Map-Based Newsfeed Visualisation in Support of Sensemaking

Project Proposal

Damask Talary-Brown

Bachelor of Science in Computer Science with Honours
The University of Bath
2016

Contents

Problem Description

1.1 Introduction

Making sense of collections of related documents is a common task when using the internet for information retrieval, and nowhere is this challenge more prevalent than in the context of reading and understanding the news.

Studies have found the public most commonly use news media to make important life decisions, for entertainment and discussion, as a requirement of their jobs, or out of perceived civic obligation [11, 6]. As a result of the ongoing shift towards online multimedia journalism, there has been an explosion of globally accessible knowledge to support these common uses, which is expanding at an unprecedented rate as the internet grows.

The rise of the internet as an ambient news platform[1] has lead to many users finding the most efficient method of reading news articles is to subscribe to various topic-specific news feeds and read what is automatically collated by their computers [14]. Ubiquitous news media may have resulted in more complete coverage of current events, but the volume of content available online has made the process of understanding it overwhelming to readers.

Broder[2] found that in almost 15% of information query searches (of which searching for news is an example), a related collection of links is the desired result of the search, as opposed to a single document. This suggests that simply reducing a collection of news articles into a single summary is approach to the information overload problem. In addition, news articles are often part of long-running stories [12] such as international conflicts, and therefore fit into a wider set of stories which span a common salient topic.

In spite of these facts, little attention has been given to addressing the problem that understanding news articles individually is inherently reliant on understanding news articles as a collection. Existing information infrastructure has been criticised both for not supporting the cross-correlation between collections of related news articles [8], and for attempting fit complex narratives into visualisations using a single unit of analysis [11]. Research into how humans' spatio-cognitive abilities can be applied to more abstract spatial metaphors [13] suggests that the best visualisation format for collections of this nature may be cartographic.

[1] https://www.researchgate.net/publication/228176202

1.2 Aims and Objectives

This project will seek to develop a tool to solve the twofold problem outlined above; the proliferation of online news content is causing information overload for reader, and there are no general purpose tools enabling readers to explore the contextual relationships between news articles in order to understand the bigger picture.

The objectives of the project are as follows:

1. Investigate efficient methods for keyword extraction and build a generic module which can download articles from a given RSS feed and use an extract keywords which it considers significant.

2. Implement or adapt an existing algorithm to fit a feed of articles into a directed graph with nodes as articles and vertices as salient topic threads, or *stories*. If necessary, the goal of the algorithm will be simplified by not attempting to maximise the coverage of the graph over the set of all topics.

3. Use existing work on information cartography such as [10, 5] to design a map-based visualisation for the graphs which preserves chronology, and refine using sets of real news data.

4. Evaluate how readers use the system, and whether the contextual information provided helps them learn or retain more effectively than simply reading the articles individually.

1.3 Related Work

Interactive exploration of news article chronology is an active area of research, with much work dedicated to visualising timelines of events [1, 12] or using probabalistic topic modelling to extract salient themes from more general document collections [3]. Both Goyal et al.[4] and Wang et al. [14] designed news timeline viewers based on contextual graphs connecting related stories, though the graphs themselves were not part of the final visualisations in either case, making them less applicable to the scope of this project.

Relative topic significance as a determiner for the physical features of a visualisation can be found in [5], where Liu et al. graph continuous keyword frequency for words extracted from 10,000 emails over the course of a year.

Information cartography as a solution to information overload has been explored by Shahaf et al, who present *Metro Maps* as a visualisation for data in the domains of news, science and legal documents [11, 9, 10]. Map-based information design has also been approached from an explicitly cartographic perspective by Skupin [13], who makes recommendations for the use of maps within information design based on the cognitive abilities of humans to process highly dimensional data.

Requirements

This early requirements specification will later form the basis for the full functional and non-functional requirements specification.

2.1 Article Acquisition

- The system must be able to accept any standard XML document compliant with the RSS 2.0 specification.[1]
- The system must be able to extract a specified number of articles from an RSS feed, in reverse chronological order.
- The system must be able to download the textual content and/or raw HTML for each article.
- The system must not be specific to any online news source, and therefore must be continually tested with various different feeds.

2.2 Keyword Extraction

- The system must be able to tokenise articles in order to perform basic NLP.
- The system must implement at least one method of keyword extraction.
- The system must store a corresponding measure of relative importance for each keyword such as TF-IDF[7].
- The system should attempt to combine keywords it considers equivalent (e.g. *UK* and *United Kingdom*) to form stronger keyword matches between or within articles.
- The system may use external services (e.g. Google's Knowledge Graph API[2]) to query any extracted keywords, in order to gain further insight or help disambiguate particular entities.

[1] http://cyber.harvard.edu/rss/rss.html

[2] http://www.google.com/intl/es419/insidesearch/features/search/knowledge.html

2.3 Topic Selection and Graph Formation

- The system must analyse the keywords extracted from all articles in a feed to choose a set of appropriate topics.
- The system must use these topics and the publish dates of the articles to form a directed graph, with articles as vertices and common topic storylines as edges.
- The system should choose topics which are specific to some but not all articles in the collection, so as to form a visually interesting graph.
- The system may attempt to maximise the coverage of the topic selection, i.e. maximise the number of articles covered by a given set of keywords.
- The system may attempt to combine keywords to form topics if it considers them strongly correlated.

2.4 Graph Visualisation

- The system must have the capability to automatically visualise graphs it generates.
- The system should provide drill-down detail for nodes, e.g. by providing a hyperlink to the original article or embedding static content from each article within the visualisation itself to provide a preview.
- The system should ensure the graphs generated are readable by ensuring nodes, edges and labels do not overlap with each other.
- The system may allow some degree of interactive customisation which does not change the underlying structure of the graph, such as dragging nodes or changing attributes including colour.

2.5 Storage and Persistence

- By default, the articles collected by each run of the system must be treated as a new corpus so keyword ranking is deterministic for any given feed.
- The system must be able to store, process and visualise enough data to form a meaningful graph. The exact value of *enough* has yet to be determined.
- The system should support the importing/exporting of an intermediate data form, in order to reconstruct graphs it had previously created.

Project Plan

3.1 Milestones

The major milestones in the form of deliverables and their deadlines are outlined here to act as the starting point for the project plan. In between deliverables are distinct phases which may later be subdivided; all key activities will fit into one of these phases.

Phase 0	Speculative reading, development and scoping.
Project Proposal	**28th October 2016 (Week 4)**

Phase 1	Requirements specification, literature survey, initial proof of concept.
Literature and Technology Survey	**25th November 2016 (Week 8)**

Phase 2	Iteration on initial prototype, refining system, integration of new requirements if applicable.
Demonstration of Progress	**20th February 2017 (Week 21)**

Phase 3 Phase 4	Experimental design, and evaluation of user interaction with application. Final write-up and submission.
Dissertation	**5th May 2017 (Week 31)**

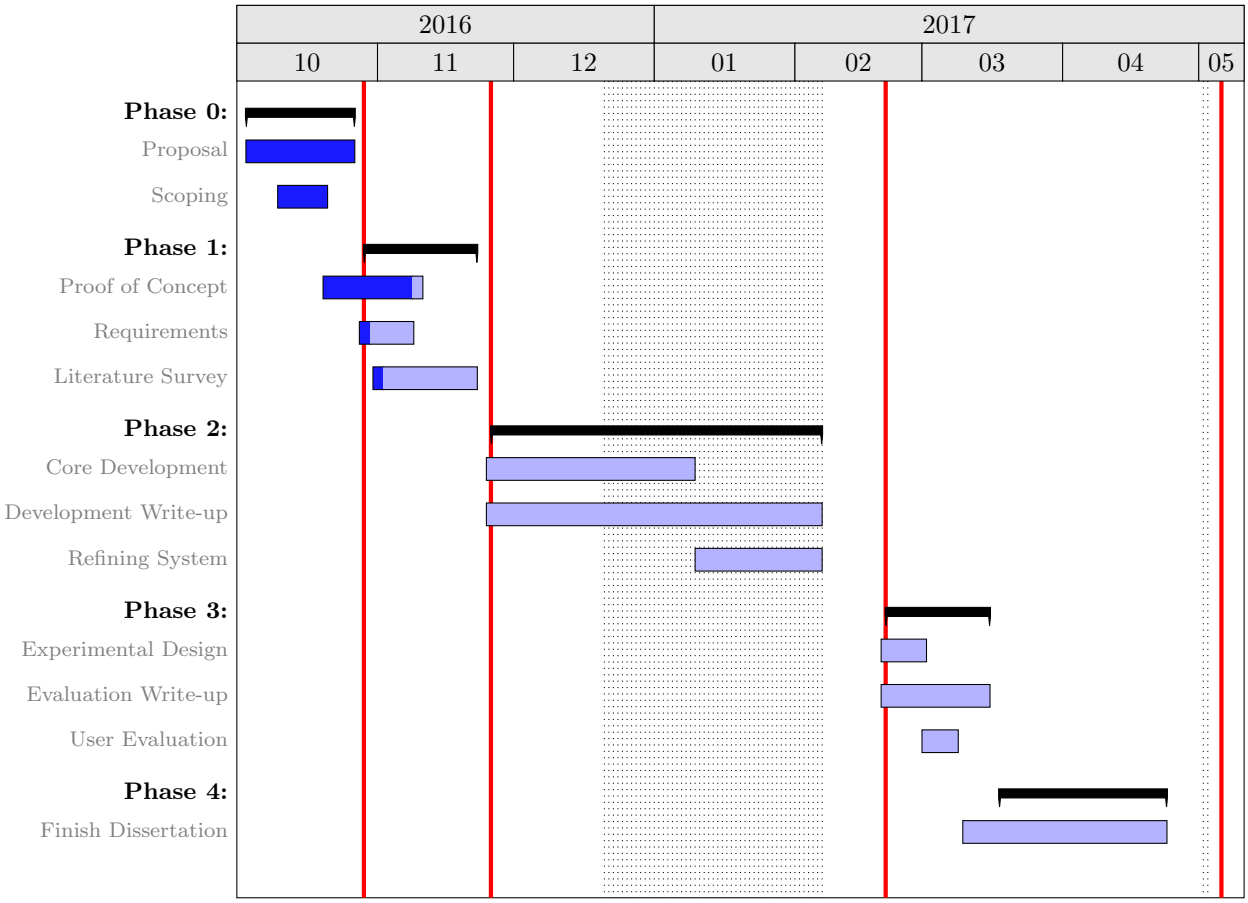
3.2 Gantt Chart

Plotting the project's phases and milestones on a Gantt chart has allowed me to plan and allocate additional time at the end of each phase as a contingency buffer which can, depending on whether or not any problems are encountered, act as a period for reflection on the previous phase or be used to extend it.

The four key deadlines (Proposal, Literature Survey, Demonstration of Progress, and Dissertation) are shown on the chart in red, with tasks for each respective phase listed under that phase and coloured to represent my current state of progress through each.

Dotted areas represent University holidays, where I have deliberately scheduled more flexible tasks in the lead-up to exams. The period from the start of the Christmas break to the end of the Inter-Semester break which includes January exams has been blocked as one break, as I do not anticipate completing a significant volume of work during exam revision.

As I begin each new phase, I will produce a more detailed plan and decompose tasks and their dependencies to ensure I organise my time effectively.



3.3 Probabilistic Risk Assessment

In an attempt to mitigate the most foreseeable high-level risks to the project, I have identified each and categorised their severity and likelihood, along with a basic contingency plan.

Hardware failure has not been recorded as a risk, as there will not be a physical component to the project, and both and the code and the documentation are hosted privately on GitHub and mirrored on Google Drive, with hourly physical backups.

Risk	Severity	Likelihood	Contingency
Underestimated project size and/or scope.	Moderate	Possible	Use scheduled buffer to refine requirements by priority.
Wrongly assessed technical feasibility of project.	Critical	Unlikely	Use external libraries or APIs to solve problems encountered which are not solvable in the given time.
Personal reasons/illness.	Critical	Possible	Use schedule buffer to make up time, or refine requirements if necessary.
Underestimated development or documentation time.	Critical	Likely	Use schedule buffer to make up time, or refine requirements if necessary.
Software or library failure.	Catastrophic	Possible	Manually download and compile older versions of the open source libraries required.
Scoped project too narrowly/Project lacks complexity.	Catastrophic	Unlikely	Use Core Development (Phase 2) to iteratively expand research and create new requirements in order to broaden scope.

Resources

I plan to develop the project in Python 2.7, which is freely available, and both maintained and supported by a large community of developers. My decision was motivated by my previous development experience in Python, and the availability of the open source Natural Language Toolkit (NLTK) package[1], which can perform linguistic processes including tokenisation, lemmatisation and stop word filtering. For other modular tasks such as parsing the RSS feeds, there are a variety of open-source Python packages available.

In terms of graph visualisation, I have two options which, during requirements specification I will have to evaluate and decide between. The first is using an open source imagine library for Python such as PIL[2] with a graphing class to support the underlying collection model. The second option is for the system to interactive web reports containing the graph structures in JSON/GraphML, with visualisations written in JavaScript.

Any APIs I may use for the system will be free and accessible without delay or constraints for academic use. Currently, I only anticipate using Google's Knowledge Graph API, for which I already have an API key.

Other resources:

- For source control and as a backup solution, I have versioned both my source code and documentation files under Git, and hosted them privately on GitHub.
- As there is no physical component to the project, there are no additional hardware resources to consider beyond my own computational resources.
- For user testing and evaluation, I have classmates in Computer Science and other degree disciplines who have agreed to participate in these activities.
- For general guidance, I have arranged weekly supervision meetings to ensure the project progresses at a consistent rate.

[1] http://www.nltk.org
[2] http://www.pythonware.com/products/pil/

Bibliography

[1] Giang Binh Tran. Structured summarization for news events. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, pages 343–348, New York, NY, USA, 2013. ACM.

[2] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, September 2002.

[3] Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Discovering diverse and salient threads in document collections. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 710–720, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[4] Rahul Goyal, Ravee Malla, Amitabha Bagchi, Sameep Mehta, and Maya Ramanath. Esthete: A news browsing system to visualize the context and evolution of news stories. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 2529–2532, New York, NY, USA, 2013. ACM.

[5] Shixia Liu, Michelle X. Zhou, Shimei Pan, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 543–552, New York, NY, USA, 2009. ACM.

[6] Kristen Purcell, Lee Rainie, Amy Mitchell, Tom Rosenstiel, and Kenny Olmstead. Understanding the participatory news consumer. *Pew Internet and American Life Project*, 1:19–21, 2010.

[7] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[8] Earl Rennison. Galaxy of news: An approach to visualizing and understanding expansive news landscapes. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pages 3–12, New York, NY, USA, 1994. ACM.

[9] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. Metro maps of science. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1122–1130, New York, NY, USA, 2012. ACM.

[10] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. Trains of thought: Generating information maps. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 899–908, New York, NY, USA, 2012. ACM.

[11] Dafna Shahaf, Carlos Guestrin, Eric Horvitz, and Jure Leskovec. Information cartography. *Commun. ACM*, 58(11):62–73, October 2015.

[12] Jaspreet Singh, Abhijit Anand, Vinay Setty, and Avishek Anand. Exploring long running news stories using wikipedia. In *Proceedings of the ACM Web Science Conference*, WebSci '15, pages 57:1–57:2, New York, NY, USA, 2015. ACM.

[13] André Skupin. From metaphor to method: Cartographic perspectives on information visualization. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 91–, Washington, DC, USA, 2000. IEEE Computer Society.

[14] Taifeng Wang, Nenghai Yu, Zhiwei Li, and Mingjing Li. nreader: Reading news quickly, deeply and vividly. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1385–1390, New York, NY, USA, 2006. ACM.