for TF*IDF is shown below:

$$TF * IDF = \frac{freq(P, D)}{size(D)} \times -\log_2 \frac{df(P)}{N}$$

where $freq(P, D)$ is the number of times term P occurs in document D, $size(D)$ is the number of words in document D, $df(P)$ is the number of documents containing term P, N is the total number of documents in the collection.

Training

Before extraction of new documents take place KEA needs to learn the extraction strategy. This is done by using training articles with manually assigned keyphrases. What KEA does is identify candidate phrases, calculate feature values, and for every phrase in a document, which has occurred more than once, mark it as a keyphrase or as a non-keyphrase by using the associated manually assigned keyphrases. KEA uses the Naïve Bayes learning scheme for building the model in order to predict a phrase being a keyphrase or not (classification) [13].

Extracting New Articles

After having built a model KEA is ready to receive articles for extraction of keyphrases. Upon getting an article to extract KEA identifies candidate phrases, computes feature values and uses the model for computing the

probability of each candidate phrase in the article. The candidate phrases with highest probability are selected.

Porter's Stemmer

The stemmer we chose for KEA is Porter Stemmer because of its wide use and it has become the standard stemmer [57]. The algorithm uses the concept that all words is of the following form:

$$[C](VC)m[V]$$

where C is a list of consonants and V is a list of vowels. m is the number of times VC is repeated. Some examples are:

$$m = 0 : by, tea, I, tie$$

$$m = 1 : aunt, value$$

A number of steps are carried out for removing the suffixes of a word by using the measure, m, and rules [43].

4.2.3 Approaches

In this section shows the approaches applied into our implementations toward solving the problem addressed in chapter 3. As more news articles arrive to a RSS reader, it becomes too cumbersome to just read through all the articles. One way to solve this is to group articles to each other. Organizing large amount of data into groups or hierarchical structures is a common way of solving information overload [8] [9]. Because of the amount of news articles that gets published everyday, automatic approaches are needed to solve this [40].

Preprocessing

By taking only the ten keyphrases with the highest weight in an article we have filtered out the less important ones. The ten keyphrases are then used for representing an article. These keyphrases are the ones not only with the highest weight but appearing in the article more than once. This should lead to more accurate keyphrases that really represent the news article compared to keyphrases that only occur once and have low TF*IDF value. This in turn should help the data processing system to give a faster result when calculating the parameters for each article because of smaller term-document matrix. The parts of a news article that were chosen to be sent into KEA for keyphrase extraction are the title and the content.

Because only keyphrases that occur at least twice will be processed by KEA, there's a high probability that terms that occur in both title and content are keyphrases.

Concept Extraction

We began analyzing this algorithm because it fits well with our aim of connecting documents together. After applying LSI a concept extraction can be done. The steps of using concept extraction are shown below:

- Build term-document matrix.
- Apply LSI to the matrix. This gives three new matrices, followed by reduction in dimensions.
- The following formula is used to create the concept matrix C:

  $$C = S_k^{-1} U_k^T$$

  where the matrices S and U were described in section 4.2.1 **Latent Semantic Indexing (LSI)**. Each row represents a concept.
- For each document d in the collection meaning for each row in the term-document matrix a cosine similarity between d and a row c from the concept matrix C are used in order to see what concept a document belongs to. A document d belong to the concept c whose cosine similarity is the highest. The formula for cosine similarity is shown below:

  $$d^T = [t_1, t_2, ..., t_i, ..., t_m]$$
  $$c^T = [t_1, t_2, ..., t_i, ..., t_m]$$
  $$\textit{cosine similarity} = \cos(\theta) = \frac{d \cdot c}{\|d\| \|c\|}$$

- This value is then saved for each news article. All news articles will be grouped together using the value.

Topic Detection

One way to group articles from various sources is to group by its topic because of the importance for the user to get an overview of what has happened. The idea is to detect the topic term that is occurring in a number of news articles. When a topic term is detected, all news articles which have that topic term as one of its keyphrases are grouped together. This makes it important to find the correct topic term. Below describe a number of approaches for detecting topic terms.

Total Weight

A hot topic word should be a term which occurs in many news sources concurrently at a certain time frame with a high term weight. This would describe a certain event reported by many news sources. By calculating the sum of term weights in each news source for each term, one can detect the topic terms. The following formula is based on TF*PDF [7]:

$$totalweight_k = \sum_{i=newssource} localweight_{ki}$$

$$localweight_{ki} = \sum_{j=doc} \frac{TF * IDF_{kij}}{number\ of\ occurrences_{ki}}$$

In short, the total weight of a term is the sum of the average of its term weights in each news source. Since this value is high when a new event occurs (high term weight because it occurs relatively often in the article and few or no other articles contain it) and it gets higher as more news sources publish articles about the event, it is good for detecting topics. This algorithm relies heavily on the term weighting which makes it important to compute a good term weight for a term.

Local DF

Topic detection rely heavily on a good term weight in order to detect a trend or an event that occurs, as mentioned above. There are improvements to TF*IDF which can be done to improve topic detection [7]. TF*IDF give significant weight to unique terms in a document, that is terms which appear in only one document and nowhere else. If a certain event occurs and a news source reports it, then there will certainly be terms which uniquely identify the event. Using TF*IDF, some of these terms will have significant weight because of its uniqueness in the collection. But as the event is told by more news sources those term weights will be lowered because the document frequency of the terms are increased. This may lead to terms, that identifies the on-going event, not being detected as topic terms, thus there's a risk of losing the detection of an on-going event. Instead of having one large corpus, containing all documents from various news sources, the term weight of a term in a document is calculated using the document frequency from only one news source and the total weight of this term is then calculated by the sum of the corresponding term weights in each news source.

Percentage of Increase

Percentage of Increase is an approach to improve the precision of detecting emerging topics. In order to measure when a keyphrase is becoming

a trending keyphrase we decided to use the idea of percentage of increase. For example if document frequency of a keyphrase has increased from 1 to 2 then that means it has had a 100% of increase. if the value then increases from 2 to 6 then that means having 300% of increase.

This approach focus on the use of keyphrases in an article. It's a simple counter of number of occurrences in documents. If a keyphrase is used often during a time frame then it's an indication of a possible keyphrase that describes a topic. Percentage of increase shows a certain increase, or decrease, in usage of a keyphrase.

In order to detect that certain events are emerging as hot topics we need to set time frames which are further explained below. All the keyphrases within that time frame will have its percentage of increase calculated by the following formula:

$$poi_k = \frac{n_{new}}{n_{old}}$$

where k is the keyphrase which the percentage of increase is calculated for, the numerator n_{new} is the number of articles in which the keyphrase appear in during the time frame and the denominator n_{old} is the number of articles the keyphrase have appeared on before the time frame. If a keyphrase hasn't appeared before then the percentage of increase is the same as the number of articles it appears on during the time frame. In the example in Figure 4.5 there are three time frames. During the first time frame four news articles were published containing the keyphrase Android. This gives the keyphrase 400% of increase as seen in Figure 4.6 because there were no news articles containing Android prior the first time frame. During the second time frame the same amount of news articles that contain Android were published as depicted in Figure 4.5, which gives a 100% of increase compared to previous time frame. The keyphrase Android only gets a 25% of increase in the last time frame because of only a single news article that contain Android was published. As also seen in Figure 4.6 the keyphrase Oil which hasn't appeared in any of the published news articles in the first time frame gets a much higher percentage of increase value in the second time frame than the keyphrase Android, although both keyphrases has the same number of occurrences in the second time frame. This is because of the number of occurrences of Oil during the first time frame is much lower than the keyphrase Android. This helps new emerging topic terms to be detected. The time frame is an important part of detecting the topics and it could span from 1 day to several days. This is discussed further below.

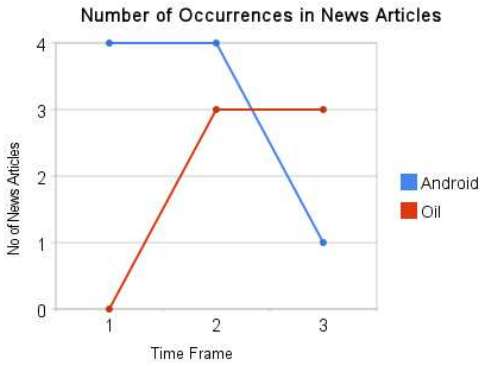

Figure 4.5: Example showing the number of occurrences during three time frames

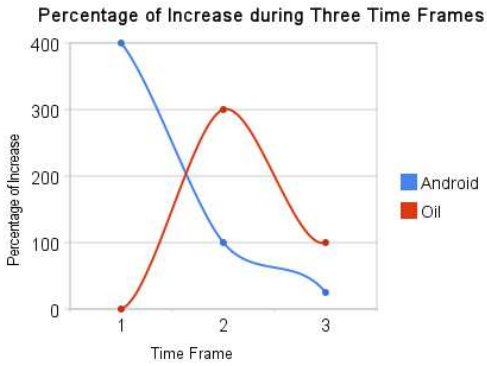

Figure 4.6: Example showing the percentage of increase during three time frames

Time Frame

When computing the percentage of increase for all topic terms, a fixed time frame is used to track the keyphrases which has occurred in news articles that were fetched within the time frame. Trying to find a good value for the time frame is hard and requires some experiments. This is because different topics have different life times, an event such as a car accident may only be a hot topic for one day, but other events such as a big sports event, that is run in several days, would be a hot topic for a longer time. Another important aspect when deciding the time frame is when a user thinks that a topic is old. Because when a topic term becomes hot then it stays hot as long as the length of the time frame. This means if the time frame is long then the user will see that topic for a longer time and the topic may not be so up-to-date anymore for the user as time passes by. For our thesis we

decided to have a 3 day long time frame.

Trend Ranking

A value, which is used to measure how hot a topic term is, is something we call trend rank. The trend rank of a topic term is computed by multiplying its total weight with its percentage of increase.

$$trendrank_k = poi_k \times totalweight_k$$

where k is the keyphrase which the trend rank is calculated for. Computing the trend rank this way, topic terms occurring in a lot of news articles from many news sources have high trend rank, but new topic terms that have just begun showing up in a few news articles and are getting step by step more coverage will also get high trend rank. This is so that a user can discover early emerging topics.

User Profile

In order for the collection of news to maintain its readability for the user it is also important to introduce a more personalized reading. Personalization can be used to make news reading easier [4]. This parameter is very important as manual input of interest of field is a very effective way to present relevant articles for the user. It is therefore preferable for our users to manually enter keyphrases matching area of interest. A user profile containing those keyphrases with weights is used to measure how similar a news article is compared to it [4]. The calculation is done by using cosine similarity as described in previous section 4.2 **Cosine Similarity**. For each news article d and the user profile u a cosine similarity is calculated between them by the following formula:

$$d^T = [t_1, t_2, ..., t_i, ..., t_m]$$
$$u^T = [k_1, ..., kn]$$
$$cosine\ similarity = \cos(\theta) = \frac{d \cdot u}{\|d\|\|u\|}$$

where d is a vector containing its term weights and u containing the term weights of the user profile. We call this value as the user rank. The problem with this is that the user might not always know what keyword is best to use to match his/her interest. This introduces therefore two ways of affecting the user profile. One of them is by manually entering keywords which have the largest weights directly after input. The other factor is a user model which monitors what articles the user has selected to read. Introducing this functionality allows for more dynamic way to present more relevant news as the user reads more. This encourages exploration of topics as a certain topic that the user follows and read will have an increased value in the user's user profile.

Chapter 5
Result

In previous chapter 4 explained the approaches and decisions which in this section will be applied into implementing a whole system. This section covers details of implementation and outputs from our finalized prototype system.

5.1 Technologies

Throughout the project different hardware and software were used. The working environment were thankfully provided by TAT such as computers and also mobile phones for running the application in order to get feedbacks and eventually the presentation, as well as getting assistance from employees during the project. The main development computers were using Windows 7 Professional. The mobile phones used for testing on a real environment were Nexus One and HTC EVO 4G which were both running Android OS [27].

Software and tools were of course more comprehensive than the hardware. Some of the software were thankfully provided by TAT. The following list show IDE, SDK as well as tools: Eclipse Galileo [41], TAT Motion Lab [51], MySQL [1], TAT Cascades for Android [50], Android SDK 2.1 Platform [24], Android SQLite Database [25], Android Xml Utility Methods [26], Java 1.6 SAX [38], Java 1.6 JDBC [32], phpMyAdmin [44], GIMP 2.6 [52] and JAMA 1.0.2 [22].

5.2 Architecture

In order to analyze and process all the information from RSS feeds we chose to use client-server solution instead of having only the client, which is a mobile application, to do all the collecting and computations. It was partially because the amount of news articles that are collected and the

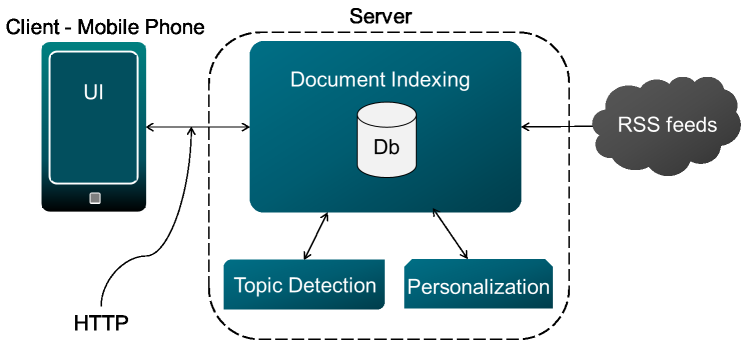

Figure 5.1: The architecture of the whole system in an overview

computations done on 6000 news articles would be too much for a mobile phone to handle and it also gave us the possibility to use softwares which use existing algorithms in data mining. The whole system therefore contain a server which aggregates feed items and send them to the client, see Figure 5.1. The client handles the visualization of the articles and that is how the user interacts with the data processing system. When deciding which protocol to use between the client and the server it was decided that HTTP communication would be used because of its widespread use. As shown by Figure 5.2 when a request is sent to server, history data and user profile are also sent to the server. The server sends a response back to the client when the request is received. The response, which is encoded as an XML file, contain feed items with associated meta data and additional data that is produced from the server. Using XML allowed us to have a more structured way of presenting result to the client such that the parsing of the result was made easier using classes from Android API [26]. The client, which is an Android application, will then present and visualize the result for the user. Synchronization between the client and the server is done periodically. In the coming sections more implementation details in the server and client will be explained.

5.3 Server

The server is implemented entirely in Java. The following tasks are performed in the server.

- Collect RSS feeds
  - Fetch and parse the feeds.
  - Save the parsed contents of news articles and its associated meta data to database.

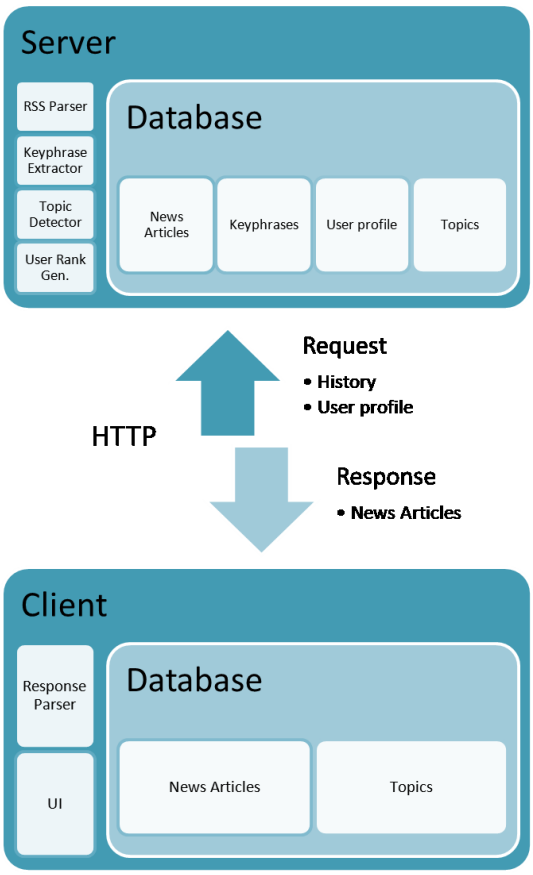

Figure 5.2: The architecture of the whole system in a more detailed view

- Extract keyphrases.
  - Top ten keyphrases with highest term weight of each news article are saved to database.
- Compute trend ranking
  - Compute total weight and percentage of increase.
  - Assign each news article to a group.
- Advance time frame
  - The time frame for which topic terms are tracked and detected, is advanced.
- Compute user ranking.

- Compare current user profile to each news article and assign the user rank as the resulting cosine similarity.

- Update user rank
  - Decrease weight of keyphrases in user profile.

- Receive client request
  - Send news articles, together with its associated data including meta data, trend rank, topic term and user rank.

5.3.1 Database Design

The server uses MySQL database [1] to save all necessary data. Figure 5.3 shows an E/R diagram of the database used by the server.

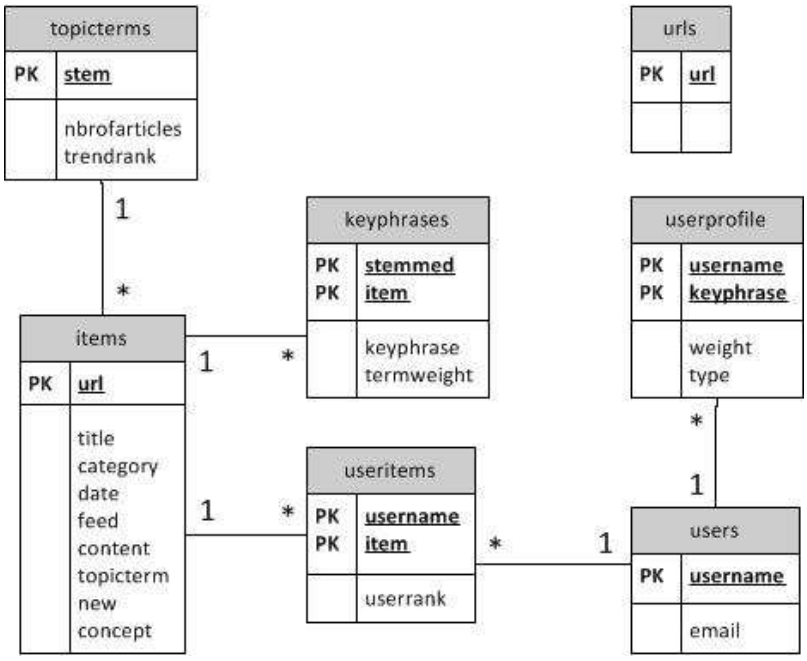

Figure 5.3: E/R diagram of the database used by the server

5.3.2 Collect RSS Feeds

The server fetches RSS feeds from a predetermined list which is stored in a table in the database. It fetches RSS feeds through an HTTP request sent to the server where the content of the RSS feeds is stored. Upon receiving the

HTTP response that contains the RSS, XML parsing is done using JAVA 1.6 SAX [38]. The following parameters of each RSS item (news article) are parsed and saved to database:

- Title
- Date
- Content
- URL
- Website title
- Categories

Only RSS items that doesn't exist in the database are saved. Fetching RSS feeds are done periodically in an interval of 10 minutes.

5.3.3 Extract Keyphrases

After saving the newly fetched news articles, keyphrases are extracted from them. As described in section 4.2.2 **Keyphrase Extraction Algorithm (KEA)** KEA is used for this task. Modifications of the source code of KEA were made in order to add support for Local DF and choosing the top ten keyphrases with highest term weight. When our server is first started an initial set of news articles are used for creating a model, and document frequencies of the keyphrases are created for later processing. The initial set of news articles used for training were chosen from general and technology RSS feeds. The requirement for the website feed was it needed to have categories for each feed item, since these are treated as manually assigned keyphrases by the author and will be used during the creation of model in KEA. For each news article whose keyphrases are to be extracted, the document frequencies of the initial set of news articles, together with other fetched news articles that come from the same feed as described in section 4.2.3 **Local DF**, are loaded and used during computation of TF*IDF feature value of each keyphrase. The title and the content of the news article are sent into KEA for keyphrase extraction. At the end of the process, all the keyphrases of the current news article are sorted by descending TF*IDF value and the top ten keyphrases are saved in the database.

5.3.4 Compute Trend Ranking

In this step the trend rank, as described in section 4.2.3 **Trend Ranking**, is computed for each unique stored keyphrase that belongs to news articles

which were fetched within the time frame. Afterwards a list of keyphrases sorted by trend rank in descending order is queried in database for assigning all news articles into groups. The keyphrase with highest trend rank is first taken and all articles that contain the keyphrase are assigned to the same group. The same process is done for the next keyphrase and so on. If the news article is already assigned to a group, it won't be re-assigned since the keyphrase with higher trend rank is more likely to be a topic term. A group is identified by the keyphrase which a news article was assigned to. The output of this step is that all related news articles are grouped together. By using the trend rank of each group, one can measure the hotness of a topic.

5.3.5 Advance Time Frame

The server advances the time frame in a fixed interval. When that happens all previously fetched news articles will be treated as old, thus keyphrases that occur in old news articles will be left out when computing the trend rank. This means news articles won't be grouped by those keyphrases anymore. In the new time frame there will be new groups/topics. Before advancing the time frame, the number of occurrences of each keyphrase within the time frame is saved for computation of percentage of increase.

5.3.6 Compute User Ranking

For each article there will be a calculation of similarity using the user profile as described in section 4.2.3 **User Profile**. In order to make this possible a term-document matrix of the collection of news articles is created. This term-document matrix contain an additional document column. This column is the user profile. This makes it possible to measure the similarity of the news articles and the user profile. This approach offered surprisingly accurate result. It was relatively simple to implement and used a formula which we were quite familiar before.

5.3.7 Update User Ranking

As user clicks and read news articles the user profile is updated. In order to keep the user profile as close to the user's interest, each keyphrase in the user profile has a weight. A manually entered keyphrase has weight 1.0 and is never changed. A keyphrase that is added by tracking the user's reading history have an initial weight and it is incremented as news articles containing that keyphrase is read by the user. The weight of those keyphrases are periodically decreased in order to filter out keyphrases which only occurred once or few times in news articles read by the user.

5.3.8 Receive Client Request

Upon receiving an HTTP request from a client the server sends an HTTP response encoded in XML which contain a fix number of news articles, together with all its associated data including meta data (title, date, URL etc), trend rank, topic term, user rank, score, novelty and most commonly used keyphrase in a topic. Score of a topic is used as a sorting parameter on the client, this will be described in section 5.5. The score of a topic is computed by the average of trend rank plus user rank of all news articles in the topic.

$$\frac{\sum(trend\ rank + user\ rank))}{number\ of\ articles\ in\ a\ topic}$$

The HTTP request from the client may contain a reading history and a user profile. The reading history contain all URLs which the user has read since last synchronization. If a reading history is received then the user profile of the user, which is stored in database, is updated by the server. The other part of the request is the user profile which contain manually entered terms, and if this is received then the existing manually entered terms in the user profile are deleted and replaced with the new. An example of how reading history and user profile looks like in the HTTP request can be found in section A.2. An example of the HTTP response from server can be found in section A.3.

5.4 Outputs

5.4.1 Test Corpus

For testing our system we have a predetermined set of RSS feeds. This set represents what we think are common well known websites in the web and match the use case described in section 3.1. These websites update frequently which introduces some difficulty for the user to have the time to go through if read in a normal RSS reader. This set has been used to test the result calculated from the server. The evaluation of the result has mostly consisted of manual observation of the result. The RSS feeds which the corpus was fetched from can be found in the appendix. All the result shown are taken from feed items that were available during 10th May - 24th May 2010 which resulted in around 6000 news articles. The time frame for which the trend rank was computed on was set to 3 days.

5.4.2 Configuration in KEA

KEA has a number of parameters that can be tweaked, here are some of them and the values we chose for running our system:

- Number of keyphrases for each article: 10
- Minimum number of occurrences in the article: 2
- Maximum number of words in phrase: 2
- Minimum number of words in phrase: 1
- Vocabulary: none
- Encoding: UTF-8
- Language: en
- Stemmer: PorterStemmer

5.4.3 Concept Extraction

Table 5.1 shows articles from a group which was computed by concept extraction. The result shown below was computed by using $k = 100$ [15]. Some other values of k, such as $k = 40, k = 10$, were tested which didn't improve much compared to $k = 100$. The result was based on news articles from 19th May to 20th May. As shown in Table 5.1 the articles' content

News articles of a group from 20th May
AT&T Not Worried About Verizon iPhone [Digital Daily]
Microsoft to give governments patch previews
Live From The Google I/O Keynote
Google I/O: The Web Is Killing Radio, Newspapers, Magazines, And TV
Adobe hastens release of HTML5 developer tool
Google Wave Opens For Everyone
Sports Illustrated Shows Off An HTML5 Magazine
Viral Video: Smoke Monster on "Lost" Gets Spin-Off! ...
Official BlackBerry Twitter app getting a much-needed update
Turkish energy minister says rescuers in mine hit by explosion have...
The Google Rule [Voices]
UK iPad App Store open for business
What Is Froyo? [Froyo]

Table 5.1: News articles grouped using Concept Extraction

is too diverse for belonging to the same group. This was done in one of the earlier phases of our project. Although trying to change the value of k, which is the maximum number of groups allowed, it was decided that this method was too insufficient because of its computation time and also because of its calculation result. It was difficult to interpret what each group represented.

19th May			20th May			21st May		
Beta	15	24.0	Google TV	28	77.86	Google TV	43	115.32
Tethering	2	23.15	TV	14	33.20	TV	21	46.31
WebM	7	9.52	Tethering	11	30.06	Froyo	23	34.12
Wave	7	9.22	Beta	12	29.70	Tethering	11	33.68
Web Store	5	6.69	Froyo	15	24.27	Beta	11	32.86
iPod Touch	8	5.52	Web Store	9	16.93	WebM	12	21.71
Touch	2	5.5	Wave	11	14.75	Web Store	9	16.93
Auction	4	4.73	WebM	9	11.94	Wave	12	16.2
Web Video	3	4.49	Museum	10	10.54	Pac-Man	8	15.26
Vietnam	1	4.23	Stolen	3	7.33	Museum	13	14.61

Table 5.2: 1st column: topic, 2nd column: no of articles, 3rd column: trend rank

22nd May			23rd May			24th May		
Nexus	9	10.03	Nexus	11	12.15	Nexus	17	22.23
Bangladesh	3	3.35	Air India	6	6.9	Disrupt	26	21.22
Deadliest	2	3.09	India	7	5.03	WWDC	7	12.92
India	8	3.09	Party	6	4.85	Jamaica	9	12.09
Shanghai	3	1.99	Derailment	3	4.44	Australia	7	9.56
Party	3	1.94	Bangladesh	4	4.24	Party	9	9.17
Father	2	1.82	Cleric	5	3.85	Air India	8	9.11
Disrupt	3	1.49	Shack	3	3.52	Duchess	7	8.51
New Leader	1	1.42	Shadow	3	3.41	Security Council	7	7.36
Drone	2	1.34	Disrupt	7	3.35	i7	8	7.1

Table 5.3: 1st column: topic, 2nd column: no of articles, 3rd column: trend rank

5.4.4 Top 10 Topics in 6 Days

Topics detected by the server during 19th May - 24th May are shown below. The time corresponds to 2 time frames, where 19th May - 21th May is the first time frame and 22th May - 24th May the second time frame. By the result in Table 5.2 and Table 5.3 one can see that the trend rank of a keyphrase increase as more news articles containing that keyphrase are fetched and stored. E.g. the WebM keyphrase on 19th May occurs in 7 news articles with a trend rank of 9.52, and the next day a total of 11 news articles has been fetched containing the WebM keyphrase which leads to an increase of trend rank to 11.94. The effect of advancing a time frame can be seen on 22nd May where a whole new list of top topics is shown. As mentioned before it's because the computation of trend rank won't be

tw & ldf & poi			tw & poi			tw & ldf		
Google TV	28	77.86	Google TV	28	70.77	TV	42	3.16
TV	14	33.20	Beta	19	29.08	Tethering	11	3.01
Tethering	11	30.06	TV	16	28.23	Facebook	20	2.23
Beta	12	29.70	Tethering	4	27.99	Chrome	17	2.12
Froyo	14	24.27	Froyo	15	22.03	iPhone	45	1.99
Web Store	9	16.93	Web Store	9	15.53	HP	18	1.8
Wave	11	14.75	Wave	11	14.76	iPod	10	1.75
WebM	9	11.94	WebM	9	10.99	Korea	15	1.68
Museum	10	10.54	Museum	9	8.57	Google	75	1.65
Stolen	3	7.33	Troop	8	7.8	Plane	5	1.64

Table 5.4: Comparison of approaches applied

based on news articles from before the advanced time frame.

5.4.5 Top 10 Topics on 20th May

The result below compare the topic detection approaches in section 4.2.3 **Topic Detection**. These topics are the topics with the highest trend rank during 20th May and correspond to only a subset of all the topics detected by our server. The titles in Table 5.4 and Table 5.5 tw, ldf and poi are abbreviations for total weight, local df and percentage of increase respectively. Some of the news articles belonging to the topics are shown in section A.4.

poi		
Google TV	28	28.00
Beta	17	19.00
Froyo	20	19.00
iPhone OS	1	12.00
Wave	11	11.00
TV	14	10.50
Museum	10	10.00
Thethering	3	10.00
amp	7	9.00
stolen	3	9.00

Table 5.5: Topics generated by percentage of increase

Total Weight & Local DF & Percentage of Increase

Table 5.4 **tw** & **ldf** & **poi** contains the result of top topics detected by our system using all approaches, which were all described in section 4.2.3.

Total Weight & Percentage of Increase

In Table 5.4 **tw** & **poi**, which shows the result based on only the two approaches, total weight and percentage of increase, give almost the same top topics as Table 5.4 **tw** & **ldf** & **poi**, except the trend ranks of each topic term have all decreased. Another important observation is the trend rank of Tethering has decreased more than Beta. If the two keyphrases are compared, both occur in 9 feeds, but the number of news articles that contain Beta is 27, compared to 15 for Tethering. When Local DF was used Tethering got higher trend rank because of its uniqueness in each occurring feed compared to Beta. This affects the term weights because document frequency of Tethering is lower in each feed than using document frequency of the whole collection. It indicates that Local DF help increase the trend rank of early emerging topic terms as more news sources reports about it. As seen in section A.4 the topics Tethering and Beta contain news articles reporting about similar content. But Tethering is a more specific topic term because it's a feature from the iPhone OS 4 Beta, which makes it preferable that it gets higher trend rank than Beta.

Total Weight & Local DF

As seen in Table 5.4 **tw** & **ldf** using the two approaches, total weight and local df, the first top topic is TV instead of Google TV. Because of this news articles that contain the term TV is grouped together, including Google TV related news articles. The real hot topic is really Google TV that is announced. Compared to Table 5.4 **tw** & **ldf** & **poi**, TV is a more general topic term than Google TV. Adding percentage of increase would give a more specific topic term. The reason is that the keyphrase Google TV is a new keyphrase and has never occurred before the current time frame, thus the percentage of increase of Google TV is quite high and that would help increase its trend rank. The same problem occurs for the topic term iPhone, where the real hot topic is about iPhone OS 4.0 Beta. In Table 5.4 **tw** & **ldf** the topic term, beta, is instead detected as the hot topic, which is more specific and give better grouping of the news articles.

Percentage of Increase

The top topics, which are listed in Table 5.5, are similar to Table 5.4 **tw** & **ldf** & **poi** but one topic that looks out of place is 'amp' which is originated from the news articles that include & which is a character entity reference for '&' in HTML/XML. Such characters are mostly unwanted and should be filtered out for example using stoplist. Using the total weight approach, the term amp wouldn't be detected as a hot topic because amp occurred mostly in Engadget news articles and was thus given a lower trend rank.

From this result we can draw the conclusions that a potential topic term should be a keyphrase that occurs in multiple news sources within a time frame.

5.4.6 Match User Profile to News Articles

Table 5.6 shows some of the top news articles matching a user profile. The news articles are from 20th May and the user profile consists of the keyphrases *movie*, *htc*, *intel*, *game* and *china*. It can be seen that all news articles in the result contain one or more keyphrases matching the user profile. In this case, the keyphrases used in the user profile matches most news articles with low trend rank. The user rank can thus help to show other kinds of news articles so that news articles presented to the user won't be dominated by news articles related to hot topics.

5.5 Client

The implementation of the client was developed on the Android platform using TAT's UI framework, TAT Cascades. Thanks to TAT Cascades we could create and experiment user interfaces freely, without the limitation of traditional user interface components. Because of how TAT Cascades works we could implement the user interface on the client without relying on the progress of our server, by using dummy data on the client when testing the user interface. This way we could implement the client and server independently throughout the development.

The client which is run on Android platform on a mobile phone does the following tasks:

- Visualize news articles
  - Using TAT Cascades for creating the user interface.

Similar.	News Article Title	Feed	Date	tr. rank	topic term
0.4212	HTC Fights Back, Suing Apple to Block Import of iPhone, iPad, iPods	DailyTech Main News Feed	2010-05-13 03:05:00	0.68	htc
0.3429	Top 40 movies not out on Blu-ray	CNET News.com	2010-05-18 19:50:00	0.00	movi
0.2855	Create your own games in LittleBig-Planet 2	CNET News.com	2010-05-10 19:33:51	0.84	creat
0.2818	Nine killed in latest China school rampage	Reuters: World News	2010-05-12 16:23:53	0.00	school
0.2767	Why iPhone Hasn't Sold Well in China [Voices]	All Things Digital	2010-05-20 09:00:05	3.46	iphon
0.2714	Take the horse, leave the prince: the games to buy this week	Ars Technica	2010-05-20 18:39:00	1.21	game
0.2696	Intel Brings A Tablet to Computex 2010	I4U News	2010-05-12 17:42:41	0.06	computex
0.2633	Drive the A-Team Around Google Earth [Movies]	Gizmodo	2010-05-18 23:40:00	0.00	earth googl

Table 5.6: Similarity between news articles and user profile

- Request and receive news articles
  - Send HTTP request for news articles and store them in its local database.

5.5.1 Visualize News Articles

Using TAT Cascades the news articles stored in the local database are visualized to the user. There are a total of three views: Overview, groupview and detailview. The overview, which is shown in Figure 5.4, shows a list of topics sorted by the score of a topic in descending order. This means that topics with high trend rank and low user rank or low trend rank but high user rank will be put to the top of the list. Each row corresponding to a topic reveals the following:

- Topic term
  - Example: *Pac-Man* shown in the picture above
- Most commonly used keyphrase in the related news articles

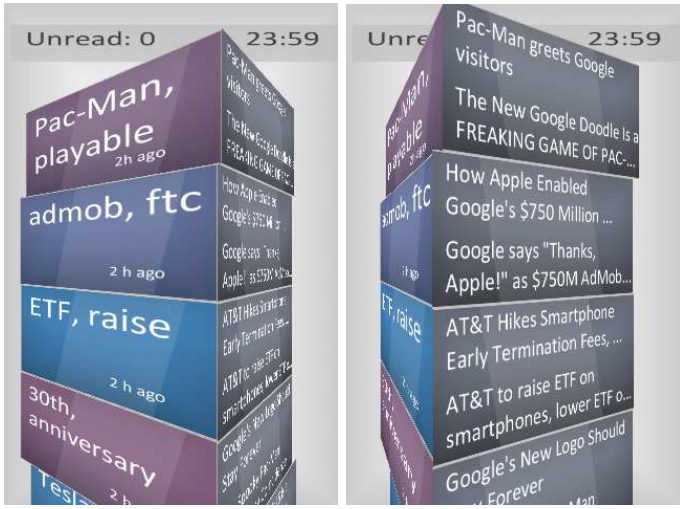

Figure 5.4: Two screenshots showing the overview of our final prototype application

  - Example: playable

- Date/time of the latest news article
  - Example: 2 h ago
- Titles of the two latest news articles
  - The two text shown on the right side of each rectangular box
- Number of news articles
  - The size of topic term and most commonly used keyphrase
- Average user rank
  - The red colour, more red means higher average user rank
- Novelty
  - The offset in the row, the closer the row is to the screen the newer the topic is, which is measured by the latest news article in the topic.

More pictures of the overview can be found in section B.3. The groupview, see Figure 5.5, is shown when clicking on a topic in the overview where a list of news articles in the current topic is revealed. Each row reveals a website title, a title and an image, if any, of a news article. The background colour

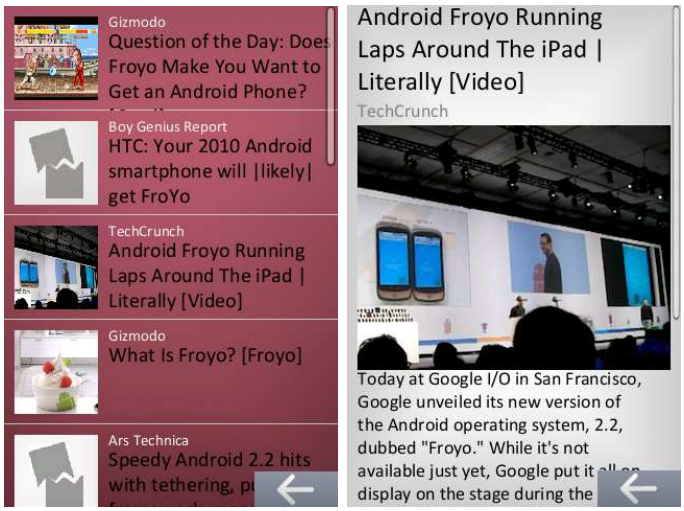

Figure 5.5: Two screenshots showing the groupview and detailview of our final prototype application

reflects the same colour as the row in the overview. The detailview, see Figure 5.5, is shown when a news article is clicked. All the details related to the news article are displayed here. This includes the title, website title, content, date/time and image if any.

5.5.2 Request and Receive News Articles

The client sends periodically HTTP request to the server for news articles in the background. The HTTP request may also contain the reading history of the user and a new user profile. The reading history consists of URLs of news articles which the user has clicked into detailview to read. The user profile consists of a list of manually entered keyphrases by the user. After receiving a response from the server, which is a response encoded in XML containing a list of news articles and its associated data (meta data, trend rank, topic term, user rank, score, novelty and most commonly used keyphrase in a topic), the client parses the response and saves all the received data to its database. The user interface will then be updated.

5.5.3 Impressions of Client

Our final prototype system was tested on some users. The mobile application was run on a HTC EVO 4G mobile phone. Here are some comments and feedbacks received from the users.

- "It is good to show more than one keyphrase (two) in order to under-

stand more about the topic in the overview."

- "It is quite good that there is an offset in the position of a topic row in the overview which might help you when you're scrolling fast."
- "It may not be obvious as to how the list is sorted. It would be good to somehow indicate that most of the topics at the top are the biggest topics."

By using the two surrogates overview and preview it aided the user in making relevance decisions of topics. It also gave an overview of what has happened in the news. The user could scan through a lot of topics and was able to gain knowledge of big events and interesting news.

Chapter 6
Discussion

6.1 Topic Detection

6.1.1 Specific VS General

Our system could detect surprisingly more specific topics as shown in Figure 6.1. What is important to notice in the diagram are the days 20th May and 21st May. 20th May was the day a new version of Android OS, called Froyo, was announced. The less specific term Android got lower trend rank than the more specific term Froyo which resulted in Froyo being detected as the topic term and news articles related to it were grouped. If Android had gotten a higher trend rank than Froyo it would result in a group containing more diverse content, including all news articles related to Froyo and Android in general. As shown in the diagram below there is also the comparison of the terms Google and Pac-man. Our system actually detected the topic Pac-man where Pac-man had its anniversary celebrated by Google. It successfully chose pac-man instead of Google as shown in the diagram where the topic Pac-man emerged on 13th May but also 21st May. The important thing here is therefore that the term Google always had much lower value than pac-man.

6.1.2 Comparison with Techmeme Topics

Comparing the result of our topic detection on 19th May - 20th May to the topics from Techmeme [45], several topics which were presented in Techmeme were also detected by our system. Some of those topics detected were Google TV, Froyo, WebM, Web store, Wave, HTC EVO 4G and Foxconn Suicide. Although our topics match to some degree but because Techmeme also has human editors involved in creating the topics it contain more correct topics than our result. This is compared to our fully automatic system.

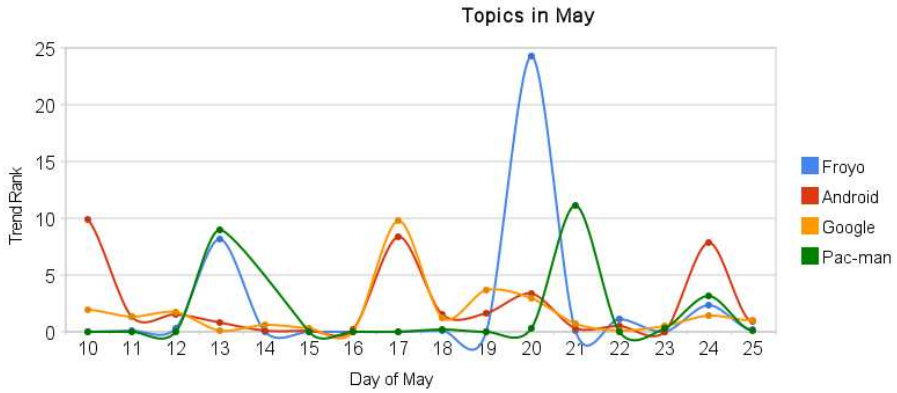

Figure 6.1: Specific topic terms vs general topic terms

6.1.3 Comparison with Google Trends

Figure 6.2, which is generated by Google Trends [29], reveals news reference volumes in May 2010 containing five keyphrases Google TV, Froyo, WebM, Web Store, Torpedo. As one can see the top keyphrases on 20th May is ordered by Torpedo, Google TV, Froyo, WebM and last Web Store. Our system gave the same result except for the keyphrase Torpedo which was ranked as number 25. The result of this depends on which news articles the trend was calculated on. Because we only used 4 feeds that published world news compared to 13 feeds publishing technology news, the number of occurrences of the torpedo keyphrase was much lower than the Google TV keyphrase.

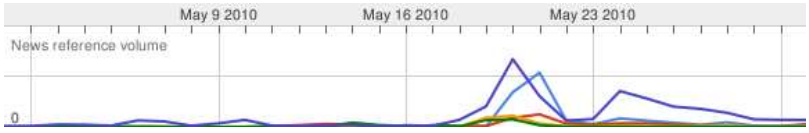

Figure 6.2: Generated by Google Trends, Google TV: light blue, Froyo: red, WebM: yellow, Web Store: green, Torpedo: purple

6.1.4 Advantage & Disadvantage of Tracking All Keyphrases

Tracking all keyphrases stored in the database for detecting topic terms has its advantages and disadvantages. The advantages are it's simple and works in some degree as described above. The disadvantages are that an event that is reported in news articles usually contain more than one keyphrase for describing an event. All those keyphrases will get a trend rank based on those news articles. But our solution for choosing a keyphrase for grouping

related news articles together is by choosing the one with highest trend rank. The other related keyphrases will still be used if there is other news articles that contain those keyphrases. The most ideal outcome would be that no other news articles contain those keyphrases and they won't be used for grouping news articles together. But a possible outcome is that some related news articles doesn't contain the chosen topic term but they really belong to that topic. The reason for this may be that the keyphrase isn't among the top keyphrases with highest term weight that is saved during extraction of keyphrases or that the keyphrase doesn't occur twice because of too short text. This is partly because of RSS which often only contain summaries of the full content. The result of this is that a user will find two or more groups that contain news articles related to the same topic. The worst outcome is that a few news articles that is totally unrelated to the topic is grouped together. An example of this problem is there are several news articles reporting about an iPod Touch with camera spotted in Vietnam. Our system detected the topics iPod Touch and Vietnam, and iPod Touch was chosen as the topic term. All news articles that contain iPod Touch were grouped together. But the topic term Vietnam which had a lower trend rank was still used because a certain news article reporting about a totally unrelated event in Vietnam contained the keyphrase Vietnam. This is certainly incorrect because that news article didn't contribute much to the increase of trend rank of the keyphrase Vietnam.

6.1.5 Introducing to New Users

The result of topic detection can be improved as more news sources are added. This may introduce some difficulty for a user when using our system for the first time because he/she may not know many news sources that publish news articles related to his/her interests. An idea would be to have a wizard at the startup of the client when the user uses our system for the first time. The wizard would show a list of subjects which the user may be interested in, e.g. Gadgets, Video Games, Fashion etc. These subjects would then have an initial set of feeds which our server will collect from. This way the user only has to decide which subjects the user is interested in.

6.2 User Profile

As the user read more news articles the user profile will get more keyphrases from those read news articles. A disadvantage of this is that it decreases the cosine similarity. This is because of how cosine similarity works, as partly described in [7]. As more keyphrase is added to user profile vector, more

matching keyphrases on a document vector are required in order to get the same cosine similarity value. The user rank assigned to news articles is better with manually entered keyphrases.

The result was only based on 6000 articles, as more sources is added and more articles are saved, it becomes a problem when computing the user rank. The term-document matrix becomes too large and it will lead to out of memory on the server. A solution would be to instead have a data structure of indices for where terms should be placed in a document vector, this can then be used to compare a news article with a user profile at a time.

6.3 Architecture

The advantage of having a server-client solution for a user is having an assurance that all news articles and its associated data are stored in a seperate storage. In case of a failure of a mobile phone the user can easily switch it without worrying about data loss. Another advantage is that the mobile phone doesn't need to do any heavy computations[1] of news articles thus also saving its limited battery life. The disadvantage of a server-client solution is since all the data is stored in a central server it could introduce a bottleneck when the number of requests is increasing.

6.4 Visualization

By dividing the visualization into overview, groupview and detailview the use of 3D could be better utilized in the overview. This is because text is not always easily read from all angles in 3D. The overview contain less detail and text than when browsing in groupview where you have the titles of all news articles belonging to a group and in the detailview showing the whole content of the article. The good thing about 3D is it is good at showing how the user can interact with the interface, for example you know you can rotate the list in the overview. It is good at hiding information where two sides of the rectangular box are utilized. By setting the viewpoint such that another side of the box is revealed, additional text can be placed there with the user easily noticing it. Utilizing the sides of the rectangular box a user doesn't have to get overloaded with information. Thus more topic can be seen on the limited screen space instead of showing all text related to a topic at once. When the user is inside a topic and browsing the articles it is better to have a 2D list because the user is reading more text which

[1] Computations of trend rank and user rank.

means focusing one item at a time. As a user who wants to get news fast and easily it is important to visualize parameters that inform as much as possible about a topic in order to help a user making the decision to read it or not [20]. This was done by revealing all the parameters described in section 4.1.3 **Overview**.

Only 2 sides of the rectangular box in the overview are used for the visualization. A concept which might be interesting would be to be able to rotate the rectangular box to all 4 sides. The last 2 sides would reveal other information related to the topic. An example would be to have the third side showing images from the news articles of the topic and the last side showing the geographic location of the topic.

Chapter 7
Conclusion

Our prototype system managed to organise news articles by topics. Using our client the user could get an overview of topics, answering the following questions:

- What has happened in the World?
- Is there any big events that I should know of?

By introducing a user profile it also answered the following question:

- Is there any interesting news?

The user will be able to see interesting news according to the user profile that describes his/her area of interest. Grouping news articles helps the user avoid flicking through tons of news articles that is related to the same topic. The user could for example just skip the topic android froyo because he/she is not interested and may not have to flick through tons of articles about android froyo. Using our surrogates it aided the user to quickly make a decision if a topic or news article is relevant or not. This let's the user gain knowledge of news in a quick way since he/she usually doesn't spend a lot of time with a mobile phone when on the move. In order to not show too much information at once for the user, we made use of 3D visualization to hide information in the overview. By having a 3D object the user saw the natural interaction of rotating the list for revealing additional information regarding a topic. 3D gives the ability to hide information and hint on possible interaction on a user interface. Although the result was mostly based on technology news sources our prototype system can handle an arbitrary set of RSS feeds regardless of its contents.

Chapter 8
Future Work

8.1 Community-based news articles

There are mainly two types of articles which is recommended for the user, news articles that reports about hot topics among various news sources, and news articles that matches the user profile. One more type of articles that might be interesting for the user is news articles which is voted by a community. These news articles may be funny or surprisingly interesting and maybe only published in one news source. This type of recommendation is already used in websites such as digg[1], reddit[2] and tweetmeme[3].

8.2 Server Push

The client application is periodically pulling data from server. This introduces unnecessary data traffic when no new feed items exist in the server. The server should instead push news to clients, much like pubsubhubbub [12]. Pubsubhubbub consist of a hub which fetches data from a server. This way multiple clients will get updates pushed from the hub. In similar idea the server could push data to our client application. This introduces more real time updates of news.

8.3 Geographic Location

News articles can also be related to geographic locations. This parameter can be used to calculate how geographically close an article's content is to the user's current location. The closer the location is to the user the more relevant it is for the user. In other words if a user lives in Sweden then news articles reporting about a hurricane in U.S. might not be so interesting for

[1] http://digg.com/
[2] http://www.reddit.com/
[3] http://tweetmeme.com/

the user. Irrelevant news is therefore also related to the location which the content is based on. A social aspect can be introduced into this. An example would be presenting news articles shared by friends which are within an area of the user's current location.

8.4 Timeline of Previous Topics

Another feature would be in case the user wouldn't read articles in a couple of weeks he would be able to navigate back in time and see the different topics in those time frames.

8.5 Track Interesting Topics

Current implementation relies on a fixed time frame and when that time frame is changed previously fetched news articles are treated as old and previously hot topics are forgotten. The user will instead see new topics in the new time frame. Instead of forgetting all previously hot topics, it might be interesting for a user to track certain hot topics over a longer time so that when new news articles belonging to that topic is published, the user will know what topic to go to get those news articles.

Appendix A

Data Processing System

A.1 Example of RSS and ATOM

A.1.1 RSS

```
<?xml version="1.0"?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd">
<channel>
<title>Engadget</title>
<link>http://www.engadget.com</link>
<description>Engadget</description>
<image>
<url>http://www.blogsmithmedia.com/www.engadget.com/media/
    feedlogo.gif</url>
<title>Engadget</title>
<link>http://www.engadget.com</link>
</image>
<language>en-us</language>
<copyright>Copyright 2010 Weblogs, Inc. The contents of this
    feed are available for non-commercial use only.</copyright>
<generator>Blogsmith http://www.blogsmith.com/</generator>
<item><title><![CDATA[Dell Streak car and AV docks now on sale,
    HDMI may or may not be included]]></title><link>http://www.
    engadget.com/2010/06/14/dell-streak-car-and-av-docks-now-on-
    sale-hdmi-may-or-may-not-be/</link><guid isPermaLink="true">
    http://www.engadget.com/2010/06/14/dell-streak-car-and-av-
    docks-now-on-sale-hdmi-may-or-may-not-be/</guid><comments>
    http://www.engadget.com/2010/06/14/dell-streak-car-and-av-
    docks-now-on-sale-hdmi-may-or-may-not-be/#comments</comments
    ><description><![CDATA[<div style="text-align: center;"><a
    href="http://www.engadget.com/2010/06/14/dell-streak-car-and
    -av-docks-now-on-sale-hdmi-may-or-may-not-be/"><img hspace
    ="4" border="0" vspace="4" src="http://www.blogcdn.com/www.
    engadget.com/media/2010/06/10x0614moin4trstreak.jpg" alt=""
    /></a></div>
Hey England, in need of some retail therapy after an
    unsatisfying sporting weekend? Dell's got the goods for you
```

with a rich new selection of cables, plugs, adapters, and cases for your new Streak. It looks like the HDMI dock we've been jonesing for has probably become a retail reality, but there's no way to be sure right now -- Dell's only listing an AV dock without going into the TV connectivity options. The fresh offerings also include the kickstand-equipped wallet case that we found intriguing, spare battery and sync cables, and an international travel kit. Of course, every silver lining has a cloud, which in this case is the pricing: nothing here costs less than £24 ($35) and the car- and TV-enabling docks will set you back a cool £55 ($80) each. Jumbo smartphone, jumbo accessory prices.

[Thanks, Kingsley] Dell Streak car and AV docks now on sale, HDMI may or may not be included originally appeared on Engadget on Mon, 14 Jun 2010 03:25:00 EDT. Please see our terms for use of feeds. <h6 style="clear: both; padding: 8px 0 0 0; height: 2px; font-size: 1px; border: 0; margin: 0; padding: 0;"></h6>Permalink | Dell<!--//--> | Email this | Comments]]></description><category>accessories</category><category>battery</category><category>car dock</category><category>car dock kit</category><category>car kit</category><category>CarDock</category><category>CarDockKit</category><category>CarKit</category><category>case</category><category>cases</category><category>dell</category><category>dell store</category><category>dell streak</category><category>DellStore</category><category>DellStreak</category><category>dock</category><category>dock kit</category><category>DockKit</category><category>hdmi</category><category>hdmi dock</category><category>HdmiDock</category><category>kickstand</category>

```
category><category>peripherals</category><category>streak</
category><category>wallet</category><dc:creator><![CDATA[
Vladislav Savov]]></dc:creator><pubDate>Mon, 14 Jun 2010
03:25:00 EDT</pubDate></item></channel></rss>
```

A.1.2 ATOM

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="eng">
<title type="text">Blog Name</title>
<subtitle type="text">Subsection</subtitle>
<id>http://anything.com</id>
<link rel="alternate" type="text/html" href="http://anything.com
"/>
<link rel="self" type="application/atom+xml" href="http://
anything.com/atom.feed"/>
<updated>2005-09-23T14:42:22Z</updated>
<rights>Copyright 2005</rights>
<entry>
<title type="text">Title</title>
<link rel="alternate" href="http://anything.com/2005/09/23/atom-
example"/>
<id>http://anything.com/2005/09/23/atom-example</id>
<author>
<name>Name of the author</name>
<email>mail@anything.com</email>
</author>
<published>2005-09-23T14:29:08Z</published>
<updated>2005-09-23T14:42:22Z</updated>
<summary type="xhtml">Summary</summary>
<content type="xhtml">
<div xmlns="http://www.w3.org/1999/xhtml">Content</div>
</content>
</entry>
</feed>
```

A.2 HTTP Request from Client

```
<history>
<username>user</username>
<url>http://feeds.nytimes.com/click.phdo?i=957490
aaf5051e75b732084666b6bf07</url>
<url>http://feeds.nytimes.com/click.phdo?i=
d86b5050840fed03c1d76109d5b00e24</url>
<url>http://feeds.nytimes.com/click.phdo?i=0
f46e59d02f669b560d69f3a09f1b37f</url>
<userprofile>
china,movie,intel,game,htc
</userprofile>
</history>
```

A.3 HTTP Response from Server

```
<?xml version="1.0" ?>
<rss version="2.0">
<channel>
<title>Ars Technica</title><group trendsstem="Google TV"
    numitems="28">
<summary><![CDATA[]]></summary><score>77.86365714</score>
<trendsrank>1.0</trendsrank><keyphrase><![CDATA[TV]]></keyphrase
    >
</group>
<item>
<title><![CDATA[
Android-based Google TV coming to living rooms this fall]]></
    title>
<link><![CDATA[
http://arstechnica.com/gadgets/news/2010/05/android-based-google
    -tv-coming-to-living-rooms-this-fall.ars?utm_source=rss&
    utm_medium=rss&utm_campaign=rss-20
]]></link>
<description><![CDATA[
<a href="http://arstechnica.com/gadgets/news/2010/05/android-
    based-google-tv-coming-to-living-rooms-this-fall.ars?
    utm_source=rss&amp;utm_medium=rss&amp;utm_campaign=rss">
  <img vspace="4" hspace="4" border="0" align="right" width
      ="230" height="129" src="http://static.arstechnica.com/
      assets/2010/05/googletv_drawing_ars-thumb-230x130-14132-f.
      png" />
  </a>
<p>Google has finally announced its <a href="http://arstechnica.
    com/gadgets/news/2010/04/google-reportedly-preparing-to-
    intro-tv-software-next-month.ars">long-rumored TV efforts</a
    > at Google I/O. Senior product manager Rishi Chandra said
    during the Thursday keynote that "video should be consumed
    on the biggest, best, and brightest screen in your house,
    and that's the TV," and that it hoped to combine the Web and
     TV-viewing experience in ways that others have yet to do.
    </p>...
]]></description>
<pubDate><![CDATA[2010-05-20 20:07:07]]></pubDate>
<category><![CDATA[
News, News, News, News, Gadgets, Open-source, Web, android,
    entertainment, googletv, intel, internet, internettv,
    logitech, sony, tv, video]]></category><userrank>0.1024</
    userrank><trendsstem>Google TV</trendsstem>
<novelty>0.9194849537037038</novelty></item></channel></rss>
```

A.4 Output of Topics with Related News Articles

Google TV 77.86		
Logitech's Google TV Box Will Have Special Powers (Google TV)	Gizmodo	2010-05-20 23:00:55
What Is Google TV? (Google TV)	Gizmodo	2010-05-20 22:20:50
Google TV Unveiled. It's All About The Ad Reach	Crunchgear	2010-05-20 20:21:02
Android-based Google TV coming to living rooms this fall	ArsTechnica	2010-05-20 20:07:07

TV 33.20		
Google unveils Android-powered TV platform	TechSpot	2010-05-20 20:07:00
Japan Promises 3D Holographic Broadcasts for 2022 World Cup [World Cup]	Gizmodo	2010-05-20 19:20:00
Google TV: It Is Real, Here Is Everything	I4U News	2010-05-20 19:05:48
Morning Edition: Google TV is Smart TV	CNET News.com	2010-05-19 15:59:06

Thethering 30.06		
Carriers Will Be Able To Decide Which Android Phones Have Tethering (And They Can Charge For It)	TechCrunch	2010-05-20 23:06:10
Froyo For Android: Tethering, Enterprise-Friendly, Handles More Monsters	TechCrunch	2010-05-20 19:05:46
iPhone OS 4 beta reveals AT&T tethering option	CNET News.com	2010-05-19 18:10:19
AT&T Tethering Option Spotted in Latest iPhone Beta (Digital Daily)	All Things Digital	2010-05-19 15:00:12

Beta 29.70		
iPhone OS 4.0 beta 4 hints at LED flash, camera for iPad, iPod, iPhone	Engadget	2010-05-19 21:33:00
Here's What's New In iPhone OS 4.0 Beta 4 [IPhone]	Gizmodo	2010-05-19 03:16:04
iPhone OS 4.0 beta 4 now available	Boy Genius Report	2010-05-19 02:25:10
Hark! iPhone OS 4 Beta 4 is here!	Boy Genius Report	2010-05-19 02:25:10

Froyo 24.27		
What Is Froyo? [Froyo]	Gizmodo	2010-05-20 22:20:30
Google Froyo: App Improvements, Music	I4U News	2010-05-20 18:23:03
Google lets the world enjoy FroYo	Boy Genius Report	2010-05-20 18:21:04
Google claims Froyo has the world's fastest mobile browser	Engadget	2010-05-20 17:56:00

Web Store 16.93		
Google Offers Up A Few More Details About The Chrome Web Store	TechCrunch	2010-05-20 10:31:01
Video: Google Chrome Web Store	Techradar	2010-05-20 09:49:00
Google previews Chrome OS usage with Web Store	CNET News.com	2010-05-20 00:30:00
Google announces Chrome web store for apps	Boy Genius Report	2010-05-19 19:14:40

Wave 14.75		
Samsung Wave hits Vodafone UK on June 1, free on Â£25 a month plans	Engadget	2010-05-20 14:44:00
In Depth: Google Wave: the beginner's guide	Techradar	2010-05-20 14:35:00
Google relaunches Wave, no invitation necessary	TechSpot	2010-05-19 20:00:00
Google Wave: now open to the public	CNET News.com	2010-05-19 19:05:53

Museum 10.54		
Mega Art Heist: Picasso, Matisse Stolen	FOXNews.com	2010-05-20 12:46:11
A tale of five small wind turbines (photos)	CNET News.com	2010-05-20 13:00:00
Police: Thieves steal Picasso, Matisse, 3 other paintings from Paris modern art museum	FOXNews.com	2010-05-20 11:58:14
The Snapping of Foucault's Pendulum [Oops]	Gizmodo	2010-05-20 04:20:00

stolen 7.33		
Precious artworks stolen in Paris heist	CNN.com - WORLD	2010-05-20 13:36:44
Prosecutors say artworks stolen from Paris museum are worth an estimated euro500 million	FOXNews.com	2010-05-20 12:46:29
US official: dozens of countries likely approve rules for return of property stolen by Nazis	FOXNews.com	2010-05-19 20:55:02

iPod Touch 7.05		
Twitter for iPhone / iPod touch downloads for free	I4U News	2010-05-20 08:30:00
iPod Touch with camera spotted in Vietnam	Boy Genius Report	2010-05-19 14:25:31
An iPod Touch With 2MP Cam Appears In Vietnam	TechCrunch	2010-05-19 13:51:10
iPod touch with camera leaked in Vietnam (video)	Engadget	2010-05-19 13:50:00

WebM 10.99		
Google launches WebM open-source video format	Boy Genius Report	2010-05-19 19:02:08
Google opens VP8 codec, aims to nuke H.264 with WebM	Ars Technica	2010-05-19 21:26:44
Steve Jobs Is Not Impressed With Google's New Video Format [Apple]	Gizmodo	2010-05-20 23:18:20

Troop 7.80		
Large numbers of troops and military vehicles gather in Bangkok near protest zone	FOXNews.com	2010-05-19 01:21:27
Troops, armored carriers advance near Bangkok protest	Reuters: World News	2010-05-19 02:16:46
Troops Move Against Protestors in Bangkok	ABC News: International	2010-05-20 10:44:24

iPhone OS 12.00		
Android OS Pulls Ahead of iPhone OS	I4U News	2010-05-10 17:53:59

amp 9.00		
SIM unlock now available for AT&T Palm Pre Plus	Engadget	2010-05-20 20:58:00
Quartet of Dell Streaks spotted in the wild in Seattle, testing for AT&T	Engadget	2010-05-20 16:33:00
Bang & Olufsen announces 40-inch BeoVision 8 LCD	Engadget	2010-05-20 15:31:00
HP G60t Intel Dual Core 2.1GHz 16" Laptop for $380 + $19 s&h	I4U News	2010-05-20 11:00:00

A.5 Sources included in our Test Corpus

A.5.1 World News

ABC News International
http://feeds.abcnews.com/abcnews/internationalheadlines

CNN.com - WORLD
http://rss.cnn.com/rss/edition_world.rss

FOXNews.com
http://feeds.foxnews.com/foxnews/world?format=xml

Reuters - World News
http://feeds.reuters.com/reuters/worldNews?format=xml

A.5.2 Technology News

All Things Digital
http://allthingsd.com/feed/

Ars Technica
http://feeds.arstechnica.com/arstechnica/index?format=xml

CunchGear
http://feeds.feedburner.com/CrunchGear

I4U News
http://feeds.feedburner.com/I4UNews

TechCrunch
http://feeds.feedburner.com/TechCrunch

Techradar
http://feeds.feedburner.com/techradar/allnews?format=xml

TechSpot
http://feeds.feedburner.com/techspot/news

Boy Genius Report
http://feeds.feedburner.com/TheBoyGeniusReport?format=xml

Gizmodo
http://feeds.gawker.com/gizmodo/excerpts.xml

Wired Top Stories
http://feeds.wired.com/wired/index?format=xml

CNET News.com
http://news.cnet.com/2547-1_3-0-20.xml?tag=txt

DailyTech
http://www.dailytech.com/rss.aspx

Engadget
http://www.engadget.com/rss.xml

Appendix B
Visualization System

B.1 Workshop Material

Figures B.1, B.2 and B.3 show the ideas from our workshop.

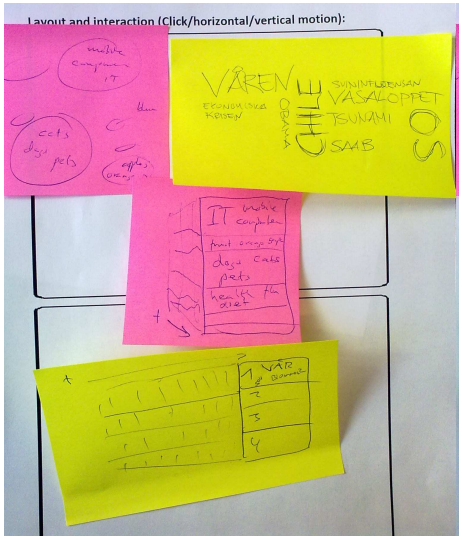

Figure B.1: First idea generated from the workshop

Figure B.2: Second idea generated from the workshop

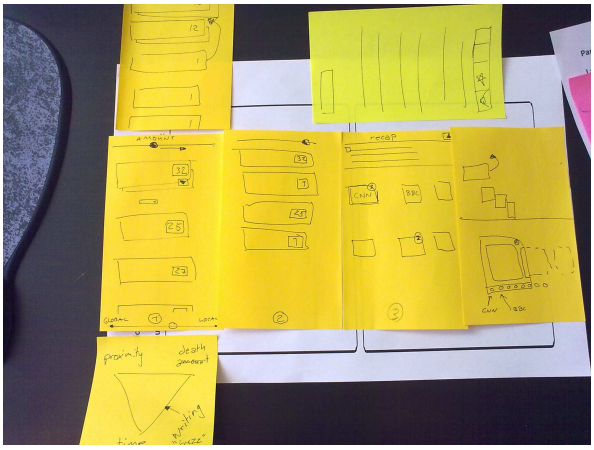

Figure B.3: Third idea generated from the workshop

B.2 Mockups

Figures B.4 and B.5 show some of our paper and pencil mockups.

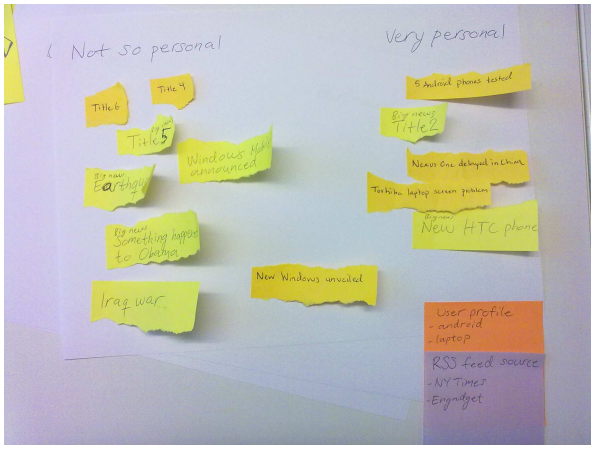

Figure B.4: A paper mockup of a 3D space

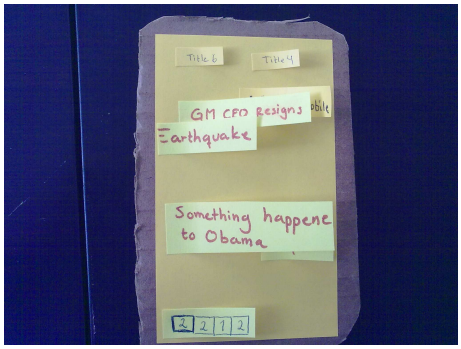

Figure B.5: A paper mockup of mobile view showing a part of a 3D space

B.3 Final Prototype Visualization

Figures B.6, B.7, B.8, B.9 and B.10 show our final prototype visualization. The visualization shown in the screenshots is based on dummy data.

Figure B.6: The top of the screen reveals information about unread news articles and time. As the list is scrolled down, that information will be hidden.

Figure B.7: The list can be rotated for revealing additional information about the topics.

Figure B.8: The other side of the list reveals titles of news articles related to the topic.

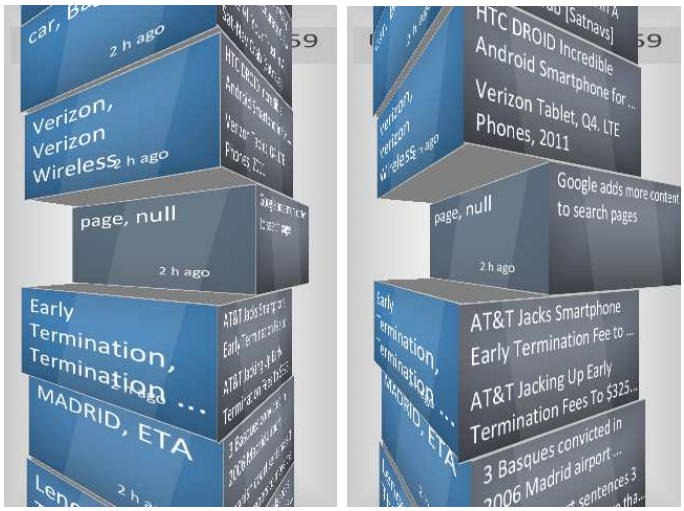

Figure B.9: The offset of a topic row informs about the novelty of the topic. Newer topics are closer to the screen and older are further away.

Figure B.10: As the list is scrolled down/up, it's rotated around the x-axis for increasing the visibility of topics further away.

References

[1] MySQL AB. Mysql. `http://dev.mysql.com/?bydis_dis_index=1`. [Online]. (2010, Jul.).

[2] S. Baumgärtner, A. Ebert, M. Deller, and S. Agne. 2D meets 3D: a human-centered interface for visual data exploration. In *CHI'07 extended abstracts on Human factors in computing systems*, page 2278. ACM, 2007.

[3] K. Bharat. And now, news. `http://googleblog.blogspot.com/2006/01/and-now-news.html`. [Online]. (2010, May).

[4] K. Bharat, T. Kamba, and M. Albers. Personalized, interactive news on the web. *Multimedia Systems*, 6(5):349–358, 1998.

[5] A. Biggs. Voyage. `http://rssvoyage.com/`. [Online]. (2010, Aug.).

[6] A. Blekas, J. Garofalakis, and V. Stefanis. Use of RSS feeds for content adaptation in mobile web browsing. In *Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility?*, page 85. ACM, 2006.

[7] K.K. Bun and M. Ishizuka. Topic extraction from news archive using TF* PDF algorithm. 2002.

[8] H. Chen, A.L. Houston, R.R. Sewell, and B.R. Schatz. Internet browsing and searching: User evaluations of category map and concept space techniques. *Journal of the American Society for Information Science*, 49(7):582–603, 1998.

[9] J.Y. Chen, C.A. Bouman, and J.C. Dalton. Hierarchical browsing and search of large image databases. *IEEE transactions on Image Processing*, 9(3):442–455, 2000.

[10] A. Cockburn and B. McKenzie. Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 203–210. ACM, 2002.

[11] Team Cooliris. Cooliris. `http://www.cooliris.com/`. [Online]. (2010, Aug.).

[12] B. Fitzpatrick and B. Slatkin. pubsubhubbub - a simple, open, webhook-based pubsub protocol and open source reference implementation. `http://code.google.com/p/pubsubhubbub/`. [Online]. (2010, Aug.).

[13] E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G. Nevill-Manning. Domain-specific keyphrase extraction. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 668–673. Citeseer, 1999.

[14] T. Fujiki, T. Nanno, Y. Suzuki, and M. Okumura. Identification of bursts in a document stream. In *First International Workshop on Knowledge Discovery in Data Streams*, pages 55–64. Citeseer, 2004.

[15] Dr. E. Garcia. Dimensionality reduction: Computing uk, sk, vk and vkt. `http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-4-lsi-how-to-calculations.html#reduction`. [Online]. (2010, Jul.).

[16] Dr. E. Garcia. Lsi keyword research and co-occurrence theory: A quantitative interpretation using co-occurrence. `http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-5-lsi-keyword-research-co-occurrence.html#quantitative`. [Online]. (2010, Aug.).

[17] Dr. E. Garcia. Svd and lsi applications. `http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-1-understanding.html#applications`. [Online]. (2010, Aug.).

[18] Dr. E. Garcia. Understanding svd and lsi: Background. `http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-1-understanding.html#background`. [Online]. (2010, Jul.).

[19] N. Glance, M. Hurst, and T. Tomokiyo. BlogPulse: Automated trend discovery for weblogs. In *WWW 2004 Workshop on the Weblogging*

Ecosystem: Aggregation, Analysis and Dynamics, volume 2004. Citeseer, 2004.

[20] S. Greene, G. Marchionini, C. Plaisant, and B. Shneiderman. Previews and overviews in digital libraries: Designing surrogates to support visual information seeking. *Journal of the American Society for Information Science*, 51(4):380–393, 2000.

[21] J. Haliburton and D. Gärdenfors. 3D Interfaces for Mobile Phones.

[22] J. Hicklin, C. Moler, and P. Webb. Jama 1.0.2. `http://math.nist.gov/javanumerics/jama/`. [Online]. (2010, Jul.).

[23] Google Inc. About google news. `http://news.google.se/intl/sv_se/about_google_news.html`. [Online]. (2010, May).

[24] Google Inc. Android sdk 2.1 platform. `http://developer.android.com/sdk/android-2.1.html`. [Online]. (2010, Jul.).

[25] Google Inc. Android sqlite database. `http://developer.android.com/reference/android/database/sqlite/package-summary.html`. [Online]. (2010, Jul.).

[26] Google Inc. Android xml utility methods. `http://developer.android.com/reference/android/util/Xml.html`. [Online]. (2010, Jul.).

[27] Google Inc. Android.com. `http://www.android.com/`. [Online]. (2010, Aug.).

[28] Google Inc. Google news. `http://news.google.com/`. [Online]. (2010, May).

[29] Google Inc. Google trends. `http://www.google.com/trends`. [Online]. (2010, Aug.).

[30] Google Inc. living stories - a new format for online news. `http://code.google.com/p/living-stories`. [Online]. (2010, May).

[31] Google Inc. Living stories experiment. `http://livingstories.googlelabs.com/`. [Online]. (2010, May).

[32] Sun Microsystems Inc. Java jdbc. `http://download-llnw.oracle.com/javase/6/docs/api/java/sql/package-summary.html`. [Online]. (2010, Jul.).

[33] Inc. Information Architects. Infographic - web trend map 4.0. http://informationarchitects.jp/wtm4/. [Online]. (2010, Aug.).

[34] M. Kamp. Newsrob - a google reader client. http://newsrob.blogspot.com/. [Online]. (2010, Aug.).

[35] D. Ken. *News Users 2009*. Outsell, 2009.

[36] J. Lowensohn. Browse the news in tags with zen news. http://news.cnet.com/8301-27076_3-10375909-248.html. [Online]. (2010, May).

[37] C.D. Manning, P. Raghavan, and H. Schütze. An introduction to information retrieval. pages 109–114,118–123, 2008.

[38] D. Megginson and D. Brownell. Java 1.6 sax. http://www.saxproject.org/. [Online]. (2010, Jul.).

[39] The University of Waikato. Keyphrase extraction algorithm. http://www.nzdl.org/Kea/description.html. [Online]. (2010, March).

[40] T.H. Ong, H. Chen, W. Sung, and B. Zhu. Newsmap: a knowledge map for online news. *Decision Support Systems*, 39(4):583–597, 2005.

[41] Object Technology International (OTI). Eclipse project. http://www.eclipse.org/. [Online]. (2010, Jul.).

[42] M. Parparita. Google reader trend - i like big charts and i cannot lie. http://googlereader.blogspot.com/2007/01/i-like-big-charts-and-i-cannot-lie.html. [Online]. (2010, May).

[43] M.F. Porter. An algorithm for suffix stripping. 1997.

[44] T. Ratschiller and M. Delisle. phpmyadmin. http://www.phpmyadmin.net/home_page/index.php. [Online]. (2010, Jul.).

[45] G. Rivera. Topics from techmeme on may 20th 2010. http://www.techmeme.com/100520/h2355. [Online]. (2010, Aug.).

[46] G. Robertson, M. Czerwinski, K. Larson, D.C. Robbins, D. Thiel, and M. Van Dantzich. Data mountain: using spatial memory for document management. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, page 162. ACM, 1998.

[47] Scintilla. Scintilla. http://scintilla.nature.com/. [Online]. (2010, Jul.).

[48] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. *The craft of information visualization: readings and reflections*, pages 364–371, 2003.

[49] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43, 2001.

[50] TAT. Tat cascades for android. `http://www.tat.se/site/products/cascades.html`. [Online]. (2010, Jul.).

[51] TAT. Tat motion lab. `http://www.tat.se/site/products/motionlab.html`. [Online]. (2010, Jul.).

[52] GIMP Development Team. Gimp 2.6. `http://www.gimp.org/`. [Online]. (2010, Jul.).

[53] Techmeme. Techmeme tech web, page a1. `http://www.techmeme.com/`. [Online]. (2010, Aug.).

[54] C. Warren. Zensify combines news visualization and social media on iphone. `http://mashable.com/2009/10/15/zennews/`. [Online]. (2010, May).

[55] M. Weskamp. projects / newsmap. `http://marumushi.com/projects/newsmap`. [Online]. (2010, May).

[56] M. Weskamp and D. Albritton. newsmap. `http://newsmap.jp/`. [Online]. (2010, Aug.).

[57] P. Willett. The Porter stemming algorithm: then and now. *Program: electronic library and information systems*, 40(3):219–223, 2006.

[58] I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, page 255. ACM, 1999.