Imperial College London

C401

MEng Computing Interim Report

News Aggregator and Summariser

Author:
Kunal M L Wagle

Student Number:
00814737

January 15, 2017

Contents

1 Introduction 4
1.1 The Project and its primary aims 4
1.1.1 News Aggregator 4
1.1.2 Duplication and Summarisation 4
1.2 Secondary Aims 5
1.2.1 Customisation 5
1.2.2 Challenging Viewpoints 6

2 Background Research 7
2.1 News Reading Habits 7
2.1.1 Digital News Report 2016 7
2.1.2 Potential User Survey 9
2.2 Related Products in the Market 13
2.2.1 Google News 13
2.2.2 Flipboard 15
2.2.3 Yahoo News Digest 16
2.2.4 Apple News 17
2.3 Machine Learning Techniques 18
2.3.1 Topic Modelling Techniques 18
2.3.2 Topic Labelling Techniques 19
2.3.3 Clustering Techniques 21
2.4 Summarisation Techniques 23
2.4.1 Extractive Summarisation Techniques 23
2.4.2 Abstractive Summarisation Techniques 24

3 Project Plan 30
3.1 What needs to be done 30
3.1.1 Machine Learning and Summarisation aspects 30
3.1.2 Back End Server 30
3.1.3 Front end application(s) 30
3.2 Milestones and fallbacks 31
3.2.1 Machine Learning and Summarisation aspects 31
3.2.2 Back End Server 31
3.2.3 Front end application(s) 31
3.3 Road Map 32

4 Evaluation Plan 33
4.1 Evaluating the summarisation 33
4.2 Evaluating the front end 34

Bibliography **35**
Articles 35
Books and Reports 36
Webpages 36
Miscellaneous 37

Index **38**

List of Figures

2.1 Why people use News Aggregators 7
2.2 Concerns with personalisation of News Feeds 8
2.3 What is a good way to get to the news? 8
2.4 Survey graphs surrounding how often news is read and how many sources are used 10
2.5 Survey graphs regarding article lengths 11
2.6 Survey graphs concerning summarisation of articles 12
2.7 Survey Graphs regarding searching for topics in the news 12
2.8 Survey Graphs about News Digests 13
2.9 Screenshots from the Google News website 14
2.10 Screenshots from the Flipboard App 15
2.11 Screenshots from the Yahoo News Digest app 16
2.12 Screenshots from the Apple News app 17
2.13 An example of a Dependency Tree 25
2.14 A concept created during multimodal semantic summarisation 27
2.15 A flowchart showing the process of semantic graph summarisation . . 28

1 Introduction

In a fast moving world it's more important now than ever to stay on top of the news. However, people are often faced with their own questions when they're searching for news: Where should I get it from? How many sources should I use to ascertain the facts about a piece of news? Can I trust this source and what it is saying?

Coupled with these questions is a study by the Statistic Brain Research Institute[29] that suggests that the average attention span of a person has dropped significantly over the last five years[22]. There is arguably more news and more sources to read news from, but even less time in which to read it all.

My project aims to resolve some of these issues, by creating a new platform that aggregates news from a variety of different news sources, merges articles about the same specific piece of news, and summarises them. This would thus provide news for readers in short pieces, allowing a user to cover more of the news than ever before.

1.1 The Project and its primary aims

The project would be a news aggregator and summariser. More details on each are provided below.

1.1.1 News Aggregator

The first aim of the project is to create a news aggregator. One of the key differentials between this and the current news aggregators on the market (such as Flipboard[13]) would be that with my project, users will be able to search for news related to much more specific topics than currently offered.

1.1.2 Duplication and Summarisation

With a key issue for users now being the time factor, an aim of the project would be to reduce the time needed for a user to surmise the key issues and facts of a news story. To do this, there are two key issues - the sources for the articles, and the length of the articles.

Article Sources

There are potentially hundreds of different sources for articles on a single piece of news. For example, if I were to search Google News[19] (on the 22nd December 2016) for "Uber" I would be greeted with articles about the fact that Uber[33] have been forced to withdraw driverless cars from the roads of San Francisco. Upon further examination, it becomes clear that Google News has a similar story from as many as 50-60 sources. But which one should we read from?

One of the main aims of the project would be to identify "duplicate" articles (articles that are about the same story) and merge them into one article for the user to read. However, this could of course raise the issue that articles suddenly become too long for a user to read. This is addressed below.

Article Length

An issue with some articles is the length. In July 2016 a study was conducted by the Statistic Brain Research Institute[29] that said that the average attention span of a person is now as low as 8.25 seconds[22]. In addition to this, the Institute's findings also said that the average person only reads 28% of a webpage of average length (593 words). Another key finding was that users spend only 4.4 seconds to cover each additional 100 words on a webpage.

With this in mind, it becomes clear that a key problem with articles is that they're too long for the average user. If a key point is not mentioned in the article until near the end, it's more than possible that the user could miss it entirely. As a result, a key aim of this project must be to try and counteract this issue.

The best way to do this is to find a way to summarise the article so that the users can see the key facts at a glance. The project would aim to automatically summarise the merged articles found by the duplication algorithm. This would be a key differential between my project and news aggregators on the market.

1.2 Secondary Aims

1.2.1 Customisation

On a personal level, something I've noticed in the past about alternative news aggregators available is that it's not easy in many cases to tailor the sources that you get your news from. To take an extreme example, someone who normally reads *The Guardian*[31] is unlikely to want to get their news from *Fox News*[14], as they are on opposite ends of the political spectrum.

As a result, a secondary aim of the project would be to allow a user to customise the sources that they receive their news from. One potential way to do this would be to present the user with a list of news sources that the project will source news from, and allow them to remove selections as necessary. The article could then be summarised so that only news from those sources is present in the final version presented to the users.

However, as described in Section 1.2.2, this could have its own pitfalls, despite the obvious benefits.

1.2.2 Challenging Viewpoints

One of the more interesting findings in the 2016 Digital News Report[27] (conducted by the Reuters Institute for the Study of Journalism[28]) was that some were concerned about the potential impact of personalising their news feeds. They felt that it could lead to both "missing information" and "missing challenging viewpoints". A secondary aim of the project would be to find a way to rectify this misgiving. This could potentially be done in different ways, from providing summarisation of articles from alternate sources of the article, or by simply posting a link to articles about the same topic from sources considered to be on the opposite end of the political spectrum.

2 Background Research

2.1 News Reading Habits

2.1.1 Digital News Report 2016

Conducted by the Reuters Institute for the Study of Journalism[28] in Oxford University[35], the Digital News Report[27] is an annual study that serves to investigate how people access and find out about news in various countries across the globe, including the United Kingdom.

The study attempts to cover, amongst other things how people get the news, how they use social media, what types of news they trust, and both the reasons to use and concerns about news aggregators. Findings of the report relevant to my concept are listed below:

Reasons to use News Aggregators

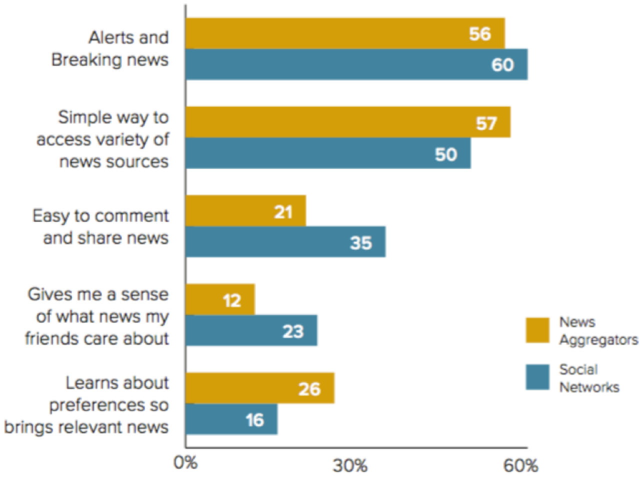


Figure 2.1: A graph showing why people use News Aggregators and Social Networks as news sources

As shown in the graph in Figure 2.1 the key to the popularity of a News Aggregator is that it's simplicity and speed. It's important to bring news as soon as it happens, and it needs to provide access to a variety of different news sources. Of lesser importance is the social aspect - the need to comment and share the news, although the fact that 26% of people want their preferences to influence the news that they're interested in should not be discounted.

Concerns about Personalised News

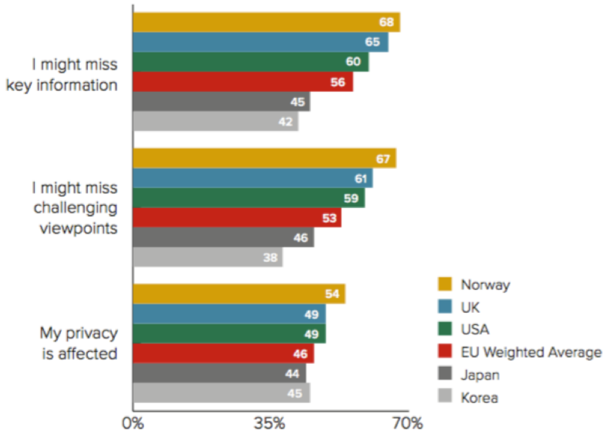


Figure 2.2: A graph showing the main three concerns that people have with Personalised News Feeds determining what they read.

Key concerns, especially in the United Kingdom, with Personalised News is that information might be incomplete. It is not difficult to see why this may be a concern - if you limit your news to a subset of the sources that might be available then it's possible to miss parts of the story - the parts that challenge aspects of the articles you're reading. It's therefore important that the project reflects this concern, which is why it is listed as a secondary aim (See section 1.2.2).

What should determine what someone reads next

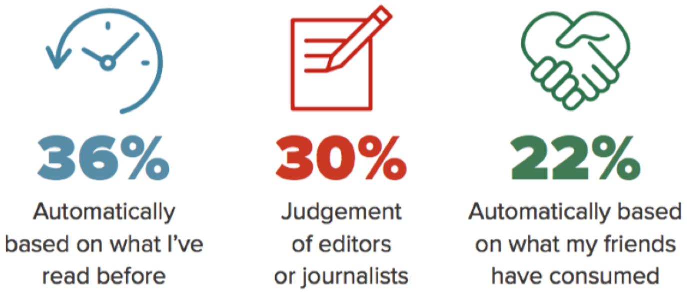


Figure 2.3: Key statistics in response to the question: Is this a good way to get the news?

Least popular are algorithms that present news based on what people's friends have been reading. In general, according to the Reuters Institute[28], "People think *they* are the best judge of what's important to them." More interesting, is that people are now less inclined to allow journalists or editors to push articles to the top of reading lists than having an algorithm suggest similar stories to what they've just consumed. This could perhaps indicate that they aren't as trusting in journalists and editors as before - something that the Reuters Institute touches upon in an essay near the end of the report.

Conclusions

The Reuters Institute for the Study of Journalism is a trusted and respected organisation, and this study has been supported by institutions varying from media outlets such as the BBC[11], news search engines such as Google, and other Universities, such as University of Canberra[34]. Importantly, it's also supported by regulators, including Ofcom[26], which is the regulator in the United Kingdom.

The survey was conducted on their behalf by YouGov[39], which is a highly respected polling company in the United Kingdom, and it surveyed more than 50,000 people, including over 2000 in the United Kingdom. In addition, the report is coupled with numerous essays on various key points that arose from the study. The writers of these essays varied from the Director of Research at the Reuters Institute to the Chief Executive Officer of The New York Times[32].

As a result of this, I deemed the Digital News Report 2016[27] to be a very reliable source of information with regards to how people read the news, and their preferences in the field.

2.1.2 Potential User Survey

Following this initial research I decided to conduct a survey of potential users to delve into more specific aspects of News Reading Habits. I was more focused on the number of sources people use to ascertain the facts about a piece of news. I also asked questions regarding the summarisation of news and news digests.

I was interested in the questions about summarisation and digests after reading some data from a study conducted by the Statistic Brain Research Institute[29]. Statistic Brain conducted a study in July 2016 that suggested that the average attention span of a human is now only 8.25 seconds[22]. They coupled the presentation of this data with statistics about people browsing the internet taken from the paper *Not Quite the Average: An Empirical Study of Web Use*[36]. The statistics suggest that the average user only reads 28% of the words on the average webpage, and that for each additional 100 words beyond that, the user would only spend an additional 4.4

seconds on the page.

Admittedly, the paper was written in 2008, and so might not reflect average web browsing activity, but it raised a question worth answering in my opinion. Do users actually read full articles on the news, and if not, would summarising articles help make sure they didn't miss out on key points, thus counteracting an issue discovered in the Digital News Report 2016[27] (Section 2.1.1)?

The survey was presented on Google Forms[18] and was answered by 96 respondents. Further findings are shown below:

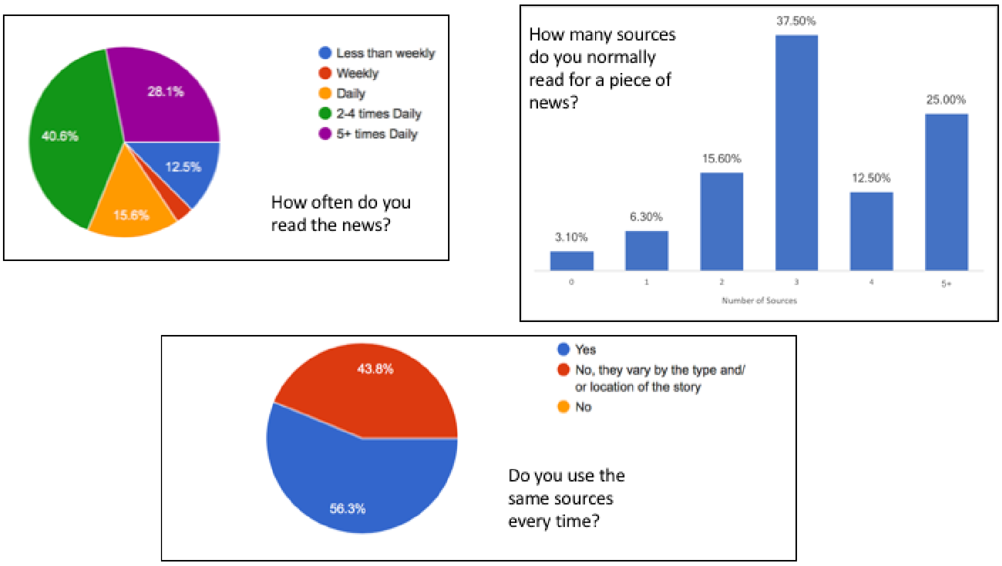


Figure 2.4: Top Left: A pie chart showing how often people read the news. Top right: A bar graph showing how many sources people normally use for a source of news. Above: A pie chart showing whether people use the same sources every time.

The first questions (Figure 2.4) on the survey were designed to ascertain how often people read the news, and how many sources they use. Overall, well over 80% of respondents said that they read the news at least daily, and nearly 60% more than once per day. In terms of sources, it was quite rare for someone to use only one source for a piece of news. In fact, over 90% said that they used two or more sources, with three and five sources being the most prominent selections. The respondents were a lot more split on the question of consistency of sources, although more than half said that they used the same sources every time, with 43.8% saying that they vary their sources based on the type and/or location of the news story.

I further expanded upon the initial reading I had done about attention spans by asking potential users how much of articles they normally read. Only 18.8% of

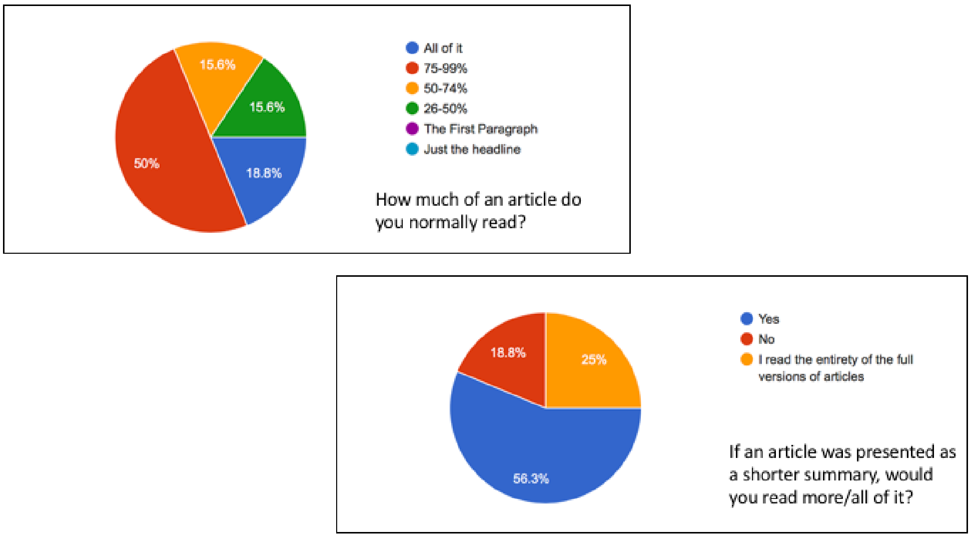


Figure 2.5: Top Left: A pie chart showing how much of articles people normally read. Bottom right: A pie chart showing what percentage of people would read a shorter summary of an article.

respondents said that they read the entirety of articles (Figure 2.5), and as much as 15.6% confess that they read less than 50% of an article on average. It's plausible that the users could be missing key facts through not reading the entire article. On reflection, I could have also asked if this was a concern to those users.

Another question asked was whether reading a shorter summary would result in users reading more or all of the article. Encouragingly, more than half (56.3%) of respondents said that they would read a shorter summary.

With these questions (Figure 2.6) I aimed to obtain an idea from potential users as to whether combining articles about the same topic from different sources and then summarising it would be considered useful. The question got a resounding response, with as much as 87.5% saying that they would read a summary like this. Consolidating this was the fact that 81.3% said they would use a News Aggregator that summarised articles. Amongst those who said they wouldn't, there were comments along the lines of "I'm not great with technology" given as explanation. An interesting comment however, said that "nuance could be lost in the summarisation". This is a valid point, and so a key point of the evaluation process will have to be focused on the summarisation of articles itself, to ensure it's not losing important information at any stage.

For the questions in Figure 2.7 I tried to get an idea of how much people search

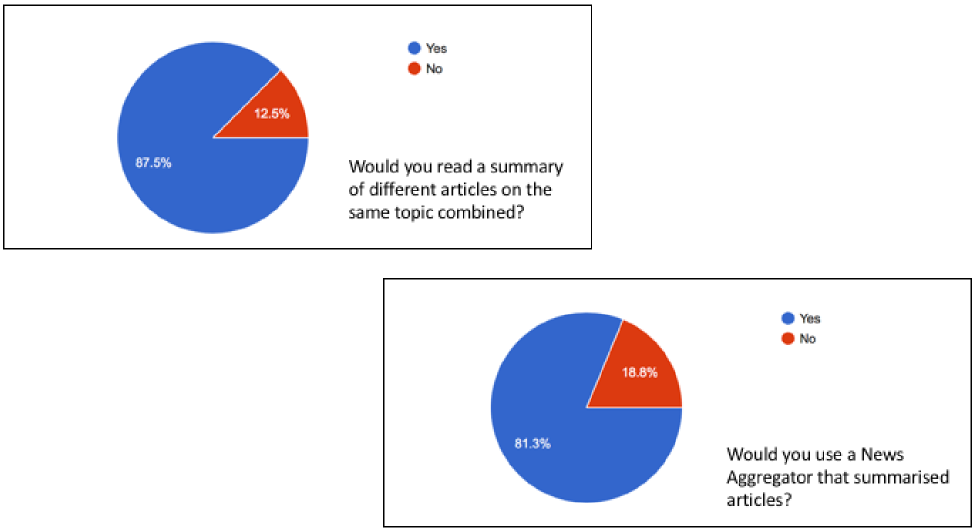


Figure 2.6: Top Left: A pie chart showing whether people would read a summary of different articles on the same topic combined. Bottom right: A pie chart showing answers to the question: "Would you use a News Aggregator that summarised articles"

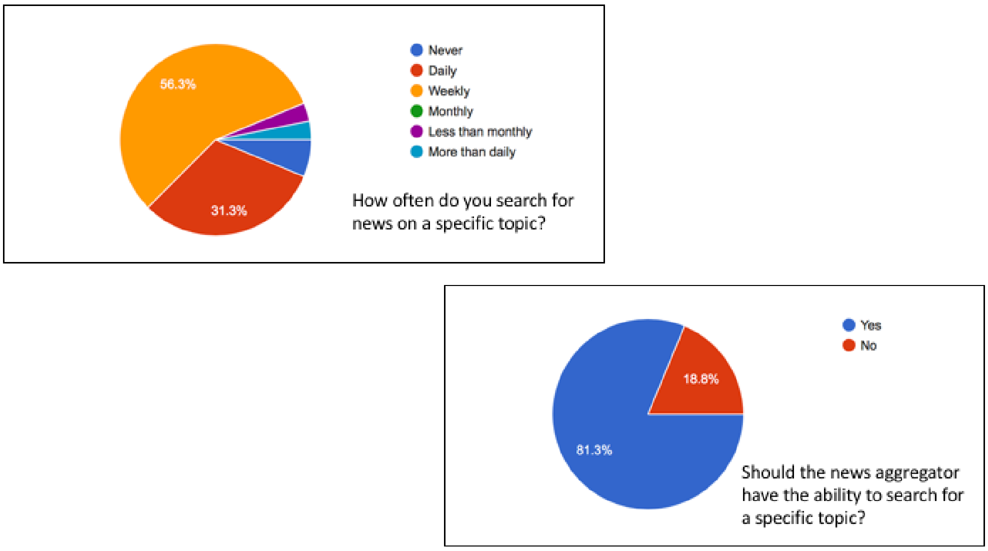


Figure 2.7: Top Left: A pie chart showing how frequently people search for news on a specific topic. Bottom right: The answers to a question asking if the News Aggregator should have the ability to search for a specific topic.

for news on a specific subject of interest to them. In general, this came out to be less frequent then reading the news itself, with just over half the respondents saying that they search for a specific topic weekly. However, a healthy proportion (31.3%) search daily for specific topics, and some search even more often than this. As much as 81.3% of respondents agreed that the News Aggregator should have a function to search for specific topics.

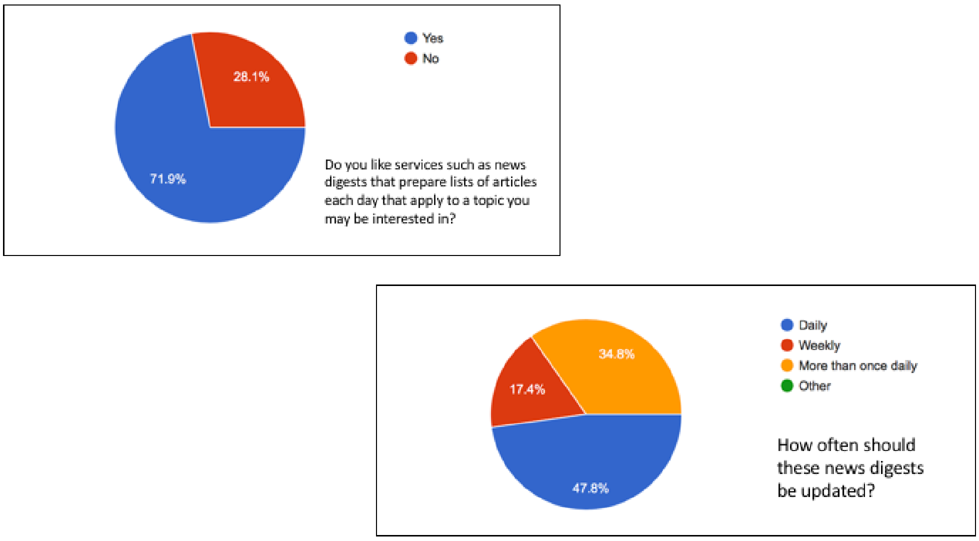


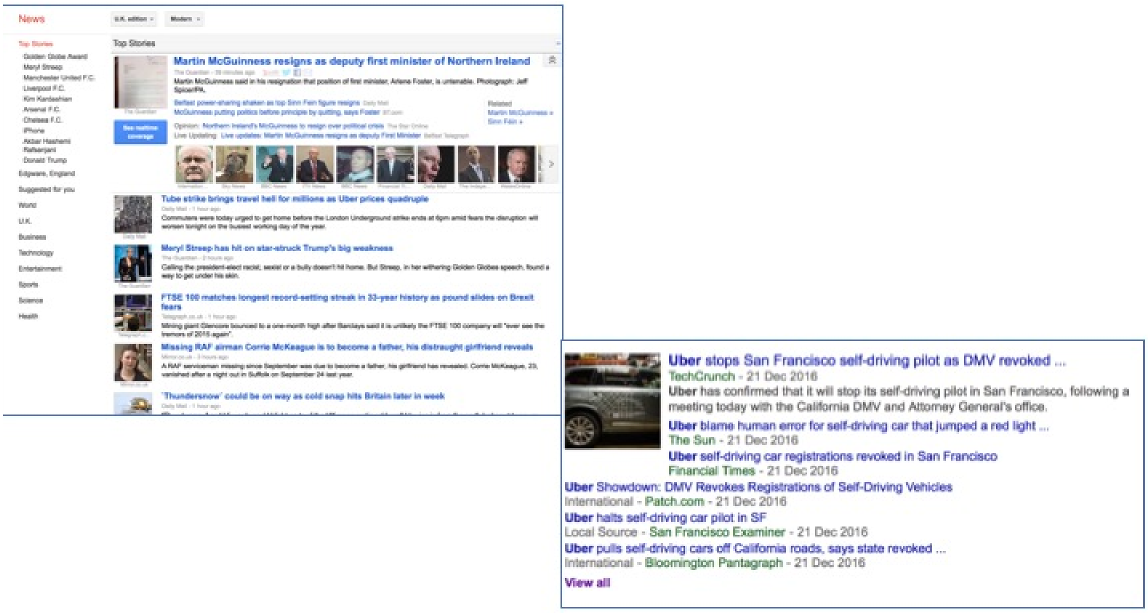
Figure 2.8: Top Left: A pie chart showing people's attitudes towards News Digests. Bottom right: Responses regarding how frequently news digests should be updated.

The last questions in the survey, shown in Figure 2.8 centred around News Digests. Nearly 72% of respondents said that they liked services that provide news digests. They in general felt that these should be updated at least daily (47.8%), with a further 34.8% on top of that saying it should be updated more than once.

2.2 Related Products in the Market

2.2.1 Google News

Initially developed early in the century and released in 2006, Google News[19] is a free-to-use news aggregator. Google News operates in a similar manner to the traditional Google[17] search engine, thus making it a go to aggregator when searching for a specific topic. Google News also groups "similar" articles together using Clustering techniques[1]. Google News operates purely as a go-between - when clicking



Top Stories

Martin McGuinness resigns as deputy first minister of Northern Ireland
The Guardian - 39 minutes ago
Martin McGuinness said in his resignation that position of first minister, Arlene Foster, is untenable. Photograph: Jeff Spicer/PA.
Belfast power-sharing shaken as top Sinn Fein figure resigns Daily Mail
McGuinness putting politics before principle by quitting, says Foster RT.com
Opinion: Northern Ireland's McGuinness to resign over political crisis The Star Online
Live Updating: Live updates: Martin McGuinness resigns as deputy First Minister Belfast Telegraph

Tube strike brings travel hell for millions as Uber prices quadruple
Daily Mail - 1 hour ago
Commuters were today urged to get home before the London Underground strike ends at 6pm amid fears the disruption will worsen tonight on the busiest working day of the year.

Meryl Streep has hit on star-struck Trump's big weakness
The Guardian - 2 hours ago
Calling the president-elect racist, sexist or a bully doesn't hit home. But Streep, in her withering Golden Globes speech, found a way to get under his skin.

FTSE 100 matches longest record-setting streak in 33-year history as pound slides on Brexit fears
Telegraph.co.uk - 1 hour ago
Mining giant Glencore bounced to a one-month high after Barclays said it is unlikely the FTSE 100 company will "ever see the tremors of 2015 again".

Missing RAF airman Corrie McKeague is to become a father, his distraught girlfriend reveals
Mirror.co.uk - 3 hours ago
A RAF serviceman missing since September was due to become a father, his girlfriend has revealed. Corrie McKeague, 23, vanished after a night out in Suffolk on September 24 last year.

'Thundersnow' could be on way as cold snap hits Britain later in week
Daily Mail - 1 hour ago

Uber stops San Francisco self-driving pilot as DMV revoked ...
TechCrunch - 21 Dec 2016
Uber has confirmed that it will stop its self-driving pilot in San Francisco, following a meeting today with the California DMV and Attorney General's office.
Uber blame human error for self-driving car that jumped a red light ...
The Sun - 21 Dec 2016
Uber self-driving car registrations revoked in San Francisco
Financial Times - 21 Dec 2016
Uber Showdown: DMV Revokes Registrations of Self-Driving Vehicles
International - Patch.com - 21 Dec 2016
Uber halts self-driving car pilot in SF
Local Source - San Francisco Examiner - 21 Dec 2016
Uber pulls self-driving cars off California roads, says state revoked ...
International - Bloomington Pantagraph - 21 Dec 2016

Figure 2.9: Left: A section of the homepage for Google News.[19] Right: Google News uses clustering techniques to group articles about the same topic together.

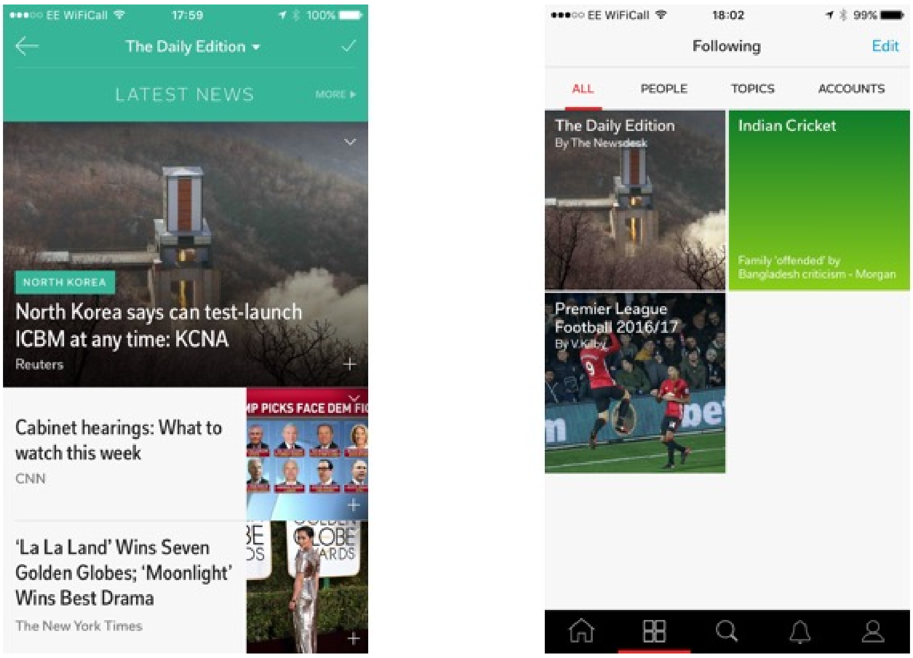
on an article the user is taken straight to the media source itself, rather than being able to read the article on Google News itself.

What it does well

- Groups similar articles together
- Similar to a traditional search engine and so is easy to use
- Obtains articles instantly, thus providing most up-to-date information when searching

What it doesn't do well

- Doesn't host articles itself - therefore making it harder to navigate than perhaps could be possible. Although, this could be to avoid any copyright issues.



North Korea says can test-launch ICBM at any time: KCNA
Reuters

Cabinet hearings: What to watch this week
CNN

'La La Land' Wins Seven Golden Globes; 'Moonlight' Wins Best Drama
The New York Times

The Daily Edition
By The Newsdesk

Indian Cricket
Family 'offended' by Bangladesh criticism - Morgan

Premier League Football 2016/17
By V.Kilby

Figure 2.10: Screenshots from the Flipboard App.[13] Left: All users are subscribed to *The Daily Edition*, which puts together the latest most popular news. Right: Users can subscribe to "magazines" that they may be interested in.

2.2.2 Flipboard

Flipboard[13] is a much more recent attempt at a news aggregator (developed in 2010), and relies on the concept of users subscribing to "magazines" on different topics. There's a central "cover page" on the home page that shows the most recent stories from across all a user's subscriptions. Like Google News, links from the desktop website send the user to the original media source, while links from within the mobile applications open a browser within the app itself.

What it does well

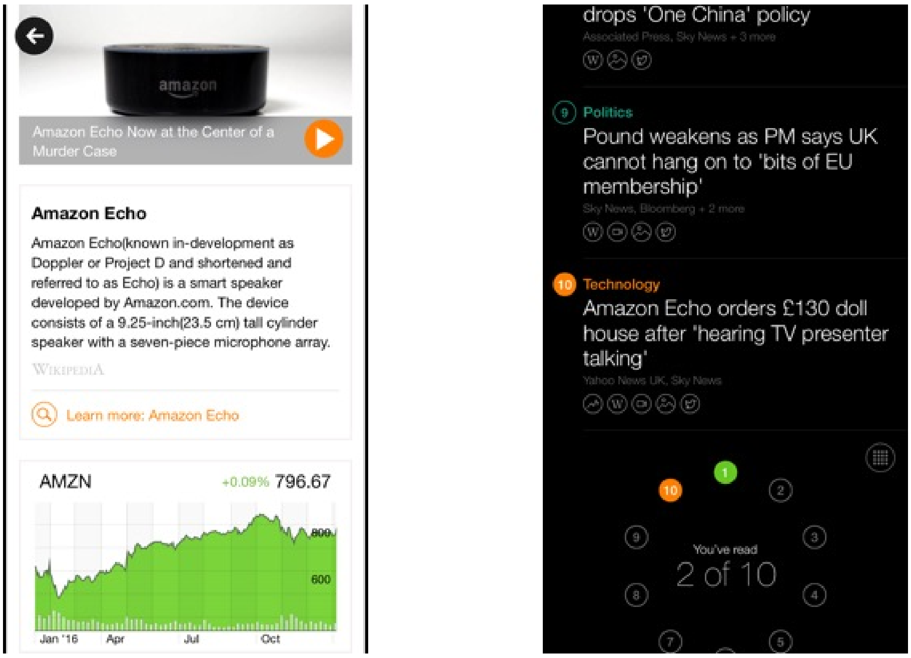
- Provides a home page that allows a user to see the most popular stories at the time related to the user's subscriptions.
- Available as a mobile application

What it doesn't do well

- Topics are much broader than Google News, thus meaning that users can't necessarily search for topics specific-enough for them.

- Flipboard requires registration before being able to read articles

2.2.3 Yahoo News Digest



Amazon Echo Now at the Center of a Murder Case

Amazon Echo

Amazon Echo(known in-development as Doppler or Project D and shortened and referred to as Echo) is a smart speaker developed by Amazon.com. The device consists of a 9.25-inch(23.5 cm) tall cylinder speaker with a seven-piece microphone array.

WIKIPEDIA

Learn more: Amazon Echo

AMZN +0.09% 796.67

drops 'One China' policy

Associated Press, Sky News + 3 more

9 Politics

Pound weakens as PM says UK cannot hang on to 'bits of EU membership'

Sky News, Bloomberg + 2 more

10 Technology

Amazon Echo orders £130 doll house after 'hearing TV presenter talking'

Yahoo News UK, Sky News

You've read 2 of 10

Figure 2.11: Screenshots from the Yahoo News Digest app[38]. Left: The Yahoo News Digest app provides links and infographics that are relevant to the article. Right: A section of the home screen, which displays the top ten stories for the day.

Yahoo News Digest[38] is a direct evolution from Summly[30], which was an app that summarised news. Yahoo News Digest is a phone application that creates two digests a day: one in the morning and one in the evening. Each digest contains the ten leading articles from the previous 12 hours. Each article provides a summary of the story and links to relevant other pages - such as articles from Wikipedia[37].

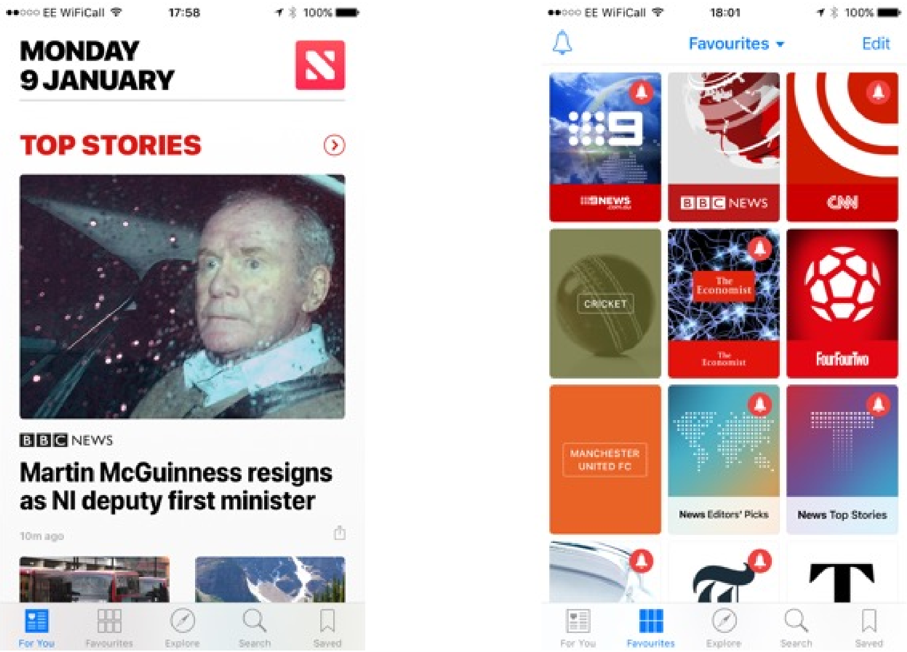
What it does well

- Yahoo News Digest won the 2014 Apple Design award[2].
- The articles also provide key infographics, quotes, and other information potentially relevant to the article.

What it doesn't do well

- There are only ten articles available per digest, and no capability for searching by topic.
- Digests are only produced in the morning and the evening, so the news articles presented could be out of date.

2.2.4 Apple News



MONDAY 9 JANUARY

TOP STORIES

Martin McGuinness resigns as NI deputy first minister

10m ago

Figure 2.12: Screenshots from the Apple News app[10]. Left: The home page of the app. Right: Apple News allows users to select sources and topics to be their favourites.

Apple News[10] is an application that is installed by default on all recent iOS devices. Apple's default attempt at a news aggregator allows users to select their preferred news sources, and from a selection of topics. Apple[9] then presents on a home screen news from those sources and topics. Clicking on each article keeps it in the native application, rather than sending the user to the media source itself.

What it does well

- Allows selection based off both topics and the news sources themselves

- The home screen for the app allows users to see a lot of headlines at a glance

What it doesn't do well

- Navigation on the application is not simple. If a user has accessed many articles from push notifications, then the user could have to press the back button several times to get back to the home screen.
- The topics that a user can subscribe to can be quite limited, and aren't reactive to current affairs - for example, if a natural disaster occurs, you couldn't then subscribe to that natural disaster as a topic.

2.3 Machine Learning Techniques

2.3.1 Topic Modelling Techniques

Topic modelling is a subsection of Machine Learning that aims to determine what topic a given document is about. The topics wouldn't be named at this stage, they would simply be given generic names such as Topic A and Topic B. Assigning names to topics will be done at a later stage (see Section 2.3.2).

Latent Semantic Indexing

Also known as Latent Semantic Analysis[3], Latent Semantic Indexing (LSI) was one of the initial forerunners in the field of topic modelling. It uses singular value decomposition to locate patterns in the text of a document and thus form a basis on which to categorise the document. A major benefit of LSI is that it is fast to train, but in general it has lower accuracy when compared to models that are probabilistic, such as Latent Dirichlet Allocation[4].

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probablistic model that assumes that the topic distribution has a Dirichlet prior. The theory behind LDA is that a document of words contains a mixture of different topics, and this is reflected in the final answers given for the algorithm.

A rough algorithm for performing LDA[5]:

1. **Set n to be the number of topics there are in the document.** We can do this by trial and error.

2. **Assign every word *w* in the document d to a random topic.** These topics are temporary. At this stage we can remove function words, such as "the". However, we keep duplicates - in fact, at this stage they could be in different topics.

3. **Check and update topic assignments.** To do this, we loop through each word in the document, taking note of how prevalent the word is across documents, and prevalent those topics are in the document. These two probabilities are then passed to a sampling algorithm to generate a new topic for the word. This step is usually completed using the statistical model *Gibbs Sampling*.

4. **Repeat step 3 until there are no more topic-reassignments**.

2.3.2 Topic Labelling Techniques

Topic Labelling techniques in the project for labelling the topics that are generated from the LDA analysis of the document in Section 2.3.1. When conducting research into this topic specifically, I found a paper (*Automatic Labelling of Topic Models* by Lau, Grieser, Newman and Baldwin in 2011[25]) that documents the creation of an algorithm that labels topics using Wikipedia[37] titles. This could work very well in my project, as Wikipedia titles as title headings would allow users to search for topics that are both broad and specific.

A rough version of Lau, Grieser, Newman and Baldwin's algorithm:

1. **Calculate the top 10 topic terms.** This is done by finding the marginal probabilities of each word from the original topic models, and taking the top ten. The marginal probability of a term is the probability of that term being randomly selected given a topic *t*.

2. **Search Wikipedia using these terms.** We also search Google[17] using a site restricted search (to Wikipedia) and take the top eight results from each. These are called the *primary labels*.

3. **Isolate all "noun chunks" from the terms.** In this case noun chunks are combinations of words from the terms that appear next to each other. For example, with the term "Summer Olympic Games" the noun chunks would be "Summer", "Olympic", "Games", "Summer Olympic", "Olympic Games" and "Summer Olympic Games". Note that "Summer Games" is not a noun chunk as the words don't appear juxtaposed. These noun chunks are added

to the primary labels from step 2 and are deemed *secondary labels.*

4. **For each noun chunk:**
   - Check to see if the noun chunk is the title for a Wikipedia article
   - Remove the noun chunk if it doesn't correspond to a Wikipedia article

5. **Calculate the *Related Article Conceptual Overlap* scores.** Related Article Conceptual Overlap (RACO), developed by Grieser et al in 2011[20], is a calculation designed to identify the strength of relationship between terms by inspecting the category overlap between the terms' corresponding articles. We do this for each secondary label still remaining - details on how to calculate the RACO scores are explained in further detail below.

6. **Discard all secondary labels with RACO score of less than 0.1.**

7. **Add five highest topic terms to the list.** Now we return to the original list of topic terms from step 1 and add the five highest to the remaining candidates.

8. **Perform candidate ranking.** There are multiple ways to do this, but the original paper recommends using a variety of statistical methods, based around a T-test, the Chi-squared test and a log-likelihood test. The aim is to estimate how closely related the candidate is to all the terms in the topic. We then take the top candidate as our final answer.

Calculating the *Related Article Conceptual Overlap*

Related Article Conceptual Overlap (RACO) was first introduced as a concept in the paper *Using Ontological and Document Similarity to Estimate Museum Exhibit Relatedness*[20] by Grieser, Baldwin, Bohnert and Sonenberg in 2011. It compares two terms and estimates their similarity by comparing the two terms' similarity on Wikipedia. The core calculation for RACO goes as follows:

$$Category-Overlap(a,b) = \left| \left(\bigcup_{p \in O(a)} C(p) \right) \bigcap \left(\bigcup_{p \in O(b)} C(p) \right) \right|$$

In this equation, *O(a)* represents the outlinks (the links to other articles from the

Wikipedia article) of an article *a* and *C(p)* represents the set of categories that article *p* is a part of.

An issue with the Category-Overlap calculation as it is, is that there's a bias for articles that are larger than others as they will have more outlinks but won't necessarily be in more categories. As a result it's normalised using Dice's coefficient to produce the final RACO equation:

$$sim_{RACO}(a,b) = \frac{2 \times \left| \left(\bigcup_{p \in O(a)} C(p) \right) \bigcap \left(\bigcup_{p \in O(b)} C(p) \right) \right|}{\left(\bigcup_{p \in O(a)} C(p) \right) + \left(\bigcup_{p \in O(b)} C(p) \right)}$$

2.3.3 Clustering Techniques

Cluster analysis is a machine learning technique that is used to put items that are similar to each other into groups. In practice it's used by Google News[19] to put articles about the same topic together for a user[1]. There are multiple commonly used types of cluster analysis:

Centroid Clustering

In Centroid Clustering[6], also known as k-means clustering, there are *k* clusters. A vector is calculated for each article in the list. The article is then assigned to the cluster that is closest to it's vector score. A major downside to this method however is that it requires *k* to be defined in advance. In the context of this project, that's not applicable as we don't know how many different articles we are clustering based on.

Density Clustering

Density Clustering[6] also involves calculating a vector score for each article. Once these have been calculated, they can be graphed, and the areas of the graph that have highest density are chosen as the clusters. An advantage of this over the centroid clustering methods is that we don't need to know the number of clusters beforehand. However, density clustering can become less accurate as it requires areas of sparse density on the graph to precisely separate the different groups, which isn't always possible.

Hierarchical Clustering

Hierarchical Clustering[7], which is also known as Connectivity Clustering, is based on the idea of using distance measures between articles to identify which ones are most similar. There are two approaches to Hierarchical Clustering:

- *Agglomerative*, which is a bottom-up approach, assigns each item to its own cluster and then merges pairs that are closer together.
- *Divisive*, a top-down approach, that begins with all items in a single cluster, and then proceeds to split it into multiple clusters.

Creating a vector score for each item

The first step in each of the three clustering techniques is to create a vector score for each item. As this will play a big part in the final results, it's important to get this stage right. This can be split into two steps:

1. **Find definitive terms within the article.** This can be done using techniques such as the popular *Term Frequency-Inverse Document Frequency* (TF-IDF) , which is explained in further detail below. Proper nouns would be useful in this step, as they are more likely to be relevant to what the article is specifically about.

2. **Create a vector.** This vector, based from the definitive terms from the first step, would be a set of keywords and corresponding weights.

Term Frequency-Inverse Document Frequency

TF-IDF[8] is designed to identify terms that appear frequently in one article that don't occur a lot over the entire set of articles (also known as the *corpus*). Variations of it are commonly used by search engines to identify search results that are most relevant to a query. The calculation for TF-IDF is as follows:

Term Frequency

Term Frequency can be most simply calculated as the frequency of a term in a document. However, this could result in a bias towards terms that appear in longer articles. A more accepted way to calculate the Term Frequency therefore is to use a normalising function, called augmented frequency:

$$tf(t,d) = 0.5 + 0.5 \times \frac{f_{t,d}}{max\{f_{t',d} : t' \in d\}}$$

Inverse Document Frequency

Inverse Document Frequency is used to check in how many documents of a corpus D a given term t occurs. It is an inverse function, so as to minimise the value for the term when it is common amongst various different documents. It is also logarithmically scaled. It is calculated using the following formula:

$$idf(t,D) = \log \frac{|D|}{|d \in D : t \in d|}$$

Term Frequency-Inverse Document Frequency

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$$

2.4 Summarisation Techniques

Summarisation techniques will be (predictably) used for summarising the merged articles that were identified by the processes of Topic Modelling , Topic Labelling and Clustering. There are two types of summarisation:

- **Extractive Summarisation**, which consists of taking sentences that are important from the original text, and discarding the rest.[21]
- **Abstractive Summarisation**, which aims to generate a piece of text using natural language techniques. A key to this is that some words in the final summary may not have been the original piece of text.

2.4.1 Extractive Summarisation Techniques

LexRank and TextRank

There are two well known examples of extractive summarisation: LexRank and TextRank.

LexRank and TextRank have very similar methods for extracting a summary[12]. Initially a graph is constructed, that consists of one node for each sentence in the corpus. Then a clustering algorithm is applied, using a TF-IDF calculation to determine similarities.

There are a couple of key differences between LexRank and TextRank. The first arises in the calculation. Both use a TF-IDF calculation, but they are varied slightly. LexRank uses a cosine similarity function in order to weight the final calculation, whereas TextRank uses a more simple logarithmic weighting to perform the calculation.

Both use Google's famous PageRank algorithm to then rank the importance of each sentence based on the calculations in the previous step. With d being a damping factor, the PageRank of a node u is given as:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{p(v)}{deg(v)}$$

After this has been calculated, the top ranked sentences are taken by the TextRank algorithm to form the summary. However, in the LexRank algorithm sentence position and length are also taken into consideration. Also, when adding a sentence to the summary, the LexRank checks the sentence against the summary to ensure that it won't be redundant. As a result of this extra step, LexRank is considered more suitable than TextRank when summarising multiple documents, whereas TextRank is only normally used for summarising a single document.

2.4.2 Abstractive Summarisation Techniques

There are six common methods for abstractive summarisation, which can be split evenly into two distinct categories[23]. These two categories centre around the creation of a representation of the given document:

- **Structure based methods** consist of techniques that involve determining the important information in a document by considering its structure[24]. Ways of doing this include fitting the given document to a template, or converting the text in a tree-like structure.

- **Semantic based methods** involve building a semantic representation of the given document and then feeding that into an algorithm that generates natural language that forms the final summary.

Structure based summarisation

Tree based summarisation

With tree based structuring, the first step is to create a dependency tree to represent the document. A dependency tree is a tree with a node for each word in a sentence, the links between the trees show which words depend on each other. For example, given the sentence *This is an example*, we can construct a tree as in figure 2.13:

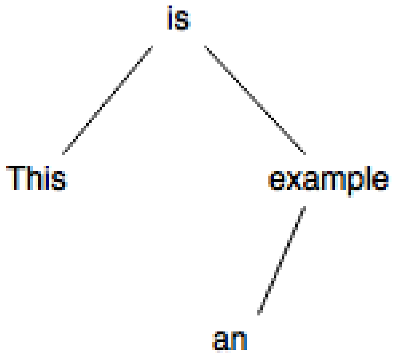


Figure 2.13: A possible dependency tree for the sentence *This is an example*

In the tree (figure 2.13), the word *This* is dependent on *is* and so is linked. *an* is linked to *example* in a similar way, and *an example* is dependent on the verb *is* and so also has a link to it.

Once dependency trees have been created for each of the sentences a similarity algorithm is used to find sentences that are similar. The common phrases between these similar sentences are then taken and form the basis of the final summary. A language generator then combines the common phrases and arranges it to create a final set of summary sentences.

An obvious disadvantage to this method is that if only common phrases are taken from the sentences, context could be lost from some of the sentences. However, on the other hand, the use of a language generator means that a more coherent, more grammatically correct summary is formed.

Rule based summarisation

In rule based summarisation[16], the process is centred around a list of pre-determined categories. With each of these categories, there is a pre-determined set of questions to be answered.

For example, with a theoretical category *Product Launch* we could have the following questions:

1. What is the name of the product?
2. What company is launching it?
3. What type of product is it?
4. What's new about it?
5. What price is it retailing at?

This represents only a subset of the possible questions we could have for this category.

To perform rule based summarisation, the first step is to analyse the document and determine which category it fits best. Once that's been determined, the next stage is to analyse the text to find answers to the questions corresponding to that category. These answers are then fed in to a natural language generator that forms sentences, and thus the summary.

Results for this method of abstractive summarisation have been promising, but the method has a major disadvantage in that the list of categories and questions needs to be pre-determined. As a result, it might not be an optimal algorithm to use for the ever-changing world of news reporting.

Ontology based summarisation

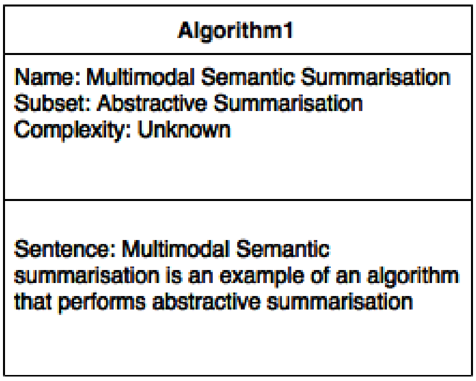
Methods of ontology based summarisation have been developed, primarily using domain based ontology. In this method a "domain expert" defines a domain ontology for a news event. Each new document is then classified into a topic using these domain ontologies. Important phrases are determined by how close they are to items in the ontology. These phrases are then passed into a natural language generator to form the final summary sentences.

A key disadvantage to this method is that a lot of the domain ontologies has to be manually determined by the "domain experts" and so can be very time consuming. As a result this may also not be particularly optimal for summarisation of news events.

Semantic based summarisation

Multimodal Semantic summarisation

Multimodal semantic summarisation can work on documents that contains both text and images. First, a semantic model is built to represent the document. This model is made up of different concepts that are surmised from the text. For example, the sentence *Multimodal Semantic summarisation is an example of an algorithm that performs abstractive summarisation* could form the concept shown in figure 2.14:



Algorithm1

Name: Multimodal Semantic Summarisation
Subset: Abstractive Summarisation
Complexity: Unknown

Sentence: Multimodal Semantic summarisation is an example of an algorithm that performs abstractive summarisation

Figure 2.14: A possible concept created from the analysis of the sentence *Multimodal Semantic summarisation is an example of an algorithm that performs abstractive summarisation*

Concepts are gradually filled with more information as the entire text is analysed. Links are also added between concepts that share some relationship. For example in figure 2.14 if there was another sentence that surmised a concept called *Abstractive Summarisation* then there would be a link from *Algorithm1* to that new concept.

The next step is to rank the concepts. This is done by taking into account the completeness of the concept, and the number of links that the concept as to others. This way, the concepts are ranked by which is most important to the original document. Once the key concepts have been identified summary sentences can be generated featuring these concepts.

Information Item based method

Information Item based summarisation[15] relies on the content of the summary being determined from an abstract representation of the original document, rather than the sentences from the document themselves. To do this, the document is first scanned so that Information Items (InIt) can be generated. An information item is defined as being "the smallest element of coherent information in a text or sentence".

Once the information items have been created, they are then ranked using frequency analysis to find the most important predicates and entities from the original document. This step is near identical to the term frequency stage in extractive summarisation (Section 2.4.1). These information items that are ranked highly are then combined and fed into a natural language generator to form the final summary sentences.

Semantic Graph summarisation

Semantic Graph summarisation centres around a rich semantic graph (RSG). The document is first converted into a RSG. Each node in the RSG represents a noun or verb in the document, and the links between the nodes represent the semantic and topological relations between these nouns and verbs. In the second stage, heuristic rules are used to reduce the semantic graph to a more minimalistic version. This will form the basis of the final summary. In the final step, the minimised RSG is passed into a generator that creates the final summary sentences.

The steps are outlined in the flowchart provided in figure 2.15.

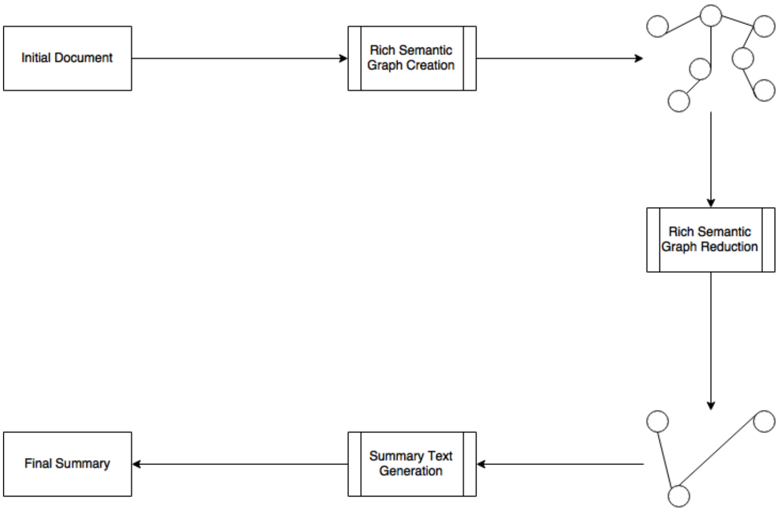


Figure 2.15: This shows the process of semantic graph summarisation in a flowchart based on those provided in [23, 24]

.

Semantic Graph summarisation has had success with producing an abstractive summary that has fewer redundant sentences, and is also good at producing grammatically correct sentences. However, it's only designed for use with a single document as input. As a result it might not be suitable as a method for summarising the multiple documents needed for the aggregator, unless it's combined with another method.

3 Project Plan

3.1 What needs to be done

The project can be split into three sections:

1. Machine Learning and summarisation aspects
2. Back End server
3. Front end application(s)

3.1.1 Machine Learning and Summarisation aspects

This stage will provide the implementation for taking in articles, identifying their topics, clustering them, and summarising them. This can therefore be split into three phases as such:

1. Topic Modelling and Topic Labelling of an article
2. Clustering of articles
3. Summarisation of articles in a cluster

3.1.2 Back End Server

This aspect of the project will have two key functions:

1. Provide articles to the Machine Learning functions of the project.
2. Provide a set of API methods for the front end application(s) that perform the functions a user may require, described in more detail in the front end section.

3.1.3 Front end application(s)

The front end application(s) will need to consist of the following core features:

- Search for specific topics
- Subscribe to specific topics

- View summarised articles
- Create and view news digests

Initially this should be completed on one platform, and if time permits, this could be extended to another platform.

3.2 Milestones and fallbacks

Each phase described in section 3.1 can be a milestone. However, for the project to be deemed a success there are minimal requirements:

3.2.1 Machine Learning and Summarisation aspects

It's easy to say that all aspects of this (section 3.1.1) will be required to make a success of the project. In this case, a minimal requirement, in my opinion, would be for Phase 1 (Topic Modelling and Topic Labelling) and Phase 2 (Clustering) to be fully complete, along with some form of extractive summarisation for Phase 3. An ideal scenario would be to have an abstractive summarisation solution, or even a combination of extractive and abstractive solutions. As a result of this, it would be wise to initially create an extractive solution for Phase 3, as it would then provide a good fallback in case the ideal scenario doesn't pan out.

3.2.2 Back End Server

Both phases of the back end will need to be fully functional, as the other aspects of the project depend on it. Phase 1 should not be particularly difficult to implement however, and phase 2 (API methods for the front end) can be built in conjunction with the front end, and so can adapt to progress on that front.

3.2.3 Front end application(s)

All phases of the front end should be available, with the possible exception of the creation and viewing of news digests, for the project to be deemed a success. At the very least, a skeleton application should be created that provides these features. As extensions, with more time, the UI can be spruced up to provide a good user experience, and then the application could also potentially be extended to new platforms.

3.3 Road Map

A road map is provided below, with a column indicating minimal requirements, and a column indicating an ideal process. At the time of writing (15th January 2017), no aspect of the road map has been started, or completed. In the road map, the dates indicate the end of a time period. For example, something listed in the row for 5th February will be finished on the 5th February, rather than started. Phases in the road map correspond to those in section 3.1.

Date	Minimal	Ideal
05/02	Machine Learning: Topic Modelling complete	Machine Learning Topic Modelling Complete; Report Design Chapter complete
19/02	Machine Learning: Topic Labelling started; Report Design Chapter complete	Machine Learning Phase 1 complete
05/03	Machine Learning: Phase 1 complete	Machine Learning: Phase 2 complete
19/03	Machine Learning: Phase 2 started	Machine Learning: Extractive Summarisation complete
02/04	Machine Learning: Phase 2 complete	Machine Learning: Abstractive solution complete
16/04	Machine Learning: Extractive Summarisation complete	Back End: Phase 1 complete; Report Implementation of Machine Learning written
30/04	Report: Implementation of Machine Learning written	Back End: Phase 2 complete; Front end: Basic Skeleton complete; Evaluation: Machine Learning Summarisation evaluated
14/05	Back End: Phase 1 complete; Evaluation: Machine Learning Summarisation evaluated	Back End: Phase 2 complete; Front End: UI upgrade complete; Report: Implementation of Back End written
28/05	Back End: Phase 2 started; Front end: Basic Skeleton started	Front End: Second Platform started; Report: Implementation of Front End written; Evaluation: Front End evaluation started
11/06	Back End: Phase 2 complete; Front end: Basic Skeleton complete; Report: Implementation chapters started, Evaluation and Conclusion chapters started	Front End: Second Platform completed; Report: Evaluation Chapter written, Conclusion started
25/06	Project report handed in on 19/06	Project report handed in on 19/06

4 Evaluation Plan

There are two key aspects that need to be tested in the evaluation of the project:

1. Summarisation of articles
2. The front end application

4.1 Evaluating the summarisation

When evaluating the summarisation, there are two items that need to be checked. First of all, has the topic been correctly identified? Secondly, is the summarisation presented an accurate depiction of the original article? The evaluation of the topic itself can be done myself initially, and then once summarisation has been completed a good marker for success would be with a user survey.

In this user survey, the user would be presented with an article (or multiple articles) and the summaries. Based on how much of the summarisation plan is actually completed, this aspect of the survey would vary. If both an abstractive and extractive algorithm exist, then the user can be presented with a summary from each. Here the user can be asked two (or possibly three) questions:

1. On a scale from 1 to 5, how good is this summary for the corresponding article?
2. Do you feel this summary misses any key information?
3. *(If multiple summaries are presented.)* Which summary do you think most accurately and fully reflects the article provided?

Question one can be used to evaluate each summary, and thus evaluate the summarisation algorithm that's been used. Question three can then be used to select the best algorithm for the final version of the project. Questions one and two would be designed to evaluate the actual success of summarisation in the project. Since summarisation is an important aspect of the project, a score of 3 out of 5 (60%) would be classed as adequate in my opinion. Average scores of 4 out of 5 or higher should be the aim for the project to be deemed a success.

With the evaluation completed as a user survey, and without the need for a front end, this part of the evaluation can actually be completed as soon as the summarisation algorithms have been completed, and so are listed in the road map as to be done in parallel with the front end development.

4.2 Evaluating the front end

The front end is a bit trickier to evaluate than the summarisation, but can still largely be achieved with the help of a survey.

The key aspects in this section would be to evaluate:

- Searching for specific topics
- Subscribing to specific topics
- Viewing summarised articles
- Creating and viewing news digests

For each of the listed aspects there are two questions: Is it intuitive (can it be done without needing help), and does it work as expected.

To evaluate this, I will provide the application to a focus group, and ask them to do the following:

1. Search for Tim Cook (Apple CEO)
2. Subscribe to Tim Cook
3. Find and read a recent article about Tim Cook
4. View a daily digest for Tim Cook
5. Repeat the process for a topic of their choice

For each step the user will be asked:

- Did you require assistance to perform this task?
- Could this process have been made easier? How?

A truly intuitive application would mean that the answer to both these questions would be no in every case. The application can therefore be deemed a large success if this is the case. In practice, I will deem the application to be a success if there is a 80% success rate.

Another aspect of evaluating the front end would be the evaluation of the user interface. For this, I will take advantage of the Undergraduate Fair, by providing the application to visitors, and asking them for feedback on the user interface and the application in general. A fair like this would be more suited to general feedback, as visitors will be viewing many projects in a short space of time.

Bibliography

Articles

[12] Gunes Erkan and Dragomir R Radev. “LexRank: Graph-based Lexical Centrality as Salience in Text Summarization”. In: *Journal of Artificial Intelligence Research* (December 2004).
Last accessed: 03/01/2017 A paper that shows how to perform LexRank, which is a method of Extractive Summarisation, designed for use on multiple documents.

[15] Pierre-Etienne Genest and Guy Lapalme. “Framework for Abstractive Summarization using Text-to-Text Generation”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (June 2011).
Last accessed: 09/01/2017 This paper provides a method for performing information item based summarisation, which is a type of abstractive summarisation that creates a summary from the abstractive representation of a document, rather than the sentences in the document itself.

[16] Pierre-Etienne Genest and Guy Lapalme. “Fully Abstractive Approach to Guided Summarization”. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (July 2012).
Last accessed: 11/01/2017 An approach to a rule-based summarisation. This method provides a guided alternative to abstractive summarisation, and focuses around multi-document summarisation.

[20] Karl Grieser et al. “Using Ontological and Document Similarity to Estimate Museum Exhibit Relatedness”. In: *ACM Journal ACM Journal of Computing and Cultural Heritage* (2011).
Last accessed: 27/12/2016 This paper tries to evaluate how closely related two terms are by using their respective Wikipedia articles and calculations involving the categories they fall in to, and the number of links to other articles they share.

[21] Vishal Gupta and Gurpreet Lehla. “A Survey of Text Summarization Extractive Techniques”. In: *Journal of Emerging Technologies in Web Intelligence* (2010).
Last accessed: 30/12/2016 A paper that summarises various methods of performing Extractive Summarisation.

[23] N. R. Kasture et al. “A Survey on Methods of Abstractive Text Summarization”. In: *International Journal for Research in Emerging Science and Technology* (November 2014).

Last accessed: 07/01/2017 A paper written by members of the Department of Computer Engineering at a University in Pune, India. The paper discusses the merits and drawbacks of various potential techniques for abstractive summarisation.

[24] Atif Khan and Naomie Salim. “A review on abstractive summarization methods”. In: *Journal of Theoretical and Applied Information Technology* (January 2014).
Last accessed 08/01/2017 A paper that summarises the various methods attempted in the field of abstractive summarisation.

[25] Jey Han Lau et al. “Automatic Labelling of Topic Models”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (June 2011).
Last accessed: 27/12/2016 A paper on a potential method for performing topic labelling, using Wikipedia article titles as candidates for naming the topics. The paper deems Wikipedia to be a good source for topic labels.

[36] Harald Weinreich et al. “Not Quite the Average: An Empirical Study of Web Use”. In: *ACM Transactions on the Web* (February 2008).
Last accessed: 22/12/2016 A paper written in 2008 that aims to evaluate how people use the web, and their internet browsing habits.

Books and Reports

[27] *Reuters Institute Digital News Report 2016*. Reuters Institute for the Study of Journalism, 2016.
Last accessed: 24/12/2016 The Digital News Report is an annual study conducted by the Reuters Institute for the Study of Journalism that aims to evaluate how people are obtaining the news year on year in various countries around the globe.

Webpages

[1] URL: `https://www.quora.com/How-does-Google-News-cluster-stories/answer/Bharath-Kumar-M?srid=Qord`.
Last accessed: 08/01/2017 An explanation from a former Google News employee on the Quora forum website on the broad strokes of Google’s method for clustering articles.

[2] URL: `http://www.macworld.com/article/2358481/wwdc-apple-design-awards-winners-for-2014.html`.
Yahoo News Digest was one of the winners of the 2014 Apple Design Award.

[3] URL: https://en.wikipedia.org/wiki/Latent_semantic_analysis#Latent_semantic_indexing.
Last accessed: 22/12/2016 An article on Wikipedia about Latent Semantic Analysis.

[4] URL: https://www.quora.com/Whats-the-difference-between-Latent-Semantic-Indexing-LSI-and-Latent-Dirichlet-Allocation-LDA/answer/Joseph-Turian?srid=Qord.
Last Accessed: 22/12/2016 An explanation from a Ph.D consultant on the difference between Latent Semantic Indexing and Latent Dirichlet Allocation.

[5] URL: https://www.quora.com/What-is-a-good-explanation-of-Latent-Dirichlet-Allocation/answer/Edwin-Chen-1?srid=Qord.
Last accessed: 08/01/2017 An explanation from Edwin Chen, professor at the University of Buenos Aires, written in 2011, about how Latent Dirichlet Allocation can be used to perform Topic Modelling.

[6] URL: https://en.wikipedia.org/wiki/Cluster_analysis.
Last accessed: 29/12/2016 The various types of clustering, according to Wikipedia.

[7] URL: https://en.wikipedia.org/wiki/Hierarchical_clustering.
Last accessed: 29/12/2016 A Wikipedia article describing the various types and metrics used for hierarchical clustering.

[8] URL: https://en.wikipedia.org/wiki/Tf%E2%80%93idf#Example_of_tf.E2.80.93idf.
Last accessed: 29/12/2016 A Wikipedia article about the justifications and calculations of TF-IDF.

[22] Statistic Brain Research Institute. *Attention Span Statistics.* URL: http://www.statisticbrain.com/attention-span-statistics/.
Last accessed: 23/12/2016 Statistics from the Statistic Brain Research Institute based on a study they conducted in July 2016. The study aims to give an insight into statistics surrounding attention spans in 2016.

Miscellaneous

[9] *Apple.* www.apple.com.

[10] *Apple News.* www.apple.com/uk/news/.

[11] *BBC.* www.bbc.co.uk.

[13] *Flipboard.* www.flipboard.com.

[14] *Fox News.* www.foxnews.com.

[17] *Google.* www.google.com.

[18] *Google Forms*. www.forms.google.com.

[19] *Google News*. www.news.google.com.

[26] *Ofcom*. www.ofcom.org.uk.

[28] *Reuters Institute for the Study of Journalism*. www.reutersinstitute.politics.ox.ac.uk.

[29] *Statistic Brain Research Institute*. www.statisticbrain.com.

[30] *Summly*. www.summly.com.

[31] *The Guardian*. www.theguardian.com.

[32] *The New York Times*. www.nytimes.com.

[33] *Uber*. www.uber.com.

[34] *University of Canberra*. www.canberra.edu.au.

[35] *University of Oxford*. www.oxford.ac.uk.

[37] *Wikipedia*. www.wikipedia.org.

[38] *Yahoo News Digest*. www.uk.mobile.yahoo.com/newsdigest/.

[39] *YouGov*. www.yougov.com.

Index

A

Apple 16
    Design award 16
    News 17
Attention span 4, 5
Automatic Labelling of Topic Models 19

B

BBC 9

C

Centroid Clustering 21
Clustering 13, 21–23, 30, 31
    Centroid 21
    Density 21
    Hierarchical 21
    k-means 21
Concept
    Multimodal Summarisation 27
Connectivity 22
Cook, Tim 34
Customisation 5

D

Density Clustering 21
Dependency Tree 25
Dice's coefficient 21
Digital News Report 6, 7, 9, 10
Domain Ontology 26
Duplicate 5
Duplication 4, 5

E

Evaluation 11

F

Flipboard 4, 15
Fox News 5

G

Gibbs Sampling 19
Google 9, 19
    Forms 10
    News 5, 13, 15
    Search Engine 9
Guardian, The 5

H

Hierarchical Clustering 21
    Agglomerative 22
    Divisive 22

L

Latent Dirichlet Allocation 18
Latent Semantic Analysis 18
Latent Semantic Indexing 18
LDA 18
Length 5
LexRank 23
LSI 18

M

Machine Learning 18

N

New York Times 9
News
    Personalised 8
News Aggregator 4, 5, 11–13
    Reasons for use 7
News Digests 13, 31
Not Quite the Average: An Empirical
        Study of Web Use 9

O

Ofcom 9

P

PageRank 24

R

Related Article Conceptual Overlap . 20
Reuters Institute for the Study of
        Journalism 6–9
Rich Semantic Graph 28
Road Map 32

S

San Francisco . . . 5
Social Media . . . 7
Source . . . 4–6
Statistic Brain Research Institute 4, 5, 9
Summarisation . . . 4, 9, 23, 30, 33
  Abstractive . . . 23, 24, 27, 31, 33
  Extractive . . . 23, 28, 31, 33
  Information Item . . . 27
  Multimodal Semantic . . . 27
  Ontology based . . . 26
  Rule based . . . 25
  Semantic Graph . . . 28
  Tree based . . . 25
Summarised . . . 11
Summariser . . . 4

T

Term Frequency-Inverse Document Frequency . . . 22, 23
  Inverse Document Frequency . . . 23
  Term Frequency . . . 22
TextRank . . . 23
TF-IDF . . . 22, 24
Topic Labelling . . . 19, 23, 30, 31
Topic Modelling . . . 18, 23, 30, 31
  Latent Dirichlet Allocation . . . 18
  Latent Semantic Indexing . . . 18

U

Uber . . . 5
United Kingdom . . . 7–9
University of Canberra . . . 9
University of Oxford . . . 7
User
  Survey . . . 9, 33
Using Ontological and Document Similarity to Estimate Museum Exhibit Relatedness . . . 20

W

Wikipedia . . . 19, 21

Y

Yahoo News Digest . . . 16
YouGov . . . 9