ENHANCING NEWS RECOMMENDATION USING A PERSONALIZED CONTENT MANAGER

University of Jyväskylä
Master’s thesis

2018

Charles Osemudiame Okojie
Faculty of information technology
Oleksiy Khriyenko
Vagan Terziyan

February 22, 2018
### Abstract:

The surge in the amount of information available on the internet and the number of users utilizing such information poses new challenges for information systems. This rapid growth in information is palpable in the news provisioning domain where users spend time deciding which of the many channels provides news in a most reliable and useful fashion. In the last few years, News aggregation platforms are now present which reduces the time users spend in consuming news from multiple sources. But news provisioning is more than just aggregating and presenting news. It must also include taking into cognizance users’ needs. Therefore, recommendation algorithms are developed by information systems and news contents are recommended and personalized for users while utilizing user’s specific data. If content recommendation is to be optimal, better and more efficient algorithms must be developed and implemented. Challenges associated with this type information systems include providing quality and novel information, feeding out relevant information while dealing with problems such as ‘data-sparsity’ commonly associated with recommending content. To this end I conduct a study which employs a hybrid approach to solving the existing problems with recommendation system and providing quality, relevant and serendipitous news content for users.

### Keywords:

List of Figures

Fig. 1. A simple structure of an RSS Feed
Fig. 2: Client-server architecture
Fig. 3: HTTP request message
Fig 4: HTTP response header
Fig 5: Example of a JSON Data
Fig 6: A News API endpoint
Fig 7: A News JSON API response
Fig 8: ExpressJS handling GET request
Fig. 9: Starting a ExpressJS web server
Fig. 10: A user profile MongoDB schema
Fig. 11: Mongoose CREATE operation
Fig. 12: Mongoose READ operations
Fig. 13: Mongoose READ operation by id
Fig. 14: IBM Watson analyzing data using natural language processing
Fig. 15: IBM Watson NLU API authentication
Fig. 16: Specifying the feature to be requested and data to be analyzed
Fig. 17: NLU analyzing features of data
Fig. 18: API request to parse data from news sources
Fig. 19: Mercury Web Parser response
Fig. 20: Content aggregation processes
Fig. 21: Content recommendation processes
Fig. 22: Response from IBM NLU analysis
Fig. 23: User’s Login Page
Fig. 24: Aggregated News presentation view

Fig. 25: Aggregated News categorized

Fig. 26: The PCM saves analyzed news stories in MongoDB

Fig. 27: UI for personalizing interest

Fig. 28: Finding top nearest neighbors

Fig. 29: Returned Object from similarity matching

Fig. 30: Users rate stories based on their judgement of quality

Fig. 31: Collaborative filtering recommendation Interface

Fig. 32: Calculating prediction value for each rated Item

Fig. 33: Final prediction using PSV
List of Table

Table 1: Common information quality definition frameworks

Table 2: User-item matrix

Table 3: Common filtering techniques pros & cons
TABLE OF CONTENT

Abstract
List of Figures
List of Tables
Chapter 1
Introduction
1.1 Objective of Study .............................................................. 5
1.2 Research Questions ............................................................ 6
1.3 Research Methodology ......................................................... 6
1.4 Structure of Thesis ............................................................... 8

Chapter 2
Defining Quality, Relevance & Serendipity
2.1 Defining Information Quality ................................................. 10
2.2 Defining Relevance ............................................................. 15
2.3 Serendipity Explained ......................................................... 17

Chapter 3
News Content Aggregation
3.1 RSS Feeds ................................................................. 21
3.2 Content Scraping ............................................................... 24

Chapter 4
Recommendation Processes
4.1 Collaborative filtering ......................................................... 26
4.1.1 Model-based collaborative filtering .................................... 28
4.1.2 Memory-based collaborative filtering ................................. 29
Chapter 5

Technologies Used in This Thesis

5.1 Hypertext Transfer Protocol (HTTP) ........................................... 39
5.2 JavaScript Object Notation (JSON) ........................................... 41
5.3 Application Programming interfaces (API) ............................. 43
5.4 Frontend and Backend Technologies .................................... 44
5.5 IBM Watson Natural Language Understanding .......................... 49
5.6 Mercury Web Parser ............................................................ 53

Chapter 6

Algorithm Implementation and System Structure

6.1 The Aggregation process ...................................................... 55
6.2 Recommendation process ................................................... 59
6.2.1 Profession Strength Value \( (PSV) \) ................................. 63
6.2.2 User walkthrough .......................................................... 64
6.2.3 Evaluation of the recommender system ............................ 65

Conclusion ................................................................. 68

Acknowledgement .......................................................... 71

Bibliography
Chapter 1

Introduction

Over the last decade, the explosion of the web associated with the exponential growth of internet users and the wave of mobile devices has had significant implications on how news is being distributed and consumed. According to the study by (Nielsen Global Survey, 2015) on “Generational Lifestyles” – after television, search engines and social media are platforms which people turn to for their news source.

The topic of news provisioning is an important subject since news plays an important part of human life. This study focuses on using digital news contents to meet users’ news demands. Though print media still exist and is much useful in providing news for consumers, more people are even more dependent on the digital version for news especially with the growing number of tech companies such as Facebook, Apple, and Google. Journalists are now turning to the digital in other to maintain their customer base and in a quest to be innovative. A study by (Ziming & Liu, 2005) on changes in reading behavior in the last ten years shows that 67% of participants in the survey they carried out spends more time reading now than ever before. Their study indicates that possible reasons for this increase is (a) increasing availability of information and (b) the digital era.

Digital news provides some advantages which cannot be found in print media such as convenience, accessibility, interactivity, and the inclusion of videos, images, and audios. It has also promoted the way data is produced and information utilized by news providers. It gets better with the uprising of Data Journalism to help understand the massive amount of data available on the internet. “Data Journalism is digging into information and then making sense of it for people” (Rebekah Monson, 2014). This field has changed the way journalism is carried out. (Parasie & Dagiral, 2013) says that - in relation to government, data journalism has contributed
massively “especially at a time when a growing number of data sets are released by government”, and has done so in three different ways: 1) “by strengthening journalistic objectivity” 2) “by offering new tools to news organizations to sustain government accountability” and 3) by increasing citizens’ political participation through their own production and analysis of data. In this study, I develop an information system that provides personalized news contents based on user's preferences with emphasis on relevance and serendipity while utilizing a combined filtering technique.

“Web content personalization is the use of technology and customer information to tailor electronic commerce interactions between a business and each customer. By using information previously obtained or provided in real time, the exchange between the parties is altered to fit that client's stated needs, as well as needs perceived by the business based on the available customer information” (Vesanen, 2007).

Personalization has been beneficial to both clients and businesses alike, but the concept itself is broad and can be interpreted differently depending on business type and the content to be personalized. One cannot think of personalization without also considering ‘value’ and ‘satisfaction’ These I think are the focal points of personalization. Therefore, personalization tries to answer the question: (1) How can customers & businesses get value for time and money? (2) How can customers get the needed and optimal satisfaction?. (Vesanen, 2007) stated that if there is no common framework for defining personalization, there would always be a problem where parties involved do not understand each other, thus hindering the development of a common ideology about personalized marketing. He then developed a framework which helps value chain actors to know each other and the different aspects of personalization from both customer and marketer perspective. It is worth remembering that for personalization to work, it must have customer data, probably some external data, a customization process, some operation using the
data and a delivery mechanism. In news provisioning, I believe the main areas to be considered are quality, serendipity, and relevance. I dedicate chapter 2 to discussing these terms as they make up the problem area this study addresses. The system developed in this study relies on users’ data and an appropriate algorithm for providing personalized news content from some online news sources for the reader.

News aggregation is a process of taking news from various sources, bringing it together and displaying it to interested readers. The current trend with digital news aggregators is to consume news content from different sources. News publishers make deals with news aggregators to place their content on aggregators’ website in exchange for some revenue or traffic. An example of such aggregators includes Flipboard¹, Google News², Yahoo News³. Yahoo News relies on data from multiple sources such as Associated Press⁴, Reuters⁵, and Fox News⁶, combines the sources and then offers them to readers.

I think that publishers are seeing that users do not want to go to one single source for all their information need instead they want to take advantage of the greatest diversity of voices available (Simon, n.d.). The system developed in this study is a news aggregation platform which collates news from sources and offers them to users in a way that meets the information needs of those users with the corresponding level of relevance. However, users are central in determining what is relevant and not just the algorithm.

¹ Flipboard – a personalized magazine app for mobile devices but also accessible on PC – https://flipboard.com
² Google News – an aggregation system that displays news according to reader’s interest.
⁴ Associated Press – an American not-for-profit news agency headquartered in New York - https://www.ap.org/about/
⁵ Reuters – A multimedia news provider and a media division of Thomas Reuter - https://www.reuters.com/
⁶ Fox News – an American cable owned by Fox Entertainment Group
In a quest to meet personalization and relevance need of users, tech giants such as Google and Facebook are continuously refining algorithms which rely on massive data from users. Though they may argue that their intentions are just to meet the needs of the user, there is an adverse effect to this. Lynn Parramore defines this effect called “filter bubble” or as (Mcnee, Riedl, & Konstan, 2006) puts it the “similarity hole” as – the personal ecosystem of information been catered by algorithms to whom they think you are (Lynn, 2010). While this may be good in some cases, it hinders serendipity, learning, and curiosity by users because it presents users with what they have grown to be interested and nothing novel. Therefore, the primary problem which news aggregators must solve is providing a balance between personalization and serendipity. This study intends to solve these problems by employing an algorithm which also helps with relevant content discovery.

I propose the use of a Personalized Content Manager (PCM). The PCM would consist of an algorithm and utilizes a NoSQL database which contains information about news readers including their preferences and ratings on news items. This information is shared with other users with similar interest and would be used to recommend relevant news content based on rating weights.

The traditional method for news publishers to get information to readers is via RSS feeds which a user must subscribe to after been submitted by the publisher. RSS feeds only contains the headline of the news and a link to the full content on the publisher web page. Publishers such as CNN⁷ and New York Times⁸ use RSS 2.0 format to describe their data and content aggregators such as Google News, Flipboard, Yahoo News all conforms to the RSS 2.0 format. This

---

study does not depend on RSS to present content to users but uses web scraping tool to extract data from news provider’s websites and make a visual presentation of the data.

1.1 Objective of Study

As explained in the introduction section, a fundamental problem to be solved in this study is relevant and serendipitous content provisioning. Given news stories on topics such as sport, politics, entertainment, how can the system recommend relevantly by providing content previously rated by other users to a new or active user while excluding the ‘filter bubble’ effect. I explore an optimal way of making such content recommendation. I do so using an algorithm which is inclusive of the two steps in every content recommendation system – similarity & prediction.

On any information system, the ‘quality’ of the information is a primary attribute which must be considered. However, there isn’t an easy definition for it because of its elusive concept as (Lacy & Fico, 1991) describes. By using the quality assessment measures as defined in section 2.1, the system puts to use users’ judgment on a story to determine the quality and then make recommendations to other users. While quality is an attribute typical for individual news stories, relevance & serendipity is essential for any news recommendation system.

Few news aggregators such as Upday⁹ employ a combination of algorithms and real life journalism in selecting articles. Their primary purpose for this approach is to avoid as much as possible the filter bubble. They set up an editorial team of six journalists in every country they have launched; this makes them stand out from other news aggregators. However, implementing such a system would be expensive and doesn't yet solve the problem of bias news content.

---

from only certain publishers. I intend to address the adverse effects of human intervention in deciding news stories offered to readers.

1.2 Research Questions

I. How is content recommendation defined and how do existing recommendation algorithm differ from each other?
II. How does the system meets users’ demand for novelty?
III. What better algorithm implementation can be used to optimize recommendation?
IV. How does the developed algorithm meet user’s requirements concerning the relevance level?

1.3 Research Methodology

For this thesis, I employ a Design Science Research Method (DSRM) methodology as proposed by (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007) because of its suitability for this information system.

They defined Design Science (DS) as involving a rigorous process to design artifacts to solve observed problems, to make research contributions, to analyze the designs, and to communicate the results to appropriate audiences (Peffers et al., 2007). The aim in their publication was to develop a methodology that would serve as a commonly accepted framework for carrying out research based on existing DS research principles. While there have been various approaches and methods developed in design science by researchers, the authors rather focus on a consensus approach to build their model. They utilize an approach accepted by researchers to ensure that they based the DSRM on well-accepted elements. Their methodology is a six activities sequence which I shall explain in the following.
Process 1: Problem Identification and motivation – To provide a solution, the problem must first be defined. I think this is the most important step in any research process because if not correctly identified, research may not be sufficient. The authors suggest that it is best to atomize a research problem in other to visualize its complicity. Though this process may be less complicated depending on the research being carried out, “resources required for this activity include knowledge of the state of the problem and the importance of its solution” (Peffers et al., 2007). This methodology separates problem identification and resolution objective because design process is a step by step process. I defined lack of relevance and novelty in current news content recommendations systems as issues to be tackled in this thesis using a hybrid algorithm.

Process 2: Objective for the solution – In addition to a knowledge of the problem, a knowledge of existing solutions are resources needed for the success of this process. This phase is a product of the problem identification phase and objectives developed which may either be quantitative such as terms in which a desirable solution would be better than current ones or qualitative such as a description of how a new artifact is anticipated to brace solutions to problems not previously addressed (Peffers et al., 2007). In this study, I elaborate on the objectives of my solution and how it solves the problems stated. I also look at existing solutions and how they differ from my solution.

Process 3: Design and development – “This activity include determining the artifact's desired functionality and its architecture and then creating the actual object” (Peffers et al., 2007). The artifact developed in this study is a model which intends to solve problems of major concern in the field of Information retrieval while employing a better recommendation algorithm. A knowledge of theories required for the solution are resources needed for this process.
**Process 4: Demonstration** – This process seeks to prove that the proposed idea works. For this process, I develop an actual system which consists of the important features of any news aggregation system, and I then employ the algorithm. A knowledge of how to use the artifact to solve the problem is a resource for this process.

**Process 5: Evaluation** – “This activity involves comparing the purpose of a solution to actual observed results from use of the object in the demonstration” (Peffers et al., 2007).

**Process 6: Communication** – This is a final process which involves communicating the importance of the solution which help to break down the resulting knowledge and lays out its significance and effectiveness to the audience or target users.

According to the authors, it is not compulsory that every research would sequentially undergo the process from process 1 through 6. “In reality, they may start at almost any step and move outward” (Peffers et al., 2007). Some researcher may decide to start with a problem-centered approach or objective centered approach or design and development centered approach. I utilize a problem centered approach and then follow the sequence by observing the problems and proposing solutions.

**1.4 Structure of Thesis**

I have structured this thesis into six chapters. Chapter 1 is the introduction and explains what this study is about, including the problem to be addressed, the objective, the scope of this thesis and the methodology employed. Chapter 2, 3 and 4 provides relevant knowledge, history, theory, and overviews other researchers view about news provisioning, collaborative filtering, aggregation processes, etc. Chapter 2 elucidates on the meaning of quality, relevance, serendipity and how they relate to the system developed. Chapter 3 provides details discussion on news
aggregation by RSS feed and content scraping. Chapter 4 expounds on recommendation systems by differing between the two main types of filtering techniques (content-based and collaborative filtering) and then justifies why a combination of both methods is used in this study. Chapter 5 discusses the technologies used in this study; when and how they are being employed including code snippet. Chapter 6 also provides code snippets but formally presents the algorithm implementation and process of the system utilization; also from the user's view. The summary section again defines the problem that was solved, summarizes the conclusion and discusses the future study in news provisioning.
Chapter 2
Defining Quality, Relevance & Serendipity

This section discusses quality, relevance, and serendipity as attributes to be considered for the personalized content manager. One major problem which has resulted from the explosion of information via the web is how to present users with the appropriate information at the right time. This challenge has led to many researchers (both academic and industrial) suggesting and implementing techniques to filter irrelevant information from the relevant ones for web services consumers. (Truong et al., 2010) asserts that this is crucial to improving the efficiency and the correctness of service composition and execution.

2.1 Defining Information Quality

Information quality, IQ or Data quality DQ are interchangeable terms in the information systems domain. “The focus on IQ from the perspective of Information Retrieval is a relatively new research area but is critical if information retrieval systems are to become useful tools for retrieving quality information from the ever-burgeoning Worldwide Web” (Knight & Burn, 2005). As (Knight & Burn, 2005) further states the lack of enforceable standards regarding the information contained on the internet has led to numerous quality problems. (Knight & Burn, 2005) is correct since there is no laid down policies or procedures for determining quality information before being pushed to the web. In defining the concept itself, (Lacy & Fico, 1991) defines quality as an elusive concept because of its different meaning to different people making it more subjective. (Giri Kumar & Donald, 1998) supports this view but defines data quality as “fitness for use” but states that this produces a personal environment where one user’s quality definition could be of little or no value to another user. That is, content considered appropriate for one use may not possess sufficient quality for another use.
Both (Strong, Lee, & Wang, 1997) and (Knight & Burn, 2005) agree that quality cannot be accessed independent of users while (Strong et al., 1997) expressed that “Information consumers’ assessments of data quality are increasingly important because consumers now have more choices and control over their computing environment and the data they use”. Over the years there have been many criteria used to determine the quality of a piece of information including timeliness, accuracy, usefulness/relevance, reliability, understandability, consistency, completeness, etc.

(Wang & Strong, 1996) discusses a framework for defining data quality. Their structure captures 118 data quality attributes, “consolidated into twenty dimensions and in turn grouped into four categories”. They concluded that their framework is methodologically sound, and that “it is complete from the perspective of data consumers” (Wang & Strong, 1996). In the framework quality is divided into four different categories. 1) Intrinsic, 2) Representational 3) Contextual 4) Accessibility. They maintain that intrinsic quality not only include accuracy and objectivity but also “believability and reputation” and that accuracy and objectivity are not enough to decide if a piece of information is of high quality. “Mismatches among sources of the same data are a common cause of intrinsic DQ concerns” (Strong et al., 1997). If there is a quality problem with information, it is first a believability problem which further develops into accuracy problem and then affects the reputation of the data sources. “As a reputation for poor-quality data becomes common knowledge, these data sources are viewed as having little-added value for any organization, resulting in reduced use” (Wang & Strong, 1996). Accessibility quality emphasizes the importance of information been easily available/accessible for users. Information consumers are also finding that this quality measure is as important as other measures. To a small extent, it also includes how “secure” is the information provided? In the case of news provisioning system, it is vital that information be readily available to news consumers. However, this is what news aggregation solves by providing users with stories from various sources.
Interpretability, understandability, consistency, concise representation according to them are subsets of **Representational quality**. Representational quality include aspects related to the structure of the data (concise and consistent representation) and significance of data (interpretability and ease of understanding) (Wang & Strong, 1996). Though information consumers make the final say in deciding whether the information is represented well or not, it is also true that data author must make sure that for data consumers to conclude that data are adequate, they must be not only concise and consistently represented, but also interpretable and easy to understand. One of the causes which they concluded was responsible for data consumer complaint was “missing (incomplete) data” (Wang & Strong, 1996) or “inadequately defined or measured data” (Wang & Strong, 1996) or “data that could not be properly aggregated” (Wang & Strong, 1996). Completeness, better measured (appropriate amount of data) information, timeliness, relevancy, and value added are all dimensions which define **contextual quality**.

“While the Wang and Strong framework was developed in the context of traditional information systems, the structure has also been applied successfully to information published on the World Wide” (Klein, 2002). However, their framework is not the only accepted framework for information/data quality, in the past decades, information researchers have developed the different framework for information systems. While varied in their approach and application, the structures share some characteristics regarding their classifications of the dimensions of quality (Knight & Burn, 2005). Table 1 below summarizes five different frameworks developed and how they vary in their approach to defining information quality.
<table>
<thead>
<tr>
<th>Author</th>
<th>Model</th>
<th>Constructs</th>
</tr>
</thead>
</table>
|                        | *Summary:*                                                          | ✔️ Intrinsic IQ  
|                        | ✔️ 4 Categories                                                      | ✔️ Accuracy, Objectivity, Believability, Reputation  
|                        | ✔️ 16 Dimensions                                                    | ✔️ Accessibility IQ  
|                        |                                                                     | ✔️ Accessibility, Security  
| [Zeist & Hendriks, 1996] | Extended ISO Model                                                 | **Contextual IQ**  
|                        | *Summary:*                                                          | ✔️ Relevancy, Value-Added, Timeliness, Completeness, Amount of Info  
|                        | ✔️ 32 Sub-characteristics                                           | **Representational IQ**  
|                        |                                                                     | ✔️ Interpretability, Ease of Understanding, Concise Representation, Consistent Representation |
| [Alexander & Tate, 1999] | Applying a Quality Framework to Web Environment                     | **Characteristics**  
|                        | *Summary:*                                                          | ✔️ Functionality  
|                        | ✔️ 6 Criteria                                                       | ✔️ Suitability, Accuracy, Interoperability, Compliance, Security, Traceability  
| [Katerattanakul et al, 1999] | IQ of Individual Web Site                                           | ✔️ Reliability  
|                        | *Summary:*                                                          | ✔️ Maturity, Recoverability, Availability, Degradability, Fault tolerance  
|                        | ✔️ 4 Quality Categories (adapted from Wang & Strong)                | **Efficiency**  
|                        |                                                                     | ✔️ Time behaviour, Resource behaviour  
|                        |                                                                     | **Usability**  
|                        |                                                                     | ✔️ Understandability, Learnability, Operability, Luxury, Clarity, Helpfulness, Explicitness, Customisability, User-friendliness  
|                        |                                                                     | **Maintainability**  
|                        |                                                                     | ✔️ Analysability, Changeability, Stability, Testability, Manageability, Reusability  
|                        |                                                                     | **Portability**  
|                        |                                                                     | ✔️ Adaptability, Conformance, Replaceability, Installability  
| [Shanks & Corbitt, 1999] | Semiotic-based Framework for Data Quality                           | ** Criteria**  
|                        | *Summary:*                                                          | ✔️ Authority  
|                        | ✔️ 4 Semiotic descriptions                                           | ✔️ validated information, author is visible  
|                        | ✔️ 4 goals of IQ                                                     | ✔️ Accuracy reliable, free of errors  
|                        | ✔️ 11 dimensions                                                    | ✔️ Objectivity presented without personal biases  
|                        |                                                                     | ✔️ Currency content up-to-date  
|                        |                                                                     | ✔️ orientation clear target audience  
|                        |                                                                     | ✔️ navigation Intuitive design |
|                        | **Explanation**                                                     | **Category**  
|                        | ✔️ Semiotic Level                                                   | ✔️ Dimension  
|                        | ✔️ Goal                                                             | ✔️ Syntactic  
|                        | ✔️ Dimension                                                        | ✔️ Consistent  
|                        |                                                                     | ✔️ Well-defined / formal syntax  
|                        |                                                                     | ✔️ Semantic  
|                        |                                                                     | ✔️ Complete and Accurate  
|                        |                                                                     | ✔️ Comprehensive, Unambiguous, Meaningful, Correct  
|                        |                                                                     | ✔️ Pragmatic  
|                        |                                                                     | ✔️ Usable and Useful  
|                        |                                                                     | ✔️ Timely, Concise, Easily Accessed, Reputable  
|                        |                                                                     | ✔️ Social  
|                        |                                                                     | ✔️ Shared understanding Understood, Awareness of Bias of meaning  
|                        | **Table 1: Common information quality definition frameworks**       | (Knight & Burn, 2005) |

Research by (Klein, 2002) shows that users do not distinguish between accuracy of information and timeliness of information in the way suggested by the theoretical framework. That is, users somehow view timeliness and accuracy together such that “evidence that information is out-of-date provides a high signal to users that it may no longer be accurate” (Klein, 2002). In listing preliminary factors associated with information system quality problems, (Klein, 2002) listed: accuracy, completeness, relevance, timeliness, the amount of data as factors to be considered.
Current research into this topic shows that ‘relevancy’ is a prominent factor to be considered in information provisioning systems. Relevancy is a dimension which this study considers as important in recommending stories to users. More on relevance is in section 2.2. When news consumers read stories, questions about relevance arise, including: How relevant is this article to me and does this article meet my need?

Providing relevant story is directly influenced by how a user defines quality. This study focuses on evaluating and providing relevant stories by letting users decide what is quality & relevant to them and then providing those stories to related users. (Lacy & Fico, 1991), says that the measurement of quality needs an elaboration from the readers and that editors may accurately discern some of the readers’ needs and wants but it is doubtful that they are perfect judges. Therefore “an editor based assessment of quality could be supplemented by a user based assessment” (Lacy & Fico, 1991). This study relies on user based judgement of what is quality and relevant.

So far, I have defined quality from the users’ perspective (as it is also the view in this study), but I believe in determining the quality of news content quality journalism cannot be overlooked and is most evident in the content produced. Multiple studies have shown that quality content increases readership. According to (Bogart, 2004), news content producers that maintain high-quality journalism are likely to be well managed in their business operations, and that good journalistic quality is directly proportional to increase circulation. (K. Kim & Meyer, 2005), by studying New England Newspapers suggested that the correlation that exists between distribution and quality may be that newspapers with higher circulation are better than newspapers with a lesser circulation Also (Bogart, 2004) while using data from Inland Press Association, found a connection between quality and news distribution.
2.2 Defining Relevance

Like quality, relevance is also not a straightforward term to describe. “Relevance is, in fact, a central concept in human communication, and a term we use always and loosely in everyday conversation. Sadly, the meaning of the concept is still not clear” (Schamber, Eisenberg, & Nilan, 1990).

It has played a significant role in the information retrieval field, and information practitioners have used it in the “evaluation of information systems and in empirical studies of human information behavior” (Schamber et al., 1990). Though previous attempts have been made to define this term, serious questions about the nature of relevance remain. However, researchers such as (Borlund, 2003), demonstrates that a “consistent and compatible understanding of the relevance concept has been reached”. Many information retrieval systems rely on user’s judgment in defining what is relevant, and feedback mechanisms are employed and used to modify existing system for future recommendation. As (Schamber et al., 1990) point out “In such systems, relevance is no longer a reactive concept, to be used primarily in the evaluation, but an active concept vital to the functioning of the system itself”.

According to (Schamber & Eisenberg, 1988) relevance is a multidimensional concept which is dependent on both internal and external factors, and is intersubjective but systematic and measurable. (Schamber & Eisenberg, 1988) defines relevance as one of matching or topicality, which considers whether the topic of retrieved information matches the subject of interest. They consider this definition to be “system oriented” where the information retrieval system relies on matching words or further measures the frequency with which terms that describe the content of document occurs or how relevant terms tend to cluster in certain linguistic patterns or sets. They suggest that topicality is not enough in determining relevance because “while it depends on matches between queries and documents for terms, it does not necessarily encompass the
information needs of the user” (Schamber & Eisenberg, 1988). Therefore, they defined a user-oriented approaches which included ‘usefulness’ and defines it as the degree to which information fulfills a user needs.

(Saracevic, 1996) expands the relevance concept further by dividing it into different types or “manifestations”: algorithmic, topical, pertinence and situational or utility. Algorithmic relevance seems the most common and clearest definition of relevance and is the type applied in the traditional assay of information retrieval systems (Borlund, 2003) and deals with which query matches the retrieved content. Topical relevance deals more with “aboutness” rather than the content itself. It is a relationship between the subject or topic conveyed in a query, and topic or subject covered by fetched texts, or more broadly, by documents in the system's file, or even in existence (Saracevic, 1996). Pertinence relevance reflects the relationship “between the state of knowledge and cognitive information need of a user, and texts retrieved, or in the file of a system, or even in existence” (Saracevic, 1996). Pertinence relevance allows that the dynamic needs of users are considered, and it involves more of human judgment factor. Situational relevance deals with the problem at hand and how the retrieved information solves the problem and how it could be useful in better decisions making for users (Saracevic, 1996). (Schamber et al., 1990) suggest that users need in this case are defined as how users perceive their situational environment as being unclear in conjunction with how they see information as helping them most effectively clarify or make sense of these circumstances. Further, they state that the notion of individuals ‘making sense’ requires an understanding of their perceptions of past experiences, present situations, and future conditions (Schamber et al., 1990). This means that users hope that their various needs which have been modeled by their environment should be met by whatever information produced or suggested.
What type of relevance does this study intend to address? – Topical, Pertinence and Situational relevance. This study first meets the topical relevance needs of users by finding similarity between users’ topics of interest and news items and then goes on to offer users relevant information by finding a match between the stories and the user’s needs at any time and by considering user’s knowledge and experience. Therefore, it is important that users can adjust their preferences at any time via the system.

2.3 Serendipity Explained

Personalization techniques have been used in the past to help people deal with the continuous growth of information available on the worldwide web. But this results in what I would call “over-personalization” which exposes users to only information the system assumes they would find interesting. “Personalized systems achieve efficiency in the provisioning of highly focused information, but at the cost of limiting the dynamic nature of user interests” (Fan, Mostafa, Mane, & Sugimoto, 2012). Eli Pariser in 2011 called this the ‘filter bubble’. “Pariser ventures to expose the defect of personalization by revealing how it impacts what the user sees, how it controls their thoughts and actions, and how it gives excessive power to entrepreneurs and computer scientists” (Hsu & Shigetoshi, 2011). Personalization is usually carried out by tech giants who have access to a broad range of user information which they have gathered from user interactions with their systems. “These internet giants argue that the world of personalization gives internet users more control over their future, but Pariser claims that it is just the opposite” (Hsu & Shigetoshi, 2011). Further, “Filter bubbles are formed by the algorithms social media sites like Facebook use to decide which information to show users, based largely on their tastes” (Sally, 2016). i.e., the algorithms, while trying to understand a user’s interest and provide information, confines that user to a type or source of information. Most often the filter bubble begins when the user clicks on a link which often signals interest, these interests are saved as
browser cookies which interact with the algorithms developed by tech companies. The algorithm then assumes and defines the user based on these interests and makes future recommendations. This has been the method used by Facebook to make recommendations to users. This approach to personalization limits learning for users. Like Pariser says – learning in the filter bubble is only limited to what a user already knows and not what is hidden or unknown. Personalization is good, but what are better ways to solve digital age information overloading using personalization while taking to cognizance the curiosity of users?

In his book “The filter bubble: what is the internet hiding from you”, Eli Pariser explained that in 2010, Google rolled out personalized version of Google News. However, that the ‘Top Stories’ section provided stories that are locally and only relevant to a user based on interests that such user has demonstrated using Google and article clicked in the past (Pariser, 2011). Google does so in the hope to provide (as the CEO says) a ‘very personalized’ and ‘very targeted’ content. However, Eli indicates that this is not the only problem with this news technology. Google news is still a “hybrid model driven in part by the judgment of some professional editorial class” which gears further towards been biased in the news provided (Pariser, 2011).

In avoiding these problems, it is beneficial that when information retrieval systems are built, serendipity must be examined. Serendipity has become a trendy topic in the information retrieval domain especially among recommendation system where there is need to amplify experience for users. Serendipity is a natural part of a human information-seeking process that can lead to unexpected and useful discoveries (Fan et al., 2012). According to (MacCatrozzo, 2012): “It is the ability to make fortunate discoveries by accident”. Serendipity is concerned with novelty unexpectedness and surprise. The goal is to recommend to users interesting contents which they wouldn’t have found by themselves. Alongside novelty comes the need to provide relevant information to users.
I introduce serendipity into the PCM by using user-oriented collaborative filtering technique in conjunction with content-based filtering technique, both of which are discussed extensively in Chapter 4. One of the solutions which Eli Pariser suggest for ‘bursting’ the filter bubble is that one should use multiple websites for diversification purposes. This is like collaborative filtering where ratings of other users are used to recommend items for an active user. A non-serendipitous news recommendation system would suggest a story and when it finds that a user is interested would recommend only stories with similar content or sources, but a serendipitous environment would also provide stories or sources with opposing views.
Chapter 3

News Content Aggregation

One usefulness of content aggregation is that it helps users collate information from the various sources into one place. (Chowdhury & Landoni, 2006) defined web content aggregator as “individual or organization that gathers web content and applications from different online sources for reuse or resale”. Aggregated content may include videos, blog posts, podcast or news stories. Advantages of content aggregation include providing diversity on a subject for users, supplies many contents, it cost less to aggregate contents and fosters personalization. News aggregation websites can be classified into two types. Type 1 include those sites that just aggregates news content independent of users’ needs or demand and Type 2 include those that gather, process and distribute content-based on the needs of users. This study focuses on news content aggregation and can be classified as using Type 2 aggregation approach. In section 5.2, I explained the two processes (Aggregation and Recommendation) as the processes to be followed by the PCM.

News contents are always changing both in formats and language; therefore, aggregation is very useful. Also, it saves users the time in retrieving news from the various sources. A survey by (Chowdhury & Landoni, 2006) was carried out by questionnaire and interview to “gather insights into the user-requirements regarding functionalities that users would like available in an ideal news aggregator service”. The users were asked to complete a two-part questionnaire. First, they were to provide information regarding what they would expect from a news aggregator and second, to go through five different aggregator services (Google news, Newsburst, Newsburst is a personal information tool for News.com readers
Headline spot, TVEyes\textsuperscript{11}, and Awasu\textsuperscript{12}) and give feedback on user experience with any of those services. Findings from this survey shows that a good aggregator system should essentially have a high quality and reputable source, personalization, contents in chronological order, user-friendly interface, relevant information, variety of subjects, subject categorized, news alert service, ease of use, categories description, description of the News channels (e.g. sky sports – for entertainment news coverage), useful, helpful, interesting, and relevant information. This study does not explore every one of these expectations but focuses on developing a platform that include most of them.

3.1 RSS Feeds

Most news aggregators use either RSS feeds to get their contents or just scrap contents from available URL on the web and places these content on their website. This section distinguishes between both methods and states possible similarities.

RSS (Rich Site Summary) is just a simple XML syntax for describing a channel of recent additions to a website. These additions may be news items, blog updates, library acquisitions or any other discrete information elements (Judith, 1983). It is an XML based document that assists content syndication (Gill, 2005). This XML file is available on a website and is made available for subscribers using RSS reader. An RSS reader, for example, checks the site where the XML file resides at intervals for feeds updates. The feed returned contains sections such as headlines, content overview, date, and author.

\textsuperscript{11} TVEyes Inc. is an international broadcast media monitoring company based in Fairfield, Connecticut - https://www.tveys.com/.

\textsuperscript{12} Awasu is a state-of-the-art feed reader that comes loaded with features for both casual personal use and professional, high-powered information management - https://www.awasu.com/
Netscape\textsuperscript{13} created the first version (RSS 0.9) of RSS in March 1999 for sharing news and information. Website users would see customized content that was often updated, but the customization would happen through technology, in the background, without the need for human intervention. (Gill, 2005). The format for this first version was in Resource description framework (RDF), but in July 1999 Netscape removed support for RDF elements which simplified the technology and was called version 0.91. America Online (AOL), another tech company acquired Netscape’s business portal in April 2001, they dropped support for RSS and all its features, documentation and supporting tools. Dave Winer of UserLand\textsuperscript{14} and RSS-Dev Working Group\textsuperscript{15} carried on with the specification and developed tools which had support for reading and writing RSS (Sikos, 2011). In December 2000, UserLand released RSS 0.92 “a minor set of changes aside from the introduction of the enclosure element, which permitted audio files to be carried in RSS feeds and helped spark podcasting” (Sikos, 2011). However, in 2002, while trying to pay homage to v0.90 initially released by Netscape, RSS-Dev developed version 1.0 which had support for RDF previously removed. UserLand was not interested in the RDF syntax but carried on with the existing “parallel development path” (Gill, 2005). Between 2001 and 2003, UserLand released numerous version which all together culminated version RSS 2.0.1 (Gill, 2005) but removed the type attribute which was previously introduced in V0.94 with news support for namespaces (Sikos, 2011). While the technology continues to grow, several tech companies have plunged in support of it and have adopted it for their news feeds.

\textsuperscript{13} Netscape is a brand name that was once associated with the development of the Netscape web browser, a series of web browser's that includes the Netscape Navigator. The Netscape brand is owned by Oath, Inc., a subsidiary of Verizon Communications - https://en.wikipedia.org/wiki/Netscape
\textsuperscript{14} UserLand is a US-based software company, founded in 1988, that sells web content management, as well as blogging software packages and services. - https://en.wikipedia.org/wiki/UserLand_Software
\textsuperscript{15} RSS-DEV Working Group was the outgrowth of a fork in RSS format development. The private, non-commercial working group began with a dozen members in three countries, and was chaired by Rael Dornfest, researcher and developer of the Meerkat RSS-reader software. - https://en.wikipedia.org/wiki/RSS-DEV_Working_Group
RSS has in recent years played a significant role in News provisioning systems. The primary goal of any News recommender system is to “promote the most relevant stories to a user based on their learned or stated preferences or their previous news consumption histories, helping the user to keep up-to-date and to save valuable time sifting through less relevant stories” (Phelan, McCarthy, Smyth, Phelan, & McCarthy, 2009). In answering the question of why newspapers have rapidly adopted RSS, (Gill, 2005) says that “data suggest that the growth in blog readership, coupled with easier-to-use technology for reading RSS feeds, appealed to editors and publishers eager to bring new eyeballs (and page impressions) to their online news sites”.

Some aggregators give users an opportunity to decide on the frequency of getting stories from an RSS feed and some RSS feed sources “will often ban an IP address if it attempts to poll more frequently than every 30 or even 60 minutes” (Judith, 1983).

Though it is a dominant syndication technology for news sharing content because of its head start, it does have its fall backs which include reasons for its none use in this study. Data integration cannot be adequately handled by just using web services, several web databases and tools do not support web services, and existing web services do not cover for all possible user data demands (Glez-Peña, Lourenço, López-Fernández, Reboiro-Jato, & Fdez-Riverola, 2014). This also applies to a syndicated news with RSS. A minimal RSS feed (see Fig. 1) comprises limited meta data such as links, title & description. So, when most news feed syndicates news, the title field would be the title of the story, the link would be a URL to the news source, and the description would be a summary of the news story (most likely the first paragraph). The system developed in this study works well with the complete content of the news story which would allow users to rate and recommend a story and not just a summary. This makes relying on RSS a poor choice for implementation.
3.2 Content Scraping

Content scraping can be categorized as data scraping and web scraping (screen scraping). Data scraping is a term used to refer to the extraction of data from a digital file. While Web scraping is the set of techniques used to automatically get some content from a website instead of manually copying it and combining the content in a systematic way (Daniel, Lourenco, Lopez-Fernandez, Reboiro-Jato, & Fdez-Riverola, 2013). This thesis focuses on getting information from various sources by scraping them and presenting them in one location. This has been used by aggregators and non-aggregators for years and has been a useful tool for these purposes. For example, in content aggregation, it is being used to structure similar data from different sources to combine them in a single source (Krijnen, Bot, & Lampropoulos, 2004). Also in the financial sector where there is much competition, web scraping has been massively utilized to scrap marketing prices of competitors, in the research field, it has been used to scrape pieces of literature from defined sources. (Haddaway, 2016) Indicates that in research, web scraping makes “searches of multiple websites more resource-efficient” and “drastically increases transparency in search activities”. In comparing literature reviewing using Google scholar with web scraping,
(Haddaway, 2016) says that a – reviewer would have to download hundred and thousands of search results for later screening but automation of activities that would otherwise be undertaken by hand is of great value to researchers.

Web scrapers work by scanning through the structure (DOM – Document Object Model) of an HTML page. They use bots to get data from internet servers by mimicking an interaction between the web servers and a human in a conventional web transversal (Daniel et al., 2013). The parsed contents are then structured in the way it suits best for the underlying project (Krijnen et al., 2004). This ability to structure data depending on the need of the application is a major advantage of web scraping and explains why it is sometimes better to scrape data instead of using a provided public API (Krijnen et al., 2004). The method is used as an alternative to syndication in this study since it solves the biggest problem with RSS as it relates to this work. Consequently, the PCM developed saves the scraped content to a NoSQL (particularly MongoDB) database and then recommends them for better use in this study (see section 5.1). Overall, though this method of information gathering has in the past witnessed major legal pressure “which could endanger the practice of web scraping, but raising public awareness around this issue might positively influence the debate” (Krijnen et al., 2004).
Chapter 4

Recommendation Processes

In this section, I explain the recommendation model used by the PCM. This study employs a hybrid solution for recommending news to users. This hybrid solution includes both collaborative filtering and content-based filtering. Both approaches have their pros and cons; I am shielding the disadvantages of one method by utilizing the advantages of the other.

4.1 Collaborative filtering

The developers of one of the first collaborative filtering systems (*Tapestry*) came up with the name and has since been used to refer to any system where people rely on other people for contents like movies, documents, books, etc. Collaborative filtering (CF) approach recommends items to users based on similar users or items. Recommendations are based either on the similarity between item which a user has liked/rated in the past or similarity between users who have similar taste to an active user.

This filtering approach is classified as either memory-based or model-based. CF analyses relationships between users and interdependencies among products, to identify new user-item associations. (Hu, Koren, & Volinsky, 2008). The only required information for CF is a past user behavior which may include their ratings. It uses a database of user ratings on items to predict additional items a user might like. “CF has its roots in information retrieval and information filtering techniques and employs many of the same principles” (Sridharan, 2014). “The goal of a collaborative filtering algorithm is to suggest new items or to predict the utility of an individual item for a user based on the user's previous likings and the opinions of other like-minded users” (Sarwar, Karypis, Konstan, & Riedl, 2001).
Table 2: User-item matrix

Table 2 shows a typical example of a user-item association where movies with different titles are rated, and Dmitri is an active user to get recommendations. Item ratings can either be in the form of ‘like/dislike' or numbers usually from 1-5 as in Table 2.

The rating data above are used to calculate the weight/similarities between users (or items) and predictions are made based on the estimated values. But problems exist which demand the need for a hybrid approach. One of such problem is ‘cold-start’. “Cold start happens at the beginning of the interaction when the system does not have enough user data to provide appropriate adaptation” (Kuflik, Vania Dimitrova Tsvi, David Chin Francesco Ricci, 2012). In scenarios relating to ours, the system may suffer from insufficient or no rating on a new story thereby limiting the possibility of it getting recommended. Also, there is the ‘sparsity’ problem where items are more than users, and therefore only a few set of items would have been rated by users.

Content-based filtering (CB) becomes superior in helping solve these problems. But there is a major problem with content-based - it isn’t so useful in cases where there is less content. Also, it is known to recommend items already known to users. Using CB alone hinders relevance and serendipity. See section 4.3 for details on CB. What differentiates collaborative filtering from
content-based is its reliance on user’s behavior while content-based uses meta data comprising item attributes or a user’s preferences. CF is divided into model-based and memory-based collaborative filtering methods.

4.1.1 Model-based collaborative filtering

“The design and development of models (such as machine learning, data mining algorithms) can allow systems to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering tasks for test data or real-world data, based on the learned models” (Su & Khoshgoftaar, 2009). This technique tries to ‘guess’ to users items they haven’t seen before. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items (Sarwar et al., 2001). In summary, model-based techniques use algorithm (e.g., Bayesian Networks, clustering) to find patterns and make prediction on ratings by a learning process. i.e. it trains items for a user using object vectors which are then used to creates a model of user ratings for future recommendations.

Bayesian Networks create a model using a tree-like structure where nodes of the tree represent user information and are usually more useful in situations where user’s preferences change slowly compared to the time to build the model and may be less efficient where user preferences change rapidly. Clustering works by clustering similar users in same category and makes an estimation of the probability that a user is part of a category, after which the probability of ratings is predicted. Once the clustering is complete, however, performance can be outstanding, since the size of the group that must be analyzed is much smaller (Sarwar et al., 2001).
Collaborative filtering has also witnessed the use of neural network in creating models which exceed the advantages of traditional collaborative techniques because it can learn complex input/output relationships and can be used in cases where information is inefficient or incomplete. In solving the problems with conventional collaborative filtering method (M. W. Kim, Kim, & Ryu, 2004) proposed the use of collaborative filtering based on neural network (CFNN) with trained multilayer perceptron to learn a correlation among preferences of the target user and the reference users which resulted in two models called user model (U-CFNN) and item model (I-CFNN). For example, “In the U-CFNN model the input nodes correspond to the users’ preferences and the output node corresponds to the target user’s preference for the target item” (M. W. Kim et al., 2004). In comparing their approach to other existing methods such as memory-based K-NN method, their method proved to have significant increase in performance since it uses neural network to “integrate additional information and selection of the reference users or items based on similarity” (M. W. Kim et al., 2004).

4.1.2 Memory-based collaborative filtering

Commonly known as neighborhood based collaborative filtering because it computes similarity amongst neighbors in-memory. Here the memory is loaded with entries from the database and used directly to make recommendations for users. Algorithms used in memory-based are based on the fact that similar users display similar patterns of rating behaviour and similar items receive similar ratings (Aggarwal, 2016). User-based and Item-based collaborative filtering are the two types of memory-based collaborative filtering that exist. After neighbors are found, algorithmic approaches are utilized to make predictions for an active user by the combining the weights of all neighbors in the neighborhood. Memory-based techniques rely on using similarity measures such as Jaccard coefficient, cosine similarity measure or Pearson correlation to
find similarities. In Table 3, I distinguished between memory-based, model-based and a combination of both filtering approaches by stating their pros and cons and the techniques specific to each.

4.1.2.1. **User-oriented collaborative filtering**: Here, “the principle of CF is to aggregate the ratings of like-minded users” (Kuflik, Vania Dimitrova Tsvi, David Chin Francesco Ricci, 2012). Items are recommended to an active user based on ratings/likes of other users who have liked same items as the active user. So, in user-oriented approach, there is the assumption that because user A and user B have liked similar items in the past and user B also likes item i, then item i should be recommended to user A. This study employs this neighborhood approach where similar users (known as neighbors) are first gathered then the best prediction for an active user is made based on ratings of the neighbors. In doing so, users profile play a major role because they differ both in personality, interest and demographics. As (Bonhard & Sasse, 2006) suggests “drawing on similarity and familiarity between the user and the persons who have rated the items can aid judgement and decision making”. Studies have shown that users demographics is a contributing factor to the type of information they consumed. For example (Uitdenbogerd & Schyndel, 2002) observed that factors affecting individual music preferences include age, origin, occupation/profession, socio-economic background, gender, personality factors (introverts, extroverts, aggressive or passive) and by utilizing them, they can be used to enhance recommendation. I believe this can also be useful for news recommendation. But this study focuses on using just the users’ profession to improve the PCM recommendation process; the algorithm first finds neighbors based on the similarity between profiles and then makes items prediction based on rating weights and users’ profession. Since the central interest for users is relevant stories, a user’s interests and ratings would play a big part in the recommendation. The confidence level for each item is estimated based on ratings and users’ profession.
Usually, to determine similarity measure in collaborative filtering, similarity algorithms are employed.

Another justification for using users oriented technique in this study is the need for the serendipity of information. Richard Jaroslovsky (“Rich Jaroslovsky: Part 1 - The Future of News: The Future of News,” n.d.) in an interview with The Future of News stated the need for serendipity to be incorporated into digital new content – “in the newspaper age, what made a good newspaper? The answer to that question was that there was something for everyone, you were discovering things through a process of serendipity, you stumbled on news stories you didn’t know you would be interested in but found them to be interesting”. (Sridharan, 2014) defines serendipity as the accident of finding something good or useful while not specifically searching for it. Serendipity in this study is achieved by using the user-based filtering rather than item-based.

4.1.2.2. Item-oriented collaborative filtering: Invented by Amazon in 1998, “item-based apply the same idea as user based, but use similarity between items instead of users” (J. Wang, Vries, & Reinders, 2006) and the similarity is calculated based on users’ behavior. The item-based method works by exploring associations between items. Items are recommended to a user based on items that the user previously rated in the past. “To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together” (Greg, Brent, & Jeremy, 2003). Thus, a products matrix can be built by iterating over all items-pairs and computing the similarity between them. Thus, an item-based (i) finds every pair of items rated/liked by the same person (ii) measure the similarity of their ratings across all users who rated both (iii) sort the item by similarity value (iv) make recommendations to users. For example, to compute similarity between items \( i \) and \( j \), users who have
rated both items are isolated and then a similarity computing technique (e.g. explained in section 4.2) is applied to get the similarity $S_{i,j}$. In computing the similarity, the meta-data of the items content are not required or used, rather only the users’ history of rating is used.

But, this method is much faster because there is no need to look for neighbors before recommendations are made. The problem with this approach is that there is a tendency for users always to see items which they have previously been recommended. “Once the most similar items are found, the prediction is then computed by taking a weighted average of the active user’s ratings on these similar items” (Sarwar et al., 2001).

4.2 Similarity Measures

Before any recommendation is made in memory-based CF, similarity either between user or items is first calculated. In the following, I explain two of the most popular similarity computing methods used (Pearson correlation & Cosine similarity measure).

4.2.1 Pearson correlation:

Pearson’s coefficient is an index of the strength of linear relationship between two variables using their covariance. It always has values between -1 to 1.

Here, similarity $S_{u,v}$ between user $u$ and $v$ is found by computing the Pearson correlation coefficient between both users.

\[
S_{u,v} = \frac{\sum_{i=1}^{n} (r_{u,i} - \bar{r}_{u})(r_{v,i} - \bar{r}_{v})}{\sqrt{\sum_{i=1}^{n} (r_{u,i} - \bar{r}_{u})^2} \sqrt{\sum_{i=1}^{n} (r_{v,i} - \bar{r}_{v})^2}}
\]
Where \( \bar{r}_u \) and \( \bar{r}_v \) is the average rating value of both users respectively. User \( u \) rating on item \( i \) is \( r_{ui} \) and user \( v \) rating on item \( i \) is \( r_{vi} \). \( \bar{r}_u \) and \( \bar{r}_v \) represent average ratings of the co-rated items by user \( u \) and \( v \) respectively. \( n \) is the number of neighbors. For the table 2, where users (Teemu, Arttu, Joonas, Dmitri) rates movies from 1-5, consider a case where the active user is Joonas. To predict item for Joonas based on the ratings on table 2 we must first find the similarity between Joonas and all other users. We can find the similarity between Joonas and Arttu by using the movies which they have both rated (in this case: ‘Gone’ and ‘As the crow flies’) with rating values 2, 5 and 5, 4. The Pearson correlation value between Joonas and Arttu using Pearson coefficient is -1.

4.2.2 Cosine similarity.

Unlike Pearson correlation, this is bound by 0 and 1 and represents the angle between two vectors. When applied to collaborative filtering, cosine similarity treats each user or item as a vector of rating frequencies and computes the cosine of the angle formed by the vector of rating frequencies. It is represented as:

\[
\cos (u, v) = \frac{\sum_{i=1}^{m} u_i v_i}{\sqrt{\sum_{i=1}^{m} u_i^2} \sqrt{\sum_{i=1}^{m} v_i^2}}
\]

When to table 2, we can find a similarity between Joonas and Arttu where \( u, v \) represent the both users for which similarity is to be found. \( \sum_{i=1}^{m} u_i v_i \) represents the common ratings by \( u \) and \( v \), \( \sum_{i=1}^{m} v_i^2 \) and \( \sum_{i=1}^{m} u_i^2 \) represents sum of ratings by \( v \) and \( u \) respectively. When calculated, the cosine similarity for (Joonas and Arttu) is 0.87.
4.3 Prediction Measure

Getting prediction is the most important step in recommendation systems, and it comes after similarity measures have been calculated. **Weighted sum** and **Regression**\(^{16}\) are the two techniques mainly used for calculating prediction for users. Weighted sum is the technique used in this study and the prediction on an item \(i\) for a user \(u\) is computed by computing the sum of ratings by other similar users on item \(i\). That is, “...a subset of nearest neighbours of the active user are chosen based on their individual similarities with the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user” (Xiaoyuan & Taghi, 2009).

For user based filtering, to predict item \(i\) for active user \(U_a\), the weighted sum formula is given as:

\[
P_{u, i} = \frac{\sum_{all\ similar\ users, v} r_{v, i} \cdot S_{u, v}}{\sum_{all\ similar\ users, v} S_{u, v}}
\]

where \(r_{v, i}\) is the rating of a similar user \(v\) on an item \(i\). \(S_{u, v}\) represents the similarity between \(u\) and \(v\). \(\sum S_{u, v}\) represents a summation of the similarity values between similar users and \(u\).

\(^{16}\) This approach is like the weighted sum method but instead of directly using the ratings of similar items or users it uses an approximation of the ratings based on regression model. (Sarwar, Karypis, Konstan, & Riedl, 2001)
<table>
<thead>
<tr>
<th>Techniques</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory-based CF</td>
<td>*Easy to implement</td>
<td>*Heavily dependent on human ratings</td>
</tr>
<tr>
<td></td>
<td>*Provide more accurate predictions that content-based</td>
<td>*Sparsity problem (cold user) when new user or item enters the system and has never been rated</td>
</tr>
<tr>
<td></td>
<td>*Content of the item is not necessary to be considered</td>
<td></td>
</tr>
<tr>
<td>Model-based CF</td>
<td>*Prediction performance more intelligent than in memory-based.</td>
<td>*Building a learning model is not so easy.</td>
</tr>
<tr>
<td></td>
<td>*Sparsity problem is handled better since models are used.</td>
<td>*also relies on human activities to create a model.</td>
</tr>
<tr>
<td></td>
<td>*offers better rationale for making recommendations</td>
<td>*where reduction models are used, user data may be lost</td>
</tr>
<tr>
<td>Hybrid Recommender</td>
<td>*loss of information is overcome.</td>
<td>*quite complex and challenging to implement.</td>
</tr>
<tr>
<td>systems</td>
<td>*sparsity problem is also handled better with improved performance</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Common filtering techniques pros & cons
4.4 Content-based filtering:

In content-based filtering (CB), various candidate items are compared with items previously rated by the user, and the best-matching item(s) are recommended (Kumar Nandanwar & Pandey, 2012). What distinguishes CB from CF is that while CF uses similar users or items to make a recommendation to an active user, CB focuses on just the textual content of the items to be recommended and makes recommendation based on item features. Therefore, “text documents are recommended based on a comparison between their content and a user profile” (Balanovic, Marko; Shoham, Balabanović, & Shoham, 1997).

In CB, the importance of items is based on how much they relate to the profile of an active user; once attributes have been specified for users and attributes for the items is available (usually in the form of keywords), a similarity function is then used to measure the distance between items and users. There are various ways to specify knowledge about users’ attributes or needs; either implicitly or explicitly. Implicit knowledge is generated from the users i.e. knowledge is in users’ mind, while explicit knowledge involves such knowledge generated from process like using neural network. For example, (Kumar Nandanwar & Pandey, 2012) carried on a research that does content based recommendations using Self Organizing Maps (SOM\(^\text{17}\)) and Latent Dirichlet Allocation (LDA\(^\text{18}\)). Their approach elicits a shared topical structure from the content based recommendation system using LDA efforts of multiple users (Kumar Nandanwar & Pandey, 2012).

\(^{17}\) The Self-Organizing Map also called the Self-Organizing Feature Map is one of the most popular neural network models. It is based on unsupervised learning, which means that no human intervention is needed during the learning and that little needs to be known about the characteristics of the input data - http://users.ics.aalto.fi/jhollmen/dippa/node9.html

\(^{18}\) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar - https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
One example of systems using CB is Pandora. Pandora meets the music needs of listeners using curated genre stations. Pandora is powered by the *Music Genome Project* which uses over 450 attributes to define a song and an algorithm to organize them. Each station consists of user interest which include artist or songs they prefer as indicated by the user. Based on these preferences Pandora plays songs which the user might also like. An active user can then rate a song by ‘thumbs up’ or ‘thumbs down’. This process of rating helps to refine the user’s station. A ‘thumbs down’ on a song means the user is not interested in hearing that song again or similar type of song. A ‘thumbs up’ means the opposite. Pandora employs CB filtering techniques for personalizing contents to listeners. It relies only on the user station and the attributes of a song for making predictions. There is an algorithm that combines a stations’ attributes into one value and then compares such value with the corresponding value of songs in that database.

In most cases, users’ profiles are built up by analyzing content rated by them without their involvement (explicit). But this is not the approach used in this study. Instead, the system allows for users-specified preferences selected from a list of topics (implicit). In recommending news stories to users, implicit data collection from the users is more accurate because stories are recommended according to data represented in a user’s profile. CB works well when users have enough data to be used for recommendations.

Other advantages of CB over CF are that recommenders exploit solely ratings provided by the active user to build the profile (Lops, de Gemmis, & Semeraro, 2011). Also, CB solves the *new item or sparsity problem* associated with collaborative filtering. But as previously stated the content-based filtering comes with its shortcomings including that it is almost impossible to recommend unexpected items to users. It only recommends items based on user’s profile.
4.5 Building Data

Building up user’s data in CB or CF can be done either explicitly or implicitly. In an explicit feedback mechanism, users are asked to provide precise information on their preferences such as rating an item, dropping a comment, requested to do searches. Netflix, for example, asks users to rate a completed movie. Most websites use a thumbs up/down buttons while some use implicit feedback mechanism where user’s behavior is observed. Examples of this include search history, a user’s buying records, search patterns or previous purchases.

Facebook uses an implicit approach in data collection for recommending pages or groups a user should join. But with Facebook’s large data set, they limit recommended pages or groups to those who have passed a threshold – above 100 likes or members. Facebook takes note of both positive and negative implicit feedback. Though implicit feedback removes the hassles experienced when a user may not provide feedback, it becomes a problem when user's negative feedback cannot easily be inferred. That is, users profile could be misinterpreted since they do not provide a positive or negative response. In Facebook, negative feedback is not readily available therefore disliking a page or leaving a group does not necessarily mean the user has given a negative feedback. This makes implicit feedback unreliable and sometimes inaccurate.

This study utilizes an explicit method for gathering users' preferences and feedbacks on stories. To provide feedback on a story, a user just rate the item, where rating values are from 1 – 5. ‘1’ signifies – less relevant and ‘5’ signifies – most relevant.
Chapter 5

Technologies Used in This Thesis

This chapter discusses the different technologies used to develop the PCM. The PCM is a client-server application which runs in a browser and is accessible via a URL. It consists of a frontend which helps the user interacts with the system and a back end. I do not intend to explain each of these technologies in detail but give an overview of what they are and how they are used in this study.

5.1 Hypertext Transfer Protocol (HTTP)

This is the most prominent protocol used on the internet today and it defines how messages are transmitted and received between clients and servers. It is stateless, meaning that when information is sent to a server, the server does not keep or retain any information about the previous request; therefore, any established connection between a client and a server is lost after that transaction is complete. Though this reduces the complexity of the interaction process between a browser (client) and a server, this is a shortcoming of the HTTP. This shortcoming makes it difficult to build intelligent web applications based on users input. But this has been addressed using HTTP cookies, session variables, variables which are hidden (especially when using web forms). HTTP as shown in Fig. 2 is very much based on a request and response paradigm and “such a paradigm is robust and flexible enough to allow most kinds of data transmission tasks besides the fundamental web browsing activities” (Li, Moore, & Canini, 2008). “HTTP has accommodated remarkably more different kinds of activities than web-browsing, for example sending and receiving email, file downloading and sharing, instant messengers and multimedia streaming if they can follow the request or response paradigm’ (Li et al., 2008).
An HTTP request comprises of different entities including the request URI which is a Uniform Resource Identifier which helps identify a resource, request method which indicates the operation to be performed on the request URI. Such operations include either a POST, PUT, DELETE, GET, OPTIONS, TRACE, CONNECT and HEAD. HEAD is very much like GET but contains the header section. GET uses a URL to retrieve information from a given server without making any other operation on the data. POST is usually used in web forms to send data to the server. PUT replaces contents of target resources with new data. OPTIONS define dependable options for the target resource and TRACE performs loopback test for the request. A request header allows a client to pass more information to the server about the request or request sender. Fig. 3 shows an example of an HTTP request. After receiving a request from a client, an operation is performed, and a response is sent to the client. This response mainly contains components including HTTP response header (contains useful information about the server environment), the response body, protocol version and status code. Fig. 4 is an example of HTTP response header.
In the PCM, when a user visits a page or updates profile or rates a piece of news etc., a client-server communication is established and HTTP is being utilized.

### 5.2 JavaScript Object Notation (JSON)

JSON, an alternative to Extensible Markup Language (XML) is a format for structuring data. “JSON was derived from the ECMAScript Programming Language Standard and defines a small set of formatting rules for the portable representation of structured data” (Crockford, 2006). It can be used to represent primitive types such as Strings, Numbers, Boolean and Null.
and non-primitive types such as Object and Array. In a client-server environment, JSON is used to transfer data between the client and the server. It is made up of a key-value pair where the keys are enclosed in quotation marks and the value is any one of the types above. “An array structure is represented as square brackets surrounding zero or more values” and “an object structure is represented as a pair of curly brackets surrounding zero or more name/value pairs” (Crockford, 2006).

In our system, when an API request is made to a news provider or contents are scraped from a URL or IBM Watson Natural Language Understanding API is utilized, responses are in the form of JSON, operations are then performed on these data and presented to the users. Fig. 5 is a JSON response containing data for making recommendations for an active user.

![Fig. 5: Example of a JSON Data](image-url)
5.3 Application Programming Interfaces (API)

In web development, an API is generally a set of code features (e.g. methods, properties, events, and URLs) that a developer can use in their apps for interacting with components of a user’s web browser, or other software/hardware on the user’s computer, or third-party websites and services (Mozilla, 2017). An API can comprise specification that defines how a program should interact with it and an endpoint to be queried for data. Therefore, one program ‘publishes’ an API and another program ‘consumes’ or ‘calls’ it.

The PCM consumes data from News API which is a simple API that returns JSON data consisting of news headline from over 70 different sources including BBC, Blomberg, Reuter, CNN, Independent, Guardian, ESPN, MTV News, etc. It provides two types of endpoints: https://newsapi.org/v1/articles (provides a list of news metadata) and https://newsapi.org/v1/sources (returns the list of news sources available on News API). The PCM utilizes the former since it provides articles about different domains, e.g., Sports, Entertainment, Technology, Politics, etc. An API key and a source parameter (as in Fig. 6) are needed to make a GET request to this endpoint and the response is a JSON data (Fig. 7) containing an array of articles with fields about each story. Optionally, a sortBy parameter is provided which specifies the type of list to be returned either top, popular or latest.

![GET](https://newsapi.org/v1/articles?source=the-next-web&sortBy=latest&apiKey= {API_KEY})

Fig. 6: A News API endpoint
In this section, I would describe the technologies used to develop the interface for user interaction and for interacting with servers. They are divided into frontend and back end technologies. The frontend of any application is what the user sees and interacts with and mostly focuses on user experience. HTML5, CSS3, and JavaScript are frontend technology used in this study. HTML (Hypertext Markup Language) is the language which the browser understands and uses to present information to users. HTML by itself is plain without CSS. CSS (Cascading Style Sheet) is used to enhance the presentation and display of web pages for users. Therefore, while HTML deals with the structure, CSS deals with the appearance. JavaScript is a language used to add functionalities to a web application. Though not solely a frontend technology, it is mostly used to develop fast frontend programs that run on the user machine (browser).

JavaScript has a variety of frameworks and libraries used for making development easier and quicker. AngularJS developed by Google is one of such framework and is being utilized for the PCM. AngularJS allows for the extension of HTML vocabularies resulting in easy to read, expressible and quick to develop applications. It also fosters single-page applications by helping

Fig 7: A News JSON API response

5.4 Frontend and Backend Technologies

{  
    status: "ok",  
    source: "the-next-web",  
    sortBy: "latest",  
    - articles: [  
        - {  
            author: "Rachel Kaser",  
            title: "US Government demands details about all visitors to an anti-Trump site",  
            description: "Web-hosting company Dreamhost revealed this week that it was locked in a legal battle with the US Department of Justice over a protest site -- and the DoJ wants all the information ...",  
            urlToImage: https://cdn0.thenextweb.com/wp-content/blogs.dir/1/files/2016/03/department-of-justice-doj.jpg,  
            publishedAt: "2017-08-15T18:26:25Z"  
        }  
    ]
}
to create dynamic views without page reload. The goal of AngularJS is simplification with support for the MVC (Model View Controller) programming design. “Angular is one of the only major frontend frameworks that utilize plain old JavaScript objects (POJOs) for the model layer. This makes it incredibly easy to integrate with existing data sources and play with basic data”. (Jain, Mangal, & Mehta, 2014). One reason for me choosing AngularJS for this project is because of its role in developing MEAN stack applications which makes JavaScript the sole language for both the frontend and the backend of the system; the ‘A’ in MEAN stands for AngularJS. Moreover, AngularJS works well with top-down programming approach, focuses more on features, pairs with AJAX effectively, and has massive support base on the internet.

**NodeJS** is used for the backend. It is an open source JavaScript framework that allows JavaScript to run on a server. Also, it is built on Chrome's V8 engine and utilizes a non-blocking, event-driven, I/O efficient and fast model. This means that operations are handled by events handlers or events callbacks and no process is blocked while waiting for another process to complete rather multiple processes can execute in parallel. Concurrency is a prominent feature of NodeJS and is carried out using the event loop. “Node application rely on big processes with a lot of shared states and offers developers a powerful way to develop networking applications that will perform well compared to mainstream solutions” (Rauch, 2012). As mentioned earlier, choosing NodeJS for the backend makes development easier because the same language runs the frontend. NodeJS is the “N” in MEAN.

**ExpressJS** is a web framework used with NodeJS and helps to publish applications as websites; it is the standard server framework for Node applications. ExpressJS is lightweight and helps in managing everything in a NodeJS application from routing to handling clients request to presenting views. It is based on NodeJS Connect components and these components are called ‘middleware’. “The framework’s middleware architecture makes it easy to plug in extra features
with minimal effort and in a standardized way” (Node.js Foundation, 2017). It solves problems associated with parsing cookies, managing sessions, and parsing HTTP request. Fig. 8 and Fig. 9 below is an example of how the PCM uses ExpressJS to respond to a request. The first parameter of the `app.get` function is the requested URL pattern in string format and the second parameter is a request handler which will be executed any time the server receives a request.

```
var app = express();

app.get('/story/:id', function(req, res, next) {
  res.sendFile(path.join(__dirname + '/public/index.html'))
});

app.get('/login', function(req, res, next) {
  res.sendFile(path.join(__dirname + '/public/index.html'))
});
```

Fig. 8: ExpressJS handling GET request

But before express can handle a request, a server connection must be established. In this case, the server listens on port 8081. ExpressJS is the ‘E’ in MEAN.

```
app.listen(8081, function(){
  console.log("started...", 8081)
})
```

Fig. 9: Starting an ExpressJS web server

**MongoDB** is a document-oriented database system that provides agility, high performance, and ease of scalability. It is a leading NoSQL database system and uses the concept of documents and collections to store data. A collection is the equivalent of tables in a relational database management system (but without a schema) and represents a group of documents. Documents are set of key-value pairs which enforce a schema, meaning that documents in the same collec-
tion have unique structure or fields. MongoDB helps perform CRUD (CREATE, READ, UPDATE, DELETE) operation on data and utilizes BSON (Binary JSON) for it storage which is a JSON style Object format. Fig. 10 conveys how MongoDB schemas are created.

Mongoose is a tool used for modeling objects in an asynchronous environment and helps in representing the application data as Objects which are then mapped to the application database. It is employed in this system to operate on data.

```
var db = require('./db');

var Post = db.model('userProfile', {
username: {type: String, required: true},
email: {type: String, required: true},
date: {type: Date, required: true, default: Date.now},
topic: {type: Array, required: false},
password: {type: String, required: true},
profession: {type: String}
});
```

**Fig. 10: A user profile MongoDB schema**

To insert new data or update a collection, the `save` function is used. Fig. 11 shows how the PCM uses Express to handle a post request and the uses Mongoose to inserts a new user to the DB.

The `find`, `findOne` and `findById` functions are used to get items from that database. `find` finds all document in the database that matches a given query and returns everything if no match is found. `findOne` returns only the first object from the database matching a specified query. `findById` uses a specified `id` to returns an object matching an `id` (see Fig. 13). Line 251 in Fig. 12 goes through a collection containing stories which have been rated by users and returns only one item matching the specified user email and story title then sends the response to the client. Line 256 returns all items in the `ratedItems` collection.
Fig. 11: Mongoose CREATE operation

```javascript
/* handles new user signup functionality and create a new user */
app.post('/users', function(req, res, next){
  if(req){
    var newUser = new PostProfile.Post({
      username: req.body.username,
      email: req.body.email,
      password: req.body.password,
    });
    /* saves new user information in PostProfile table */
    newUser.save(function(err, post){
      if(err){
        Object.keys(err.errors).forEach(function(key) {
          var message = err.errors[key].message;
          console.log('Validation error for "%s": %s', key, message);
        });
        if(post){
          res.json(201, post)
        }
      }
    })
  }
})
```

Fig. 12: Mongoose READ operations

```javascript
app.get('/ratedstories', function(req, res){
  if(Object.keys(req.query).length > 0){
    PostProfile.Ratings.findOne({email: req.query.email, title: req.query.title}, function(err, item) {
      res.send(item)
    });
  }
  else {
    mongoose.model('ratedItems').find(function(err, ratedItems){
      res.send(ratedItems)
    })
  }
})
```
Performing an UPDATE operation is like the CREATE operation but without creating a new object. Instead, a query for the item is made and altered and then saved. `findOneAndRemove` is used to make DELETE operations. MongoDB is the “M” in MEAN.

5.5 IBM Watson Natural Language Understanding

“IBM Watson is an artificially intelligent cognitive computer system capable of processing large amounts of unstructured data and answering to queries posed in natural language” (Khriyenko, n.d.). “Watson builds on the current era of programmatic computing but differs in significant ways” (High, 2012).

Watson can perform Natural Language Processing (which helps in analyzing and understanding unstructured data), Hypothesis Generation (evaluate responses based on available evidence) and Dynamic Learning (ability to get even smarter and improve learning with each interaction). Watson provides sets of APIs which allows developers to build cognition into their apps. One of such API is the **Watson Natural Language Understanding (NLU)** formally AlchemyLanguage. “This cloud-based Watson service offers a suite of Natural Language Processing (NLP) capabilities that make it possible to quickly and easily extract and analyze metadata from an unstructured text” (Susan, 2017).

The PCM relies on NLU’s API by sending HTTP request to it and classifies stories according to existing taxonomy/category. NLU uses natural language processing techniques to extract the
semantic features of a text and then returns a JSON database on specified features to be analyzed in the text. As indicated by (IBM, 2017a) the features include **Categories** (consisting of a 5-level hierarchy and returns three different results of best matches ordered by scores), **Concepts** “identify high-level concepts that aren't necessarily directly referenced in the text” (IBM, 2017a). **Emotions** “analyzes emotion conveyed by particular target phrases or by the document as a whole” (IBM, 2017a) and returns emotions such as ‘sad’, ‘anger’ or ‘happy’. **Entities** find people, places, events, and other types of entities mentioned in the content such as ‘company’, and ‘location’. **Keywords** searches for relevant contents in the text, **Metadata** returns “the author of the web page, the page title, and the publication date” (IBM, 2017a) of a text, **Relations** finds relationships between two entities in a text, **Sentiment** “analyze the sentiment toward specific target phrases and the sentiment of the document as a whole” (IBM, 2017a), **Semantic Roles** “parse sentences into subject-action-object form, and identify entities and keywords that are subjects or objects of an action” (IBM, 2017a).

The PCM uses the **Category** feature. I stated earlier that, this study employs a combination of user-based Collaborative filtering and Content-based filtering technique to make recommendations and that the justification for using such approach is that Collaborative filtering runs into the cold-start problem or data sparsity. Therefore, the PCM uses IBM NLU to accomplish Content-based filtering. NLU analyzes a story, classifies the content into specific category and then returns and an object containing a key called *Label* with a value representing the category of the analyzed content. The PCM then recommends the story to a user if there is a match between the Label and the user’s specified interest. Categories include ‘art and entertainment’, ‘business and industrial’, ‘food and drink’, ‘finance’, ‘health and fitness’, ‘science’, ‘sports’ etc. and each category may be composed of sub-categories about 5-levels deep.
Finally, I should indicate that the Semantic Web technology or Linked Data plays a significant role in IBM Watson cognitive activities and how it deals with unstructured data. The semantic web “provides standards and tools to enable explicit semantics of various Web resources based on semantic annotations and ontologies” (Khriyenko et al., 2005). In a Linked Data environment, related data are connected using URIs and modeled with RDF (in a triple form consisting of Subject, Action and Object) as in Fig. 14. When Watson gets data from sources, it uses Semantic Web technology to reasons over the data. To connect to the NLU API, an Object containing the API endpoint, service version number, IBM Bluemix\textsuperscript{19} username and password is provided as parameter to the service call. Fig. 15 shows this configuration. Also, a parameter Object is needed to make the request. This Object contains values including the type of content to be analyzed (either URL, text or HTML) and the features that are needed. For example, Fig. 16 specifies that ‘category’ feature is to be analyzed from the text and returned. The code snippet in Fig. 17 shows how NLU takes the Object and analyzes it.

**IBM Watson News Discovery** included with IBM Discovery is another service which is continuously updated with news articles and blogs daily. It provides cognitive insights such as keyword extraction, semantic role extraction, semantic analysis, category classification, relations, entity extraction and meta-data information such as authors, publication dates from the news stories. Discovery news is updated continuously with over 300,000 news daily. In addition, discovery news offers historical search for the past 60 days of news data (IBM, 2017b). It is a powerful tool for terms/actions checking (such as election results, IPO, inflation, acquisition etc.) also users can “identify popular topics and monitor increases and decreases in how frequently they (or related topics) are mentioned” (IBM, 2017b). Even with the benefits, it has some cons including that – configuration cannot be adjusted and new documents cannot be

\textsuperscript{19} IBM Bluemix – a cloud service offered by IBM which allows organizations and developers to develop, install and manage their applications on the cloud
added by users. That means, to use this service, one will have to rely solely on IBM Watson’s news content. But this study uses only IBM Natural Language Understanding and flexibly uses other news sources which would allow the algorithm to be enhanced at any time.

Fig. 14: IBM Watson analyzing data using natural language processing

Fig. 15: IBM Watson NLU API authentication

Fig. 16: Specifying the feature to be requested and data to be analyzed

Fig. 17: NLU analyzing features of data
5.6. Mercury Web Parser

The Mercury Web Parser is a product from Postlight Labs that extracts meaningful and pure content from the chaos of any web page—from articles, blog posts, and similar web content (Adam, 2016). Mercury is beneficial in migrating content into new database and generating mobile experiences; using a scaled-back version of an entire website to create a mobile web (Adam, 2016). The PCM uses Mercury client to scrape data from news source pages by making an HTTP request to the news source URL (Fig. 18). This returns a JSON response (Fig. 19) with only relevant content including the content headline, author, main body, corresponding images, etc. The main body is then passed to IBM Watson NLU to be analyzed. To use Mercury Web Parser, an API key is needed.

Fig. 18: API request to parse data from news sources

```javascript
var mc = new MercuryClient('6QuS6lTQWpbFN4bqVDpuHOzyUm1zNghF4K5Rqmb');
mc.parse(item.url)
```

Fig. 19: Mercury Web Parser response
Chapter 6

Algorithm Implementation and System Structure

After detailing the technologies underlying the functioning of the PCM, this section discusses the intricacies by giving a formal representation of the PCM and describes the overall system processes involved in users' interaction with the system. Both Fig. 20 and Fig. 21 showcases how the system processes are organized including when API requests are made to news sources, users creating an account and users receives news recommendations, etc. I have divided this section into two different phases which make up the entire recommendation process — Aggregation phase and Recommendation phase

Fig. 20: Content aggregation processes
Fig. 21: Content recommendation processes

It is apparent from the figures above that the PCM uses IBM Watson technology, Mercury Web Parser and a combination of two recommendation techniques for providing relevant and serendipitous stories to a user. The system recommends items to $U_a$ based on *topics of interests* and similar users. To make such recommendations possible, $U_a$ interest must be a subset of the IBM Watson Natural Language Understanding taxonomies, i.e., $U_k \subset U_N$ where $U_k$ is user’s interest keywords and $U_N$ is Watson taxonomies. Before stories from *News API* are stored in the database, IBM Natural Language Understanding first analyzes the text into a 5-level taxonomy. In the following section, I will explain the aggregation process in detail.

6.1. The Aggregation process

Data used within the PCM are of two types - (1) aggregated news scraped by Mercury API (2) user-specific preferences information which are both used to make the recommendation. For ‘trying out’ purposes the study uses a few news providers including New York Times, The Guardian, Associated Press and the Washington Post. Fig. 20 shows the flow of aggregation process by first making HTTP GET requests to news providers via *News API*. The system then gets the URL field from the responses (see Fig. 7) and passes it to Mercury Parser API which
returns a JSON Object with relevant content about the stories (see Fig. 19). For each news, the content field (news text) from the Object is passed to Watson NLU to be analyzed and categories retrieved. Fig. 22 shows Watson response for a story with the corresponding categories. The response contains the top three categories for a news text and the respective scores. The PCM saves the text content, title, URL, author, news provider name, categories and category score in MongoDB as a new document. Fig. 25 shows a piece of news with the corresponding categories after Watson analyzes.

![Fig. 22: Response from IBM NLU analysis](image)

When users visit the homepage, they are welcomed with an aggregated list of news stories (as shown in Fig. 24). For users to get personalized contents, they must first be logged in. Fig. 23 shows the login page. For log out users or users without any topics of interest, the PCM recommends non-personalized and only a limited list of stories.
Fig. 23: User’s Login Page

Fig. 24: Aggregated News presentation view
Trump, Graham feud over President's Charlottesville response

The feud between President Donald Trump and Sen. Lindsey Graham over the President's response to racially motivated protests in Charlottesville, Virginia, continued Thursday, with the South Carolina...

Fig. 25: Aggregated News categorized

```javascript
/*
 * IBM NLU analyzes aggregated stories
 */
natural_language_understanding.analyze(parameters, function(err, NLUAnalyzed) {
    if (err){
        console.log('error:', err);
    }
    else{
        /*
         * Analyzed stories are saved to DB
         */
        var newNews = new PostProfile.News({
            title: data.title,
            author: data.author,
            source: response.source,
            content: data.content,
            date_published: data.date_published,
            lead_image_url: data.lead_image_url,
            domain: data.domain,
            url: data.url,
            excerpt: data.excerpt,
            category: NLUAnalyzed.categories[0]
        })
        newNews.save(function(err, post){
            if(err){
                Object.keys(err.errors).forEach(function(key) {
                    var message = err.errors[key].message;
                    console.log('Validation error for \"%s\": %s', key, message);
                });
            } else if(post){
                // post successful
            }
        });
```
6.2. Recommendation process

The recommendation process for an active user $U_A$ starts by the PCM noting the interests of $U_A$ and finding users who have a similar topic(s) of interest as $U_A$. In doing content-based filtering, the system uses $U_A$’s topic(s) of interest to make predictions. Fig. 27 is a user interface allowing users to select from existing list of topics which they find interesting. The PCM compares stories saved in the DB with $U_A$ preferences and then makes recommendations. The result is a list of sorted news stories which $U_A$ would be interested to read.

Users may also provide information about their profession which IBM Watson NLU analyzes to get the category(ies) which is saved in the database alongside the profession to be used for collaborative filtering.

![PERSONALIZE YOUR PREFERENCES](image)

**Fig. 27: UI for personalizing interest**

If the PCM finds matching users for $U_A$, it filters those users who have rated items in the past and finds similarity value between $U_A$ and each user to get nearest neighbors (as shown in Fig. 28)
That is, for each user $U_i$, the similarity value is calculated by comparing each $U_a$’s topics of interest with that of $U_i$. For a smooth implementation of similarity measure, the PCM uses an existing Node npm package called string similarity. It has a findBestMatch method which accepts as parameter - a main string (the string to be matched against – in this case $U_a$’s topics of interest) and a set of target string (an array containing strings to be matched with the main string – in this case, $U_i$’s topics of interest) – see Fig. 28.

An array of objects with ratings is returned with a rating property showing the similarity value between each target strings with the main string (Fig. 29). For testing purposes, PCM uses a threshold value of 0.5 to get nearest neighbors, i.e., users whose similarity value with $U_a$ is greater than 0.5. I believe that the system would provide an optimal result, better recommendation and relevant content when the threshold is higher and when there are more users.
When users provide information about their profession, PCM uses NLU to analyze the text, get possible categories and saves the result alongside users’ profile information. This information acts a major contributing factor when predicting new stories for other users. For a user $U_i$, when rating item $x$, the profession categories are compared with the categories of the news being rated to find the similarity value. High similarity demonstrates that the user’s opinion on such a story should be valued high (see more on section 6.2.1). Fig. 30 shows how the system explicitly accepts users rating.

the door to her potential appointment to his seat.

Trump, meanwhile, plans to hold a rally in Phoenix, Arizona, next week, where he’ll have another chance to wade into the state's turbulent Republican politics.

Next, is predicting items to $U_a$ by using all items that nearest neighbors have rated in the past and by computing a weighted average of ratings provided by the neighbors. In this case, only
items which neighbor rated are considered while others are filtered away. The final prediction $P_{u, i}$ is represented by:

$$P_{u, i} = \frac{\sum_{\text{all similar users, } v} (R_{v, i} * S_{u, v} \pm PSV)}{\sum_{\text{all similar users, } v} S_{u, v}}$$

where $S_{u, v}$ is the similarity value between active user $u$ and neighbor $v$ and $R_{v, i}$ is the rating value of neighbor $v$ on item $i$. $PSV$ is a neighbor’s Profession Strength Value (discussed in section 6.2.1). Also see section 4.3 for more explanation about this prediction measure.

Fig. 31: Collaborative filtering recommendation Interface
6.2.1 **Profession Strength Value (PSV)**

From the above equation, **PSV** is a value representing the relationship between a neighbor’s profession and a rated news story. It is a value of 0 or ± 0.25. 0 for users whose profession does not contribute to overall prediction, +0.25 for users who rate a story high (>2) and whose profession is related or is relevant to the story rated and -0.25 for users who rate a story low (<3) and whose profession is relevant to the story rated. 0.25 is just a random value which would contribute in preventing situations where the final prediction value, \( P_{u,i} \) of an item exceeds 1.0. This value depends on the user rating on an item and it plays a vital role in determining predictions of news stories for active users \( U_a \). If a neighbor's profile indicates that ‘I am a nutritionist’ and then rates a story relating to ‘health’ or ‘fitness’ with a rating value of either 3, 4 or 5, PCM adds 0.25 to the calculated prediction value for that story. On the flip side, if the neighbor gives a rating value of 1 or 2 then 0.25 is subtracted. For users, whose profession shows no similarity with the story rated, a default value of 0 is assigned. The JavaScript code snippet in Fig. 32 and Fig. 33 shows how this process is achieved.

```javascript
neighboursRatedItems.forEach(function(element) {
    var clonedObj = Object.create(finalObj)
    if(typeof element.similarity !== 'undefined'){
        Esum = element.rating * element.similarity
        // ...normalize the sum of the product of ratings and similarities
        var similaritiesRatingsSum = ((Esum - 0) / (5-0)).toFixed(2);
        var predictions = (similaritiesRatingsSum / summedSimilarity).toFixed(2)
        clonedObj.predictionValue = predictions
        clonedObj.story = element
        // similarityWithContent
        userBasedF.push(clonedObj)
    }
}, this);
```

**Fig. 32: Calculating prediction value for each rated Item**
Fig. 3.3: Final predictions using PSV

6.2.2. User walkthrough

I have hosted a prototype version of this system using Heroku cloud platform\(^{20}\) at https://news-recomd.herokuapp.com/. Using the system from a user’s perspective is straightforward. Users visits the system and are welcomed with a list of unfiltered stories, they may decide to create a more personalized environment for themselves through their personal profile but they must be logged in to do so. The account creation page requires the user to specify an email, password & username. They are redirected to a login page to enter their account authentication details which lands them on the homepage.

To personalize their needs a user clicks on ‘choose your preference’ link, a modal pop open where they select from a list of topics which they find interesting and would like to be recommended on. They can optionally enter some information about their profession. The system refreshes with stories matching users’ preferences. A user clicks on a story thumbnail and is directed to a page to read the full story which they can also give a rating value from 1-5 depending on how relevant they think the story is to them.

---

\(^{20}\) Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud.
6.2.3 **Evaluation of the recommender system**

Evaluation of recommender system has been an area of active research. Evaluation is useful in informing the selection of recommender algorithms, how they are tuned and how the system is designed but most importantly it helps in enforcing discipline about recommendations and user experience. Most research on recommender system focuses on evaluation using accuracy metrics (e.g. Mean Absolute Error, MAE, Root Mean Squared Error, RMSE, Precision, Receiver Operating Characteristics, ROC & Recall) but (Mcnee et al., 2006), (Ziegler, McNee, Konstan, & Lausen, 2005) & (Pu, Chen, & Hu, 2011) argues that such narrow focus had been misguided, but has even been detrimental to the field. Moreover, accuracy metrics do not judge the content of the recommended items but only looks at the accuracy of individual prediction.

In their paper, (Mcnee et al., 2006) review three other aspects that should be considered when evaluating recommender systems, one of which I used as a metric for the evaluation of this system. For (Mcnee et al., 2006) the similarity of recommendation lists, recommendation serendipity, and the importance of user expectations in a recommender are paramount if recommendation is to be done from a user-centric perspective. In a similar study by (Ge, Delgado-Battenfeld, & Jannach, 2010), they agree that “accurate recommendations may sometimes not be the most useful ones to the users, and that evaluation metrics should take into account other factors which impact recommendation quality such as serendipity and be applied to recommendation lists and not on individual items”. They went further to focus on metrics which takes into consideration ‘coverage’ and ‘serendipity’. Serendipity metrics measure the degree of “unexpectedness” of recommendations made.

In the end, users are the main reason why recommendation system exist and although user satisfaction can be evaluated using different criteria, I believe that it depends on whether a recommender system provide unexpected items that are relevant to users’ needs or not. Therefore, to
consider user satisfaction, evaluation metrics must also include serendipity and novelty. PCM system aims to provide novel and serendipitous item, I focus on serendipity and consider a metric for measuring it.

I used the same approach as (Murakami, Mori, & Orihara, 2007), which is based on the idea that “unexpectedness is low for easy-to-predict items and high for difficult-to-predict items” and the metric proposed by (Ge et al., 2010) which “determines the ratio of serendipitous recommendations in the recommendation list and also takes their usefulness into account. I create two list which include (1) Items predicted by primitive prediction model (PPM) which returns high ratability i.e. items that have been rated high by users, denoted by RS and (2) items using our prediction model which returns relevant & unexpected items denoted by PM. I assume that when an item in PM cannot be found in RS it is considered unexpected and relevant. I calculate the unexpected set as follows:

\[ UNEXP = RS - PM \]

\( UNEXP \) represents a set containing only unexpected items. i.e. if RS and PM are two sets of items, \( RS - PM \) (UNEXP) represents items in PM not found in RS. But unexpectedness does not also mean relevance. If each item in \( RS - PM \) is represented as \( RSi \), the relevance of each item is then \( r \). \( r \) represents the aggregated \( psv \) (see section 6.2.1) of an item (i.e. sum of each user’s \( psv \) for the item). The serendipity value of PCM’s recommendation of items set for a user is:

\[ SRDP = \frac{\sum_{i=1}^{N} r^i}{N} \]

where N is total value in the unexpected items list.
While evaluating, I had only a few users and over 400 news items in the database. I compared the approach used in this study to Amazon’s item-to-item collaborative filtering. Using item-based approach, based on the rating action of a user, items which are similar (using cosine similarity as in section 4.2) in content to those the user has previously rated highly are recommended. I selected 0.6 as a threshold when calculating the similarity between news items. To validate my approach using the above metric, I calculated the serendipity value at any point in time where $PM$ represent recommendations generated by Amazon’s item-to-item approach and $RS$ represents recommendations generated by using our hybrid approach. For some users, the $UNEXP$ value was over 30% since $RS$ contained items not present in $PM$. As noted earlier, this does not mean that those items are relevant to a user. Therefore, for each news story $i$ in the unexpected recommendations, $r$ represents the relevance of the item to a user. Results take into consideration the aspects which we deal with in this study (both Serendipity and relevance).
Conclusion

The web has evolved as the most powerful news delivery platform. But with the amassing content, readers expect that they are only provided with stories tailored to their interest. Over the years, news aggregators have collated news from various news providers which are presented to readers. While different personalization techniques have been employed to provide users with the best news, it comes at the expense of trading off novelty and relevance for simple personalization. That is, while most aggregating system may provide news stories which meet with readers' interest, many do not provide content which are considered relevant and serendipitous. One reason for this is that the recommendation technique used usually focus on using data from users' previous activities such as rating, searching for and viewing item.

At the beginning of this study, I indicated that one objective was to solve the ‘filter bubble’ problem, a problem resulting from what I called ‘over-personalization’ by modern personalization approaches, I do so by using a combination of two recommendation techniques (content-based and collaborative filtering) without the use of user’s previous activities as in the case of traditional approaches. Instead, similarity is based on the profiles closeness, enhancing relevance and final recommendation is based on neighbors previous rating in addition to information about their profession. By combining the two approaches what I mean is – I have created a platform where both methods are employed differently in making recommendations for users. A main section which utilizes content based recommendations and a sidebar which recommends items based on neighbors’ interest. Both approaches utilize IBM Natural Language Understanding for better personalization by semantically extracting information from news stories which are useful for the system’s recommendation process. Although this study, utilizes Watson for just getting taxonomies from a news content, there are other areas in which it could
be used alongside optimal predictive analysis to enhance better personalization such as getting historical information about concepts in a news story as the reader reads or providing readers with information regarding the emotions expressed by the news author or even information on sentiments expressed (whether positive or negative). There were a few times in this study, where Watson analysis of a news story didn't produce the correct category for that story. In that case, the solution to improving the accuracy of Watson Natural Language Understanding is by customizing text analysis with custom models using **Watson Knowledge Studio**.

While the approach used in this study solves the problem of data sparsity by using users’ profiles, it doesn't yet address the issue of scalability which may arise because of many users. That is, when the volume of data set is huge, computational load would be very high. Therefore, nearest-neighbor algorithms usually require that they scale with both the number of users and items. Some solutions to this problem have been proposed in the past by researchers, for example (Zhi-Dan Zhao & Ming-Sheng Shang, 2010) suggested the use of cloud computing platforms such as Hadoop (which implements the MapReduce framework), Dryad of Microsoft, or Amazon’s Dynamo. According to them, by using a platform such as Hadoop, the program can execute in parallel while using MapReduce to break big problems into smaller ones thus improving speed. (Siavash & Ali, 2011) also proposed a “method which features hybridizing density-based clustering and user-based collaborative filtering”. In their approach, after preprocessing (getting similarity measures based on demographic attributes such as age and gender of users), users are clustered based on these attributes and each cluster is used as an input to user-based collaborative filtering. (Lee, Choi, & Woo, 2002) proposed a system which combines collaborative filtering(CF) with Self-Organizing Map (SOM) neural network. Therefore, in future work, I intend to implement neural network into PCM to enhance recommendations which would allow for diverse kinds of information to be integrated, application of users’ demographic information (age, sex etc.), various data types considered and information inefficiency handles
properly. This is would mean a more complex system that meets users’ needs even more accurately. Overall, this study addresses the initial questions posed at the beginning. In chapter 5, I elaborated in details the various recommendation processes and the reasons behind my choice of algorithm. In chapter 2, I discussed why relevant and serendipitous contents is necessary. By using a hybrid system of user-based collaborative filtering and content based filtering, novelty needs for users is meet, recommendation is better by using a demographic attribute of users (profession) and the relevancy level of recommended content is optimal.

Conclusively, the news industry is undergoing a wide-ranging transformation with the hype around social media and online instant video recording of events. This means that anyone can create their own news stories and push it to the internet. In such scenario, personalization and relevance are important factors to be considered if news content is to be delivered right. More so, In the future, I think it is news companies that leverage the capabilities of the semantic web for personalizing content by linking all entities representing places, things or people for machine readability that would get the most audience.
Acknowledgement

Foremost, my sincere gratitude goes to my supervisors Dr. Oleksiy Khriyenko & Vagan Terziyan for their patience, motivation, support and immense knowledge. Their guidance helped me throughout the writing of this thesis. I couldn’t have imagined better advisers and mentors for this study.

Besides my supervisors, I would like to show my appreciation to the Faculty of Information Technology where I studied the Web Intelligence and Service Engineering program. The study structure is designed in a way that facilitates knowledge in developing intelligence software applications for web services. I have no regret in choosing this course program.

My sincere thanks also go to Truth Lumor, for the thought provoking discussions we had about this subject area. Also, I thank my teacher Fergal Carolan for helping me proofread and better structure this thesis.

Lastly, I thank my parents Henry Okojie and Blessing Okojie for their sponsorship and immense support both in my studies and career.

Charles Osemudiane Okojie
Jyväskylä,
February 22, 2018
Bibliography


72


mobile-apps


Xiaoyuan, S., & Taghi, M. K. (2009). A Survey of Collaborative Filtering Techniques. Retrieved from http://delivery.acm.org/10.1145/1730000/1722966/p4-su.pdf?ip=130.234.244.80&id=1722966&acc=PUBLIC&key=74A0E95D84AAE420.06A1DC718DC957B2.4D4702B0C3E38B3.5.4D4702B0C3E38B35&&_acm__=1519154909dc7f11d8e2b72e126159f36640fc8d57

Zhi-Dan Zhao, & Ming-Sheng Shang. (2010). User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. In *2010 Third International Conference on Knowledge Discovery and Data Mining* (pp. 478–481). IEEE. https://doi.org/10.1109/WKDD.2010.54
