NTNU
Norwegian University of
Science and Technology

# Semantically Enriched Recommender Systems for Smart Closets

**Anders Kolstad**

Master of Science in Informatics
Submission date: June 2017
Supervisor: Jon Atle Gulla, IDI
Co-supervisor: Özlem Özgöbek, IDI
Simon Litlehamar, Accenture

Norwegian University of Science and Technology
Department of Computer Science

# Abstract

With increasingly low prices on clothes and a tremendous interest in fashion, results in that people's wardrobes keep piling up. This causes people to struggle with daily selection of an outfit that includes clothing items that are matching and that suits the day's weather. Moreover, the environmental sustainability could benefit from people recycling their textile waste. This thesis investigates how exploiting new technologies, such as Big Data, recommender systems, semantic web, and Internet of Things can guide people in organizing their wardrobes more efficiently.

The thesis proposes the architecture of a system consisting of a smart closet where the usage history of the user's clothing items can be tracked using RFID technology. Through a mobile application, the user can view wardrobe inventory, and receive daily outfit recommendations and recommendations for clothing items to recycle.

To generate the daily outfit recommendations, the thesis proposes an approach based on collaborative filtering enabled with a machine learning algorithm—carefully selected through an experiment comparing multiple models on a real-world dataset. In addition to daily outfit recommendations, the approach facilitates the possibility of targeted advertisement by clothing retailers. Moreover, the thesis proposes a traditional content-based approach for recycling recommendations. The content-based approach utilizes semantic web technology and results from an evaluation shows that it outperforms a baseline approach without semantic web technology.

As a whole, the thesis includes the state of the art for recommender systems utilizing semantic web technology, with especially focus on the domain of fashion recommendation from people's physical wardrobes.

# Sammendrag


Med lave priser på klær og en enorm interesse for mote, har gjort at folks klesskap fyller seg opp. Dette fører til at folk sliter med å velge daglige antrekk av klesplagg som passer sammen og som passer til dagens vær. Videre kan resirkulering av tekstiler føre til et mer bærekraftig miljø. Denne oppgaven utforsker hvordan man kan anvende ny teknologi som big data, anbefalingssystemer, semantisk web og tingenes internett, for å hjelpe folk med å organisere klesskapene sine mer effektivt.

Denne oppgaven foreslår en arkitektur for et system bestående av et smartskap hvor brukshistorikken til brukernes klesplagg blir sporet ved hjelp av RFID-teknologi. Gjennom en mobilapplikasjon vil brukeren kunne se beholdningen i skapet, samt motta anbefalinger til daglige antrekk og anbefalinger for hvilke klær som kan resirkuleres.

For å generere anbefalinger av daglige antrekk foreslår oppgaven en framgangsmåte basert på samhandlingsfiltrering som anvender en maskinlæringsalgoritme valgt ved å sammenligne modeller utført i eksperimenter på et datasett. I tillegg til daglige antrekksanbefalinger legger framgangsmåten til rette for å inkludere målrettet annonsering fra kleskjeder. Videre foreslår oppgaven en tradisjonell innholdsbasert framgangsmåte for å anbefale plagg til resirkulering. Framgangsmåten anvender semantisk webteknologi og utkonkurrerer en grunnleggende framgangsmåte som ikke anvender semantisk webteknologi.

I sin helhet så gir oppgaven en oversikt over dagens løsninger og den gjeldene kunnskapen for anbefalingssystemer som anvender semantisk webteknologi—spesielt i domenet omkring anbefalinger fra folks klesskap.

# Preface

A thesis submitted to the Department of Computer Science, in partial fulfillment of the requirements for the degree of Master of Science in Informatics at the Norwegian University of Science and Technology, under the supervision of Prof. Dr. Jon Atle Gulla and Dr. Özlem Özgöbek. The thesis has been composed in a research cooperation with Accenture, under the supervision of Simon Litlehamar.

Submitted by Anders Kolstad
Trondheim, June 6, 2017

# Acknowledgements

First and foremost, I would like to thank my supervisors at the department. Dr. Özlem Özgöbek for her valuable advice and constructive criticism, and for her patience and availability when the deadlines were tight. Prof. Dr. Jon Atle Gulla for taking on this project idea, his continuous encouragement and his ingenious ideas for improving the system. I would especially like to thank Prof. Dr. Jon Atle Gulla for giving me the opportunity to attend and present part of my work at an international conference during the work on this thesis.

Furthermore, I would like to thank my supervisor at Accenture, Simon Litlehamar, for providing me with the access to the system's source code and his valuable advice and feedback throughout my work. I would also like to thank him for organizing my presentations at Accenture's offices. Thereby receiving valuable feedback from some incredible smart and talented consultants at Accenture.

Lastly, I wish to thank everyone at Accenture who have given feedback and contributed to this project, and especially the team who first developed the idea and the first prototype of the system. Without them, this thesis would not be possible.

# Contents

# List of Figures

# List of Tables

# Part I

# Research Overview

# Chapter 1

# Introduction

This chapter starts by motivating the work of this thesis (Section 1.1) and describing the research context of the project (Section 1.2). It states the thesis' problem formulation (Section 1.3) and defines the goals and research questions for the project (Section 1.4). Moreover, it summarizes the contributions (Section 1.5) of this thesis and describes the three papers (Section 1.6) that comprise the thesis. An outline for the rest of the thesis concludes the chapter (Section 1.7).

## 1.1 Background and Motivation

With the advent of Web 2.0 platforms, such as social media websites, user-generated content has contributed to an exponential growth of data available on the Internet [2]. By the latest wave in the era of computing, data collection and gathering by so-called Internet of Things (IoT) devices [3], the growth of data on the Internet continue to increase [4]. This unprecedented scale of data, called Big Data [5], has made it possible to build predictive analytic tools who now drives many aspects of mobile services, retail, and our everyday lives. Together with IoT, recommendation technology, and large machine-understandable knowledge bases, there is a potential to build applications that simplify cumbersome tasks and help people in meeting their needs and goals in more efficient ways.

The increasingly low prices on clothes and a tremendous interest in fashion, have resulted in that the typical Norwegian has 359 clothing items in their wardrobe, where 20% of the clothes are rarely used [6]. This contributes to complicating the task of daily outfit selection of an outfit that includes clothing items that are matching and that suits the day's weather—which is a socially and culturally important task that requires time, effort, and thought [7]. Moreover, textile recycling of rarely used items could have huge benefits for the environmental sustainability [8]. Fashion retailers could also benefit from data on usage history, by generating targeted advertisement, decreasing the number of unnecessary purchases and increasing the number of satisfied customers.

This motivation demands for a smart closet leveraging IoT technology such as radio-frequency identification (RFID) [9], tracking clothing items' usage history. Furthermore, the closet should be explorable on a mobile device, displaying content generated by recommender technology.

In the summer of 2016, the author of this thesis participated in a summer internship at Accenture. During the internship, the author and a team of other students built a prototype for a smart closet called Connected Closet. The work done by the students provided an apt skeleton for a smart closet with basic functionality, but lacked algorithms for providing users with outfit recommendations and recycling recommendations as motivated above. The project gained a lot of attention and was featured in one of the largest online IT newspapers in Norway (Figure 1.1) [1].


Figure 1.1: **News article by digi.no about the summer internship.** *The internship was featured in a news article containing an interview with the thesis author and the project manager [1].*

## 1.2 Research Context

The source of this project originates from Accenture's internal global idea competition called Accelerating Innovation IoT Challenge, where the idea for the system originally came from. This idea was then extended to an internship at Accenture where a prototype was developed. Accenture is a global company employing over 401,000 people, providing management consulting, technology services and outsourcing services to clients in more than 120 countries [10].

The core motivator for extending Accenture's project into a research project at the Norwegian University of Science and Technology (NTNU) is the NTNU Smartmedia program led by Prof. Dr. Jon Atle Gulla [11]. The program's research focus is on mobile context-aware recommender systems with an aim to aid the media industry and to present a context-aware news experiences that are mobile and based on deep understanding of textual content. Central technologies for the project include; Big Data architectures, information retrieval and recommendation, semantics, text analysis, and mobile platforms.

The work of this thesis is conducted as part of a research cooperation between the NTNU Smartmedia program and Accenture Norway, with the aim to develop a second prototype of the smart closet and enable it with the technology that is researched at Smartmedia.

## 1.3 Problem Formulation

The main goal of this thesis is to develop semantic web and recommender technology into the prototype initially developed at Accenture. To evaluate the developed recommendation approaches, data collection of a suitable dataset is required. The problem that this thesis addresses can be formulated as follows:

> *A large number of people struggle to organize their wardrobes in efficient ways. This contributes to a weaker economic and environmental sustainability. This thesis investigates how contextual signals obtained from a smart closet, can be leveraged in order to generate accurate recommendations that can guide smart closet users to achieve more organized wardrobes.*

## 1.4 Goals and Research Questions

The aim of this thesis is ultimately to serve as a research contribution. Moreover, it should increase Accenture's business value and contribute to innovation in the company. Therefore, a set of goals and research questions have been defined.

The following project goals are mostly concerned with increasing Accenture's business value:

**G1** *The 'New IT' strategy*

This project should demonstrate how to realize Accenture's IT strategy called the 'New IT' strategy [12]. This thesis focuses especially on demonstrating how to develop 'liquid applications' by using the micro-service architecture [13] enabling continuous delivery using DevOps [14]. Moreover, this should serve as a demonstration of how to build 'connected applications' enabled with IoT [15] devices—who uses lightweight communication protocols—and how to handle their generation of large volumes of data. Finally, it should show how to exploit large volumes of data to build 'intelligent applications' that maximize business value with integrated analytics.

**G2** *IoT demonstrator*

A second goal of this thesis is to build a fully functional IoT demonstrator in order to showcase and demonstrate the validity of the system worked out in this thesis. The demonstrator should be easy to transport and it needs to reflect the system's possible applications and significance in a full-scale production mode. Furthermore, it should work as a conversation starter when meeting with potential clients or business and research partners.

**G3** *Show off at conferences*

With an IoT demonstrator, it is a goal to bring the demonstrator to conferences and other events to showcase the system and get a lot of feedback on the research.

**G4** *Publicity*

Lastly, a goal is to promote the project and the system's idea, primarily in Accenture's and the university's community, getting as much publicity and attention as possible. Moreover, as a research project, there is an aim to publish the work in international conferences or journals.

Next are the research questions, which summarizes and defines the research problem that this thesis addresses:

**RQ1** *What are the core components and the functionality of a smart closet and how can this be supported by recommendation technology?*

The first research question is about describing the architecture and design of a system that will solve the problem formulated earlier. The research question requires development of a set of components that will provide some functionality to the user that will support the user in receiving recommendations as motivated earlier.

**RQ2** *How can we automatically extract datasets that help us build and evaluate content-based recommendation and collaborative filtering for smart closets?*

After the core components are developed, a dataset generated using functionality that mirrors the same functionality found in RQ1, is required to assess the quality of the system. The data should help in constructing the steps for the novel content-based and collaborative filtering approaches for the system, which can be evaluated through experiments.

**RQ3** *To what extent (accuracy) can a recommender system help users choose clothes from a smart closet?*

The answer to this question should support a selection of recommendation models that generates the most useful recommendations to the users in terms of some accuracy metrics.

## 1.5 Contributions

The main contributions of this thesis may be summarized as follows:

**C1** *The architecture and design of a smart closet.*

**C2** *A novel collaborative filtering approach for daily outfit recommendations.*

**C3** *A novel semantic content-based approach for garment recycling recommendations.*

**C4** *Thorough evaluations of the approaches using a real-world dataset.*

## 1.6 Papers

This thesis is submitted as a paper collection of three papers. The first one (Paper I) is published in the Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017). The second (Paper II) has been submitted to the 11th ACM Conference on Recommender Systems (RecSys 2017) and awaits author's notification. While the third (Paper III) is submitted to the 13th International Conference on Semantic Systems (SEMANTiCS 2017), also awaiting author's notification. Paper I and Paper II are classified as 'regular/long papers', meaning they report on substantial contributions of lasting value that is completed or validated. Paper III is submitted as a 'short paper', discussing an exciting work-in-progress with preliminary evaluations.

Figure 1.2 illustrates how the papers are tied together, and how they comprise this thesis. Paper I sets the foundation for the whole system that this thesis' research focus is about. It gives an extensive description of the system's architecture and design, and propose two recommendation approaches in the domain of fashion recommendation. Paper II is then an extension to Paper I, addressing Paper I's proposed collaborative filtering method for daily outfit recommendations. Paper III is also an extension to Paper I, but is not directly related to Paper II. It takes ideas from Paper I's proposal of a recycling recommendation approach and improves it with semantic web technologies. Paper II and Paper III both reports on experiments on a collected dataset.

It should be noted that the recommendation approaches proposed in Paper I are simplified descriptions, and to fully understand how the approaches were implemented in the system, one should read Paper II and Paper III.

In this thesis, all the papers are included in the same format as published in the conference proceedings or in the format required for the camera-ready submissions to the conferences. This is done to preserve the papers' structure and semantics as they were intended to be read. Moreover, since the papers addresses the same system, but are written so that each can be read individually, some content overlap throughout the thesis, is to be expected.

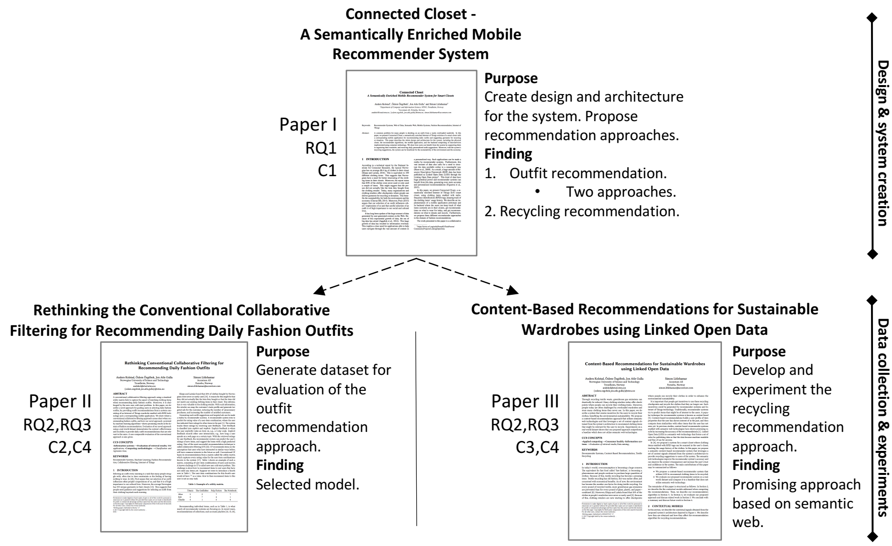



Figure 1.2: **Illustration of how the three papers ties the thesis together.** *Each paper is stated with the paper's purpose and the finding of the paper. Moreover, each paper is labeled with the research questions (RQ) and contributions (C) they mainly addresses.*

## 1.7 Thesis Outline

The rest of this thesis is structured as follows:

- Chapter 2 provides an overview of the theoretical background and defines the terminology used throughout the thesis.
- Chapter 3 provides a short survey of some related work.
- Chapter 4 presents Paper I titled "Connected Closet - A Semantically Enriched Mobile Recommender System for Smart Closets", which gives a full description of the system's architecture and propose some recommendation approaches for the system.
- Chapter 5 presents Paper II titled "Rethinking Conventional Collaborative Filtering for Recommending Daily Fashion Outfits", which describes a collaborative filtering approach implemented in the system.
- Chapter 6 presents Paper III titled "Content-Based Recommendations for Sustainable Wardrobes using Linked Open Data", which describes a semantic content-based approach implemented in the system.
- Chapter 7 discusses some insight obtained throughout the work on this thesis.
- Chapter 8 concludes the thesis by giving some concluding remarks and by discussing future work.

# Chapter 2

# Background

The motive of this chapter is to give a brief introduction to the theoretical background for the system presented in this thesis. Since all of the papers cover some background theory, this chapter is meant to complement the theory given in the papers.

## 2.1 Internet of Things

The term Internet of Things (IoT) was first used in a presentation by Kevin Ashton back in 1999 [3]. He presented an idea of using radio-frequency identification (RFID) in the supply chain of a large consumer goods corporation and linking it to the Internet. The main idea is still the same today—tagging 'things'[1] with identifiers and linking them to the Internet. In recent years, applications such as healthcare, transportation, utilities, and home equipment, have been included in this definition of 'things' [4].

IoT meets the need for data-on-demand by intuitive interactions with ubiquitous computing devices and thus overcoming the gap between 'things' in the physical world and their representation on the Internet [4].

### 2.1.1 Identifiers

For internet-connected devices to identify 'things', numerous identification systems can be used. The most common ones for IoT devices are RFID, Quick Response (QR) codes, and barcodes. The system described in this thesis utilizes RFID which was first proposed in 1948 [16]. Some very early adopters

[1] 'things' refers to everyday objects.

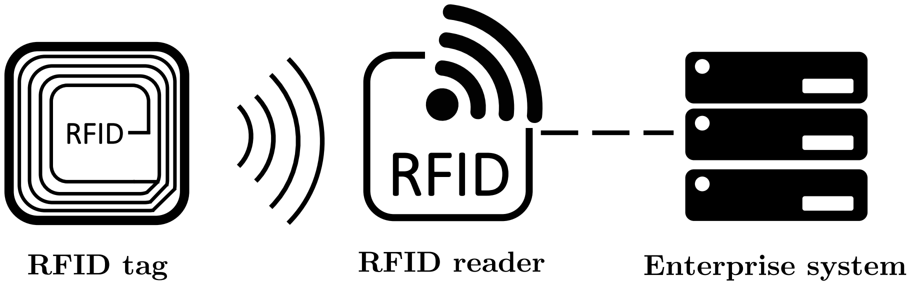



Figure 2.1: **RFID system.** *The architecture of a typical RFID system.*

implemented RFID during the 70's, but it was not until the 90's that RFID was widely deployed and is now part of everyday life [9].

As depicted in Figure 2.1 typical RFID system comprise of an RFID tag and an RFID reader connected to an enterprise system.

### 2.1.2 Lightweight Messaging Protocol

The internet-connected devices that handle the data about the 'things' are sometimes located at remote locations with limited bandwidth capabilities that requires a small code footprint. Therefore, these devices demand a lightweight messaging protocol. The MQ Telemetry Transport (MQTT) protocol was designed to meet the requirements for such devices [17]. MQTT differs from Hypertext Transfer Protocol (HTTP) by using a publish/subscribe style and its much smaller message size, giving a more reliable delivery over fragile networks.

### 2.1.3 Fog Computing

IoT devices require mobility support, location-awareness, and geo-distribution. The Fog Computing [18] platform meets these needs by providing a computing, storage, and networking layer between the devices and the data centers. For example, through the orchestration layer of Fog, smaller real-time analytics tasks can be computed before larger data is sent to the data centers. This reduces latency and avoids potentially unnecessary requests to the data centers.

## 2.2 Recommender Systems

The objective of a recommender system is to support the users in making choices by giving the users some form of recommendation of an item [19]. The motivation

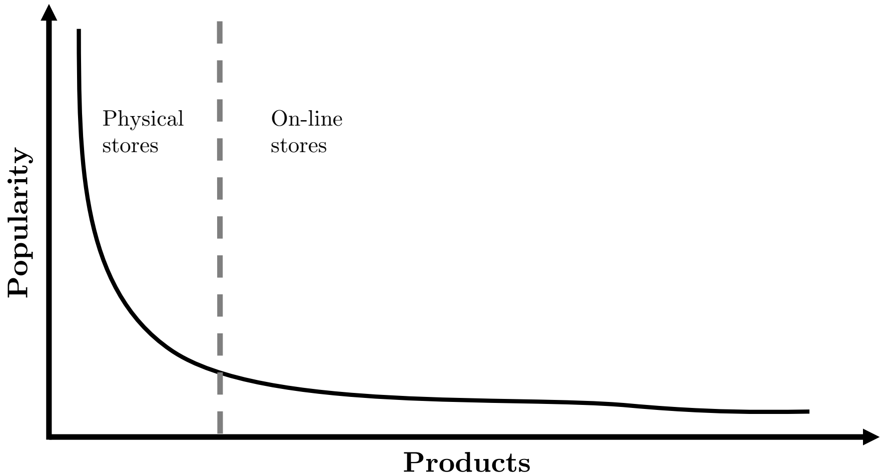



Figure 2.2: **The long tail phenomenon.** *A distinction between physical and on-line stores.*

for this comes from the long tail phenomenon (Figure 2.2), which is a distinction between the physical and on-line worlds [20]. Physical stores have limited space for their products, and can only show the customer a small portion of the products that are available in a certain domain. As a result, the physical store will only provide the most popular products because this will sell the most. In on-line stores, you can offer every product that is available in a given domain. This forces the on-line stores to recommend products to the customer that the customer is most likely to buy. This will help the customers navigate through the numerous available products, selecting the product that best fit the customers' needs, resulting in higher customer satisfaction.

Formally, the problem of recommendations can be seen as mapping the set of users $U$ and the set of items $I$ to a rating [19]:

$$f_R : U \times I \to rating, \tag{2.1}$$

and then recommending new items of high rating to the users.

There are multiple models that tackle this problem. This thesis addresses the two most popular. The first one, called *content-based*, use the attribute information about the users and items such as textual descriptions. The second, *collaborative filtering*, works with user-item interactions, such as previous ratings or buying behavior.

### 2.2.1 Content-Based

Content-based recommender systems try to recommend items that are similar to the ones the user have liked in the past [19]. Figure 2.3 shows an example of a plan of action for a content-based recommender system. Here, the user has liked two items in the past: a red balloon and a yellow star. The system will then build a user profile by learning the attributes of the items that the user have liked. Items that not yet have been seen by the user are then matched to the user profile and the items that are most similar to the user profile is recommended. Of the lower left items in Figure 2.3, the block is most likely to be recommended because it shares two attributes with the user's user profile. The user profiles and item profiles are often represented as vectors and the matching between them is determined with similarity measures as described in Section 2.4.4.

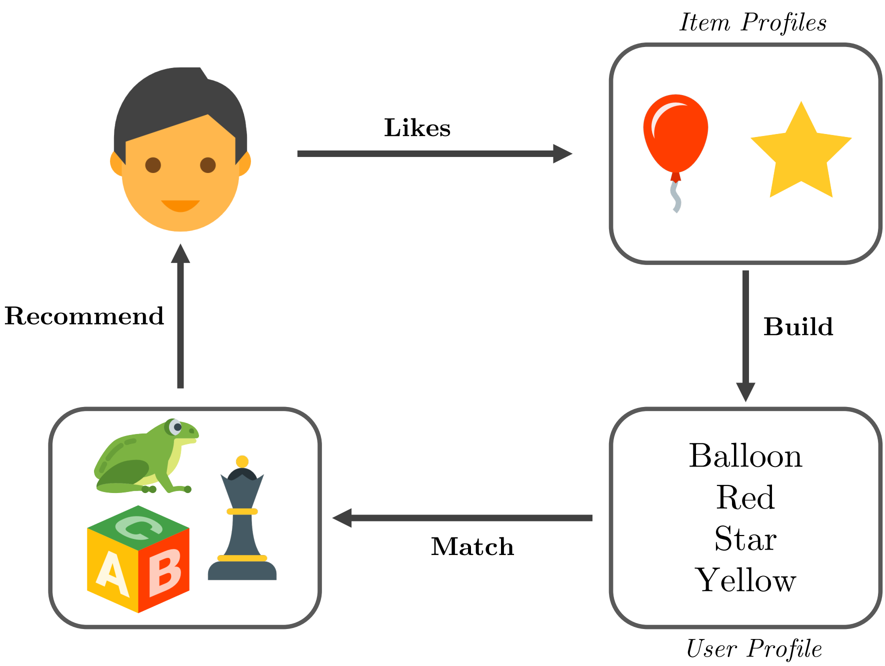



Figure 2.3: **Content-based recommender system.** *An example of a plan of action for a content-based recommender system.*

### 2.2.2 Collaborative Filtering

As opposed to content-based, collaborative filtering does not use item attributes to compute recommendations. Instead, it uses the correlation in behavior patterns across users [19]. The behavior patterns can be tracked by looking at how users rate a set of items. Consider the users in Figure 2.4, they have all rated two items on a unary rating scale by indicating that they like them. The users' unspecified ratings of items can be imputed when a user has highly correlated observed ratings with other users. For example, in Figure 2.4, the leftmost user and the middle user have both liked the Pokémon, while the leftmost user has also liked cycling. Because the two users share a rating, cycling can then be recommended to the middle user.

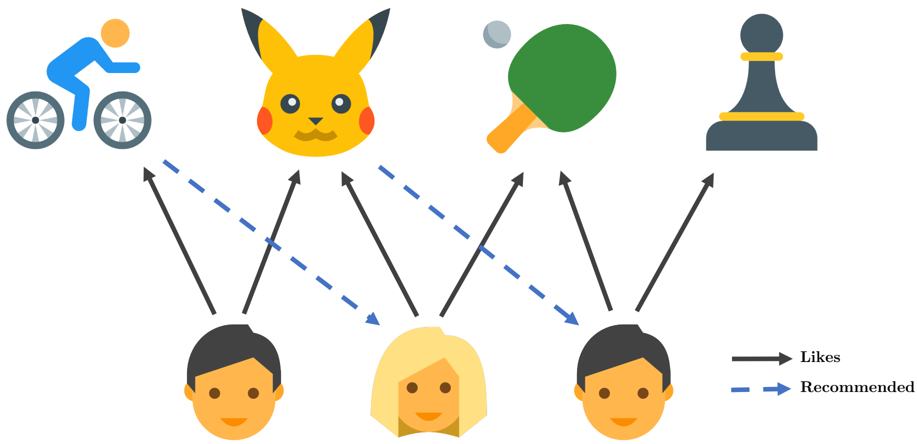



Figure 2.4: **Collaborative filtering.** *An example of correlation between a set of users who have rated a set of items.*

#### Utility matrix

In collaborative filtering, the item ratings are often captured in a set of user-item combinations. These are collected in a matrix called the *utility matrix* [20]. By expanding the rating scale in Figure 2.4 to a five-star rating scale, a possible outcome could be represented as shown in Table 2.1. Here, the '?' values represent the unspecified ratings and are the values the recommender system will try to predict and possibly recommend to the user.

Table 2.1: **Utility matrix.** *An example of a possible utility matrix using a five-star rating scale.*

| | **Cycling** | **Pokémon** | **Ping pong** | **Chess** |
|---|---|---|---|---|
| **Bob** | 5 | 4 | ? | ? |
| **Alice** | ? | 5 | 4 | 1 |
| **Charlie** | 2 | ? | 5 | 5 |

**Collaborative filtering methods**

To predict the unspecified ratings, collaborative filtering is commonly divided in two methods [19]:

1. *Memory-based*, where the whole utility matrix is used to predict the ratings by comparing the ratings of neighboring users (user-based) or neighboring items (item-based). To find neighbors, similarity measures—as addressed in Section 2.4.4—are applied.

2. *Model-based*, where a model of user ratings is generated that will predict the ratings of new items to a user. The model is often generated using machine learning algorithms (Section 2.5).

### 2.2.3 Feedback

The ratings used by a recommender system are obtained through user feedback which is classified as *explicit* or *implicit* [21]. Explicit feedback is when the system explicitly asks the user to rate an item. The advantage of using explicit feedback is that if the user is willing to explicitly rate items, the system is getting the most accurate information. The disadvantage is that the system is requiring additional effort from the user. On the other hand, implicit feedback requires no additional effort from the user and learn user ratings by watching the users' behavior in the

Table 2.2: **Feedback.** *Some examples of feedback techniques classified as explicit and implicit.*

| **Explicit feedback** | **Implicit feedback** |
|---|---|
| Like button | Browsing history |
| Dislike button | Click behavior |
| 5-star rating scale | Songs listened to |
| Reviews | Movies watched |
| | Time on page |

system. However, correct interpretation of user behavior is hard. Table 2.2 gives some examples of feedback classified as explicit and implicit feedback.

### 2.2.4 Context-Aware Recommender Systems

As discussed earlier in this section, the problem of recommendations can be seen as a mapping of a set of users $U$ and a set of items $I$ to a rating. Context-aware recommender systems utilizes additional information (context) about the users or the items to predict the ratings [19]. An additional set $C$ of the context is now introduced to the system. More formally, the recommendation function can now be written as:

$$f_R : U \times I \times C \rightarrow rating \tag{2.2}$$

Here, the set $C$ can for example be the locations $\{home, work, car\}$. This is because the user might have different preferences for items at different locations, and $C$ is therefore included to the mapping to achieve a more accurate recommendation.

### 2.2.5 Challenges in Recommender Systems

Table 2.1 shows a relatively dense utility matrix. However, in the real world, the utility matrix tends to be very sparse—meaning that a lot of the user-item combinations have '?' as values. This poses a challenge for both memory-based and model-based methods, because computing similarities or predictions tends to get harder with little data. This problem is commonly known as the *sparsity problem* [21].

A special case of the sparsity sparsity problem is called *cold-start* [21]. Say a forth user were introduced in the utility matrix in Table 2.1, without any previous ratings of the items (all ratings would have the value '?'). How to recommend items to this user is not an easy tasks and is known as new user cold-start. Similarly, for a new item that no one has rated, is hard to recommend to anyone and is known as new item cold-start.

### 2.2.6 Evaluation of Recommender Systems

Recommender systems can be evaluated in many ways. In this thesis, evaluation of recommender systems is performed using so called *offline experiments* [22]. Here, evaluation is done on a pre-collected dataset that should simulate user behavior in the system. A small portion of the dataset is hidden from the system while the recommender system tries to predict the hidden values. Using a set of evaluation metrics, the quality of the recommendations can be measured in how relevant and well ranked they are to the users.

### Confusion matrix

The evaluation metrics used in this thesis are calculated from the *confusion matrix*. The confusion matrix (Table 2.3) summarizes the number of correctly or incorrectly recommended items [23].

Table 2.3: **Confusion matrix.** *Summarization of how correct recommendations are distributed.*

| | | **Predicted** | |
|---|---|---|---|
| | | Recommended | Not recommended |
| **Reality** | Correct recommendation | $TP$ | $FN$ |
| | Wrong recommendation | $FP$ | $TN$ |

The following terms are used to describe the confusion matrix:

- True positive ($TP$) is the number of relevant (correct) items that were recommended.
- False negative ($FN$) is the number of relevant items that were not recommended.
- False positive ($FP$) is the number of non-relevant (wrong) items that were recommended.
- True negative ($TN$) is the number of non-relevant items that were not recommended.

### Precision and Recall

By using the numbers from the confusion matrix, the two very popular metrics, Precision and Recall, can be used.

Precision is the fraction of correctly recommended items among all the recommended items, and is defined as follows [22]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.3}$$

Recall is the faction of correctly recommended items among all relevant items, and is defined as follows [22]:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

### Receiver operating characteristic

A theoretically grounded alternative to Precision and Recall is the receiver operating characteristic (ROC) curve [22]. It attempts to measure how well the recommender system successfully distinguish between relevance and noise. More formally, the ROC curve is a graphical display of the tradeoff between the true positive rate (TPR) and the false positive rate (FPR), both defined from the confusion matrix as follows:

$$\text{TPR} = \frac{TP}{TP + FN} \qquad \text{FPR} = \frac{FP}{TN + FP} \tag{2.5}$$

Figure 2.5 shows an example of an ROC curve comparing two different recommendation algorithms. The best algorithms tends to be located in the upper left corner. Moreover, an ROC curve is a great way to compare two different algorithms, where the best—on average—achieves the largest area under the curve, which is often referred to as the AUC [22]. From the figure it is shown that $M_1$ is the best performing algorithm until $M_2$ becomes slightly better for FPR $> 0.6$.

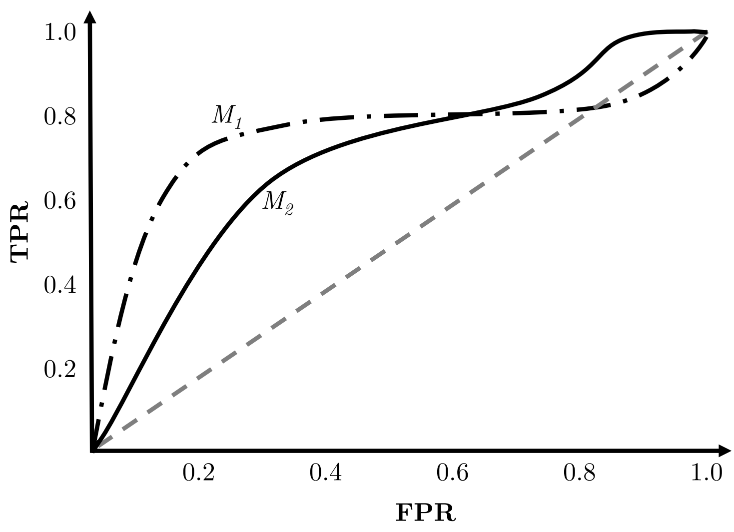



Figure 2.5: **ROC curves.** *ROC curves of two recommender systems $M_1$ and $M_2$. The dotted diagonal line illustrates the performance of a random recommender system.*

#### R-score

To quantify the utility of a recommended list has to the user, the list's top items are given higher importance. The utility of item $j$ to the user $u$ is defined as follows:

$$F(u,j) = \frac{\max\{r_{uj} - C_u, 0\}}{2^{(v_j-1)/\alpha}}, \tag{2.6}$$

where $C_u$ is a neutral rating value to the user, $r_{uj}$ is the user's ground truth rating of the item $j$, $v_j$ is the rank of item $j$, and $\alpha$ is a half-life parameter.

The total R-score for the user $u$ over a ranked list of size $L$, is then the sum of all $F(u,j)$ for all items $j \in I_u$ [19]:

$$\text{R-score}(u) = \sum_{j \in I_u, v_j \leq L} F(u,j) \tag{2.7}$$

The overall R-score for all users in the system can be calculated as follows:

$$\text{R-score} = \sum_{u=1}^{m} \text{R-score}(u) \tag{2.8}$$

To achieve a high R-score, the system needs to be able to place items rated highly by the users in the top of the ranked recommended list.

## 2.3 Semantic Web

The Semantic Web is an initiative trying to structure the available data on the Web so that it becomes machine-understandable [24]. Much of the Web today is uploaded as raw CSV or XML dumps, or in HTML documents. This sacrifices much of the structure and semantics, meaning that data cannot be automatically extracted and connected together to be reused and to be looked at from a different point of view.

### 2.3.1 Ontologies

In order to achieve the Semantic Web vision, ontologies have been introduced. An ontology is an explicit and formal specification of a conceptualization [25]. It describes the domain of discourse by defining the concepts and the relations between them.

### 2.3.2 Resource Description Framework

As mentioned, most of the available data on the Web is published as raw data, such as XML. This type of data does not offer any means of semantics to the data. This means that it is up to each individual application to interpret the data to extract useful information. The Resource Description Framework (RDF) model offers a standardized ontology language to describe data using statements kept in triples following a *Subject – Predicate – Object* format [26]. A triple describes a resource (a thing), a property, and its value. A value can either be another resource or a literal. Resources are identified by a Universal Resource Identifier[2]. Multiple RDF statements makes up a directed graph with labeled edges. In the graph, the Subject is the source of an edge, the Predicate is the edge's label, and the Object is the target. Figure 2.6 shows an example of an RDF graph describing the scientific article [24] (wdt:Q29164671) by Tim Berners-Lee (wdt:Q80) titled "The Semantic Web" published in the journal Scientific American (wdt:Q39379). In the graph, "wd" and "wdt" are namespace prefixes for the vocabulary for a huge online ontology described later in this section.

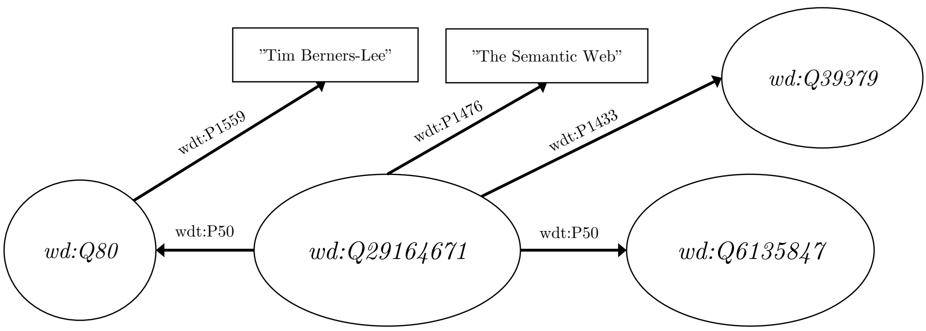



Figure 2.6: **RDF graph.** *Some RDF triples represented as a graph. The circles are resources, rectangles are literal values, and the arrows are the properties.*

### 2.3.3 Linked Open Data

Publishing data using RDF is a step in the right direction for the vision of the Semantic Web. In order to find more related data about a certain topic, *linked data* is introduced. In 2006 Tim Berners-Lee outlined these four principles one must follow in order to achieve linked data [27]:

[2]Usually as a URL.

**Linked data principles**

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF)
4. Include links to other URIs, so that they can discover more things

Data published using these principles and that is published under an open license is known as Linked Open Data (LOD) [28].

### 2.3.4 Wikidata

One of the most essential components for the Semantic Web has become online ontologies that provides a large amount of RDF instances making up a large knowledge base (KB) [28]. Wikidata [29] is an example of such a KB which can be read and edited by any computer agent or human agent. It is a project by the Wikimedia Foundation [30] and stores a vast amount of structured data from Wikimedia's sites including Wikipedia, Wikivoyage, Wikisource, and others.

### 2.3.5 Named Entity Recognition

The extract Wikidata entities in free text, named entity recognition[3] (NER) can be used for recognizing proper nouns (entities) and extract them from the text. One of the first examples of NER was done in 1991 when Rau proposed a system for extracting and recognizing company names [31]. The earliest NER systems was handcrafted linguistic grammar-based techniques, but in recent years machine learning methods have become the trend [32].

## 2.4 Representation and Similarity Measures

To represent the features of a document, many techniques can be used. A document is whatever unit the recommender system is build over. In this thesis, documents are either an outfit or a clothing item description.

This section covers the classical *Bag of Words* approach and the *Bag of Concepts* approach which can utilize the semantics of LOD. To find similar documents and to calculate recommendations, the Vector Space Model and a set of similarity measures are described.

[3]Also known as *entity extraction.*

### 2.4.1 Bag of Words

The most popular method for representing documents is known as the Bag of Words model. It is referred to as a 'bag' because the ordering of the document's words does not count. The documents are represented as a single vector where each dimension corresponds to a single term in the total dictionary for the document collection [23]. Each term is assigned a weight for each document.

#### Term frequency

A simple weighting scheme called *term frequency* (TF) assumes that a term that occurs multiple times in a document is more important that a term that occurs a single time. Hence, the TF for a term in a document is defined as the number of occurrences in the document [23].

### 2.4.2 Bag of Concepts

With the advent of Semantic Web, an alternative model known as Bag of Concepts, poses as an intriguing approach to represent documents using entities from online ontologies. First proposed by Sahlgren and Cöster in [33], Bag of Concepts is a model that instead of looking at each word individually, it looks at the concepts the words forms together. To find the concepts in the text, entity extraction from external sources, such as Wikidata, can be used.

#### Concept frequency

Similarly to TF in Bag of Words, the number of occurrences of a concept in a document can be used as a weighting scheme [34]. This weighting scheme is known as *concept frequency* (CF).

### 2.4.3 Vector Space Model

By representing each document as a vector using either TF or CF as a weighting scheme for each dimension, some distance measures can be imposed to calculate the similarity between documents. To represent the documents in a common vector space is called the Vector Space Model [23].

Figure 2.7 shows an example of two documents represented as vectors in the same vector space of two dimensions. The figure illustrates two different approaches for caluculating the similarity between the two documents, namely the *Euclidean distance* and the *cosine similarity.*

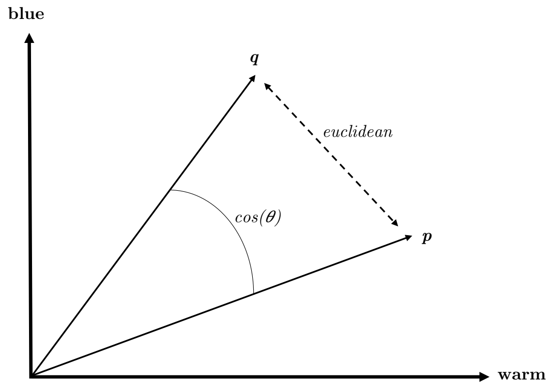



Figure 2.7: **Vector Space Model.** *Example of two vectors in the same vector space. Both the cosine similarity and the Euclidean distance is visualized.*

### 2.4.4 Distance Measures

#### Jaccard similarity

The simplest way to calculate the similarity between two document vectors $\mathbf{q}$ and $\mathbf{p}$ is to convert them into binary vectors and represent them as two sets $A$ and $B$. The Jaccard similarity of the two documents are defined as follows [20]:

$$sim\big(\mathbf{q},\mathbf{p}\big) = JAC\big(A,B\big) = \frac{A \cap B}{A \cup B} \tag{2.9}$$

#### Cosine similarity

The most used similarity measure is the cosine similarity measure. It is defined as the cosine of the angle between the two vectors [20]:

$$sim\big(\mathbf{q},\mathbf{p}\big) = cos(\theta) = \frac{\mathbf{q}\cdot\mathbf{p}}{|\mathbf{q}| \times |\mathbf{p}|} = \frac{\sum_{i=1}^{n} q_i p_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} p_i^2}} \tag{2.10}$$

#### Euclidean distance

As shown in Figure 2.7, the Euclidean distance is the length of the shortest line between two points [20]:

$$sim\big(\mathbf{q},\mathbf{p}\big) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{2.11}$$

## 2.5 Classification

Classification is the task of training a learner (model) by feeding it a set of records—all containing a set of attributes and a predefined class label. The ultimate goal is then to make the model able to predict the class label of novel instance records [35]. In this section, three learning algorithms are presented together with a common challenge in the field of classification.

### 2.5.1 Naïve Bayes

The Bayesian approach to predicting a class label $Y$ of a set of attributes $\mathbf{X}$ of length $d$, is to select the class label that maximizes the numerator of this equation:

$$P(Y \mid \mathbf{X}) = \frac{P(Y)P(X_1, X_2, \ldots, X_d \mid Y)}{P(X_1, X_2, \ldots, X_d)} \tag{2.12}$$

To estimate $P(X_1, X_2, \ldots, X_d \mid Y)$ requires a huge amount of training records. So to simplify, one can assume that the attributes are conditionally independent. This means that $P(X_1, X_2, \ldots, X_d \mid Y)$ can be written as $\prod_{i=1}^{d} P(X_i \mid Y)$, which is used to define the Naïve Bayes learner [36]:

$$P(Y \mid \mathbf{X}) = \frac{P(Y)\prod_{i=1}^{d} P(X_i \mid Y)}{P(X_1, X_2, \ldots, X_d)} \tag{2.13}$$

More formally, Naïve Bayes predicts the class label $v_{NB}$ of a novel record with the attribute set $\mathbf{X}$ as follows:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_{i=1}^{d} P(X_i \mid v_j), \tag{2.14}$$

where $v_j$ is all the possible class labels in the finite set $V$.

Due to its simplicity, Naïve Bayes has become known as "the punching bag of classifiers" and has been successfully used in applications such as spam detection and sentiment analysis.

### 2.5.2 Adaptive Boosting (AdaBoost)

In recent years, machine learning methods called ensemble methods have gained a lot of attention. Ensemble methods works under the intuition that a set of several simpler classifiers can perform better than one single classifier [37]. A simple classifier, often referred to as a weak learner, is a learner that tends to performs only slightly better that random[4], but when selected carefully using a boosting algorithm, they can be combined to a strong learner.

[4]For example a short decision tree.

The first practical example of a boosting algorithm is known as the AdaBoost algorithm [38] which is still one of the most widely used boosting algorithms. It takes $m$ labeled training records $(x_1, y_1), \ldots, (x_m, y_m)$ as input, where the $x_i$'s are in some domain $D$ and the $y_i$'s are binary class labels $y_i \in \{+1, -1\}$. Each training records is assigned a weight. In a forward stage-wise manner of $t = 1, \ldots, T$ rounds, a new weak learner is selected by trying to correctly classify the records with high weights. If the learner misclassifies a record, the record's weight increases. The final classifier $H(x)$ is then a weighted majority vote of all the weak learners $h_t(x)$ with an assigned weight $\alpha_t$ indicating the learner's importance:

$$H(x) = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big) \tag{2.15}$$

### 2.5.3 Gradient Boosting

Five years after AdaBoost was introduced, another boosting algorithm called Gradient Boosting was introduced [39]. It follows the same fundamental idea as AdaBoost, but instead of up-weighting records that are misclassified, Gradient Boosting finds difficult records by the gradient of the loss function of the computation in previous iterations.

### 2.5.4 The Curse of Dimensionality

Coined by Bellman in 1961, "the curse of dimensionality" [40] says that the classifier's performance decreases as the dimensionality of the vectors increases (Figure 2.8). To mitigate this performance issue for classifying vectors of high dimensions, several techniques have been proposed to reduce the dimensionality space.

#### Principal component analysis

Principal component analysis (PCA) is a linear dimensionality reduction technique that tries to find the principal components of the data. PCA transforms the data from one vector space to a new lower-dimensional space where the first dimensions retain most of the original variation [41]. In other words, it puts the data in a new coordinate system where the first axis corresponds to the component that explains the greatest variance in the data.

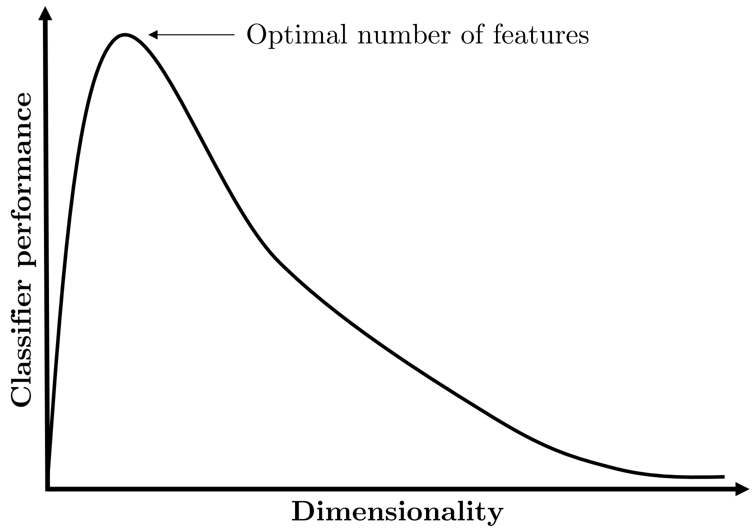



Figure 2.8: **The Curse of Dimensionality.** *For an optimal performance a classifier requires some number of features. As this increases further, the performance of the classifiers weakens.*

## 2.6 Software Design and Tools

To implement the proposed system in this thesis, several tools and architectural patterns were used. How this thesis' components and their platforms are connected is summarized in the architectural view in Appendix A.

### 2.6.1 Microservices

Microservices is a relatively new architectural pattern. The main idea is to build a set of loosely coupled independent small services that collaborates by doing their own processes [42]. The services communicates often using representational state transfer (REST). Microservices enables high modularity and let the developer choose the service's platform they deem most fit for the task of the service.

#### Representational state transfer

REST is a architectural style for distributed systems which comprises of several principles to design distributed services [43]. First, it is about exposing resources on the Web using structured URLs. These resources are transferred from a server to a client using some form of a representation (often JSON or XML) to represent the data. Moreover, the server is not aware of the application state of the clients.

#### Microservices in this thesis

Most of the microservices implemented in this thesis were implemented as REST APIs using Node.js [44]. Node.js is a JavaScript environment that is used to run JavaScript on the server-side. To create the API endpoints, the Node.js web framework Express [45] was used. This was implemented using a superset of the JavaScript syntax called Typescript [46]. Moreover, for the classification and recommendation tasks, a microservice written in Python deemed more expedient. This was written using the Python web framework Flask [47].

### 2.6.2 Libraries and Third Party APIs

For the implemented microservices to achieve the techniques discussed in the theory above, several libraries and third party APIs that provides state-of-the-art techniques, were used.

#### scikit-learn

scikit-learn [48] is a machine learning library for Python. It offers a wide range of state-of-the-art techniques for classification, vectorization, similarity measures, and dimensionality reduction.

#### TextRazor

TextRazor [49] is a natural language processing tool that offers entity extraction, text classification, and more. It it connected to multiple knowledge bases—including Wikidata. By using TextRazor's REST API, Wikidata entities can be extracted from free text.

#### Weather Company Data

The service Weather Company Data from The Weather Company [50] provides extensive weather forecast for many countries all over the world. Using their REST API[5] it it easy to integrate the service in custom applications.

[5] Available at https://twcservice.mybluemix.net/rest-api/.

### 2.6.3 Progressive Web Applications

The term Progressive Web Applications (PWA) was first coined in a blog post by Russel [51]. PWA is an approach for developing mobile applications using modern web capabilities to deliver an "app-like" experience to the user.

In [52], Biørn-Hansen et al. pointed out a lack of academic involvement in PWA research, which denotes a significant knowledge gap. Moreover, they showed how PWAs' much lower size of installation and with a competitive launch time, poses as an intriguing development approach compared to the much used hybrid and interpreted approaches.

#### Angular

In this thesis, a mobile application developed in Google's front-end web app framework Angular [53] is developed with capabilities for a PWA.

### 2.6.4 Database Management Systems

Modern web applications and IoT applications requires database management systems that can handle massive volume of data that are both semi-structured and unstructured. Moreover, they have other requirements than traditional applications when it comes to availability and partition tolerance. NoSQL (Not Only SQL) solves the shortcomings that traditional relational database management systems have for these requirements [54]. In this section, some NoSQL databases that were carefully selected for the system in the thesis' system, are described.

#### Document-oriented database

Document-oriented databases stores data as a collection of documents of self-describing data. The fact that the documents are self-describing means that document-oriented databases require no specific schema for the data. In this thesis, MongoDB [55] is used, which store its documents in a BSON (Binary JSON) format.

#### Graph database

Graph databases represent their data as graphs with nodes and edges between them. Nodes and edges are often labeled, storing data about the associated nodes. A very popular Java-based graph database called Neo4j [56] uses a data model using concepts of nodes and relationships. The relationships can be viewed as the edges between the nodes. Neo4j comes with a rich query language called Cypher which makes it easy to traverse the graph for queries.

#### Time series database

Time series databases lies in the heart of IoT applications. Sensors in IoT applications are generating a vast amount of data entries per second. Traditional databases are not capable of handling the same kind of velocity. Hence, time series databases are designed for this special purpose to facilitate faster data retrieval and high query loads of large chunks of timestamped data. InfluxDB [57] is an example of such a database that comes with an expressive SQL-like query language and high performing query HTTP APIs.

### 2.6.5 Continuous Delivery

In a world where customers requires frequently demonstration of progress of their software and quick fixes of bugs is vital for companies' economy, demands a way to shorten the time to production of software systems. Continuous delivery is a software engineering approach that ensures that software is readily available for production mode at any time [58]. Docker [59] is an OS-level container-based virtualization method. By developing applications in so called Docker containers, enables continuous delivery.

# Chapter 3

# Related Work

This chapter aims to give a brief survey of the related work that is closely akin to the system described in this thesis. Additional related work that is in some parts similar to the system, is given in the corresponding related work sections in the papers.

## 3.1 Physical Wardrobes

Some physical smart closets leveraging IoT technology have been proposed in past work. The system most similar to Connected Closet is a system developed by Goh et al. [60]. They built a system divided into two parts: "the hardware prototype" and "system software". The hardware prototype is similar to the physical closet prototype described in this thesis, with clothing items attached with RFID tags that can be scanned through a reader. Moreover, the system software provides some functionality to the user with the aim to guide the user in selecting clothes. For recommending an outfit they describe a simple algorithm without much detail or any evaluation.

A work that describes a more detailed outfit recommendation approach, is found in a paper by Yu-Chu et al. [61]. However, they only assume that RFID technology is available and the details of the high level architecture is not worked out. The recommendation approach is based on Bayesian networks that considers season, temperature, and occasion when generating outfit recommendations. Moreover, the approach is evaluated in a small user study ($n = 10$) which shows that their proposed approach outperforms a baseline.

The earliest works of similar systems are found in [62] and [63]. Here, the main application seems to be inventory overview. Both these works propose RFID tags embedded in the clothing hangers instead of the clothing items. This is possibly because washable RFID tags were not available ten years ago. It should be mentioned that in [63], Ling et al. does provide clear descriptions on how to build a user profile that could be use to generate outfit recommendations in future work.

As seen above, there are some works that can be related to the system described in this thesis. However, accuracy evaluations on a real-world dataset are non-existent in all of these works. Moreover, none of them addresses the recommendations of clothing items to be recycled. It should also be mentioned that the previous works use pretty old techniques and technologies. While in this thesis, latest technologies, such as microservices, semantic web, state-of-the-art machine learning algorithms, etc., are leveraged.

## 3.2 Virtual Wardrobes

Systems that do not leverage IoT technology requires much more effort from the users to generate recommendations of clothes. However, the techniques of such virtual wardrobes can be compared to the mobile application proposed in this thesis.

In a highly cited paper by Liu et al. [64], they propose a clothing recommendation approach based on the user's inputted occasion. Based on a latent support vector machine, the approach finds visual features and attributes in photos of the user's clothing items, to generate and rank clothing recommendations. This approach is evaluated in an offline accuracy evaluation using a large dataset of clothing photos.

Another work interesting work is a mobile application developed by Dumeljic et al. [65]. They propose an application that recommends daily outfits based on the user's mood. In [65], the application provides functionality to add clothing items with a photo and annotate them with suitable moods. Possibilities to view the clothing items in the user's closet it also available. Moreover, they report from a user study ($n = 10$) that the use of moods fascinated the users and motivated them to use the system.

The last virtual wardrobe to be mentioned is a work by Limaksornkul et al. [66]. They develop a mobile application that aims to help the users manage clothes in their closets. In the mobile application, the user can manage his closet, select clothes, receive suggestions, and view history of previously used items. To generate suggestions, they propose a statistical-based recommendation engine that requires the users to select outfit combinations he likes. Based on these selections, the system will recommend outfits that are similar to the dressing style identified by the outfit combinations.

## 3.3 Other Domains

A natural question that arises when researching recommender systems for smart closets, is to look for similar architectures and approaches in other domains than the fashion domain. For this thesis, the only work found that can be considered as related work is an application proposed by Xie et al. [67]. They propose a smart refrigerator that leverages RFID technology to collect food information, perceive user activity, and locate food.

# Part II

# Papers

# Chapter 4

# Paper I

Anders Kolstad, Özlem Özgöbek, Jon Atle Gulla and Simon Litlehamar. (2017). **Connected Closet - A Semantically Enriched Mobile Recommender System for Smart Closets**. In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, ISBN 978-989-758-246-2, pages 298-305.
https://doi.org/10.5220/0006298002980305

# Connected Closet
## *A Semantically Enriched Mobile Recommender System for Smart Closets*

Anders Kolstad[1], Özlem Özgöbek[1], Jon Atle Gulla[1] and Simon Litlehamar[2]
[1]*Department of Computer and Information Science, NTNU, Trondheim, Norway*
[2]*Accenture AS, Fornebu, Norway*
*andekol@stud.ntnu.no, {ozlem.ozgobek, jon.atle.gulla}@ntnu.no, simon.litlehamar@accenture.com*

Keywords: Recommender Systems, Web of Data, Semantic Web, Mobile Systems, Fashion Recommendation, Internet of Things

Abstract: A common problem for many people is deciding on an outfit from a vastly overloaded wardrobe. In this paper, we present Connected Closet, a semantically enriched Internet of Things solution of a smart closet with a corresponding mobile application for recommending daily outfits and suggesting garments for recycling or donation. This paper describes the whole design and architecture for the system, including the physical closet, the recommender algorithms, the mobile application, and the backend comprising of microservices implemented using container technology. We show how users can benefit from the system by supporting them in organizing their wardrobe, and receiving daily personalized outfit suggestions. Moreover, with the system's recycling suggestions, the system can be beneficial for the sustainability of the environment and the economy.

# 1 INTRODUCTION

According to a technical report by the National Institute for Consumer Research, the typical Norwegian has on average 80.4 kg of clothes in their closet (Klepp and Laitala, 2016). This is equivalent to 359 different clothing items. This suggests that Norwegians have a need for better structuring of the clothing items in their closets. Moreover, the report states that 20% of the clothes were never used or only used a couple of times. This might suggest that the person did not actually like the item they bought from the clothing retailer. Today, many organizations and clothing retailers offer checkpoints where people can deliver garments for recycling or donation. This benefits the sustainability for both the environment and the economy (Chavan RB, 2014). Moreover, Pruit (2015) argues that our selection of an outfit influences others' impressions of us and that careful selection of an outfit is of high importance to our social and cultural lives.

It has long been spoken of the huge amount of data generated by user-generated content on the Web. Because of this exponential growth of data, the era of big data has arised (Jagadish et al., 2014). This huge growth of data has resulted in information overload. This implies a clear need for applications able to help users navigate through the vast amount of content in a personalized way. Such applications can be made a reality by recommender systems. Furthermore, this vast amount of data also calls for a need to structure the data available online in a meaningful way (Bizer et al., 2009). As a result, a huge amount of Resource Description Framework (RDF) data has been published as Linked Open Data (LOD) through the Linking Open Data project[1]. This kind of data have huge potential power and recommender systems can benefit from this data, generating even more accurate and personalized recommendations (Figueroa et al., 2015).

In this paper, we present Connected Closet, a semantically enriched Internet of Things (IoT) smart closet, using clothing items enabled with radio-frequency identification (RFID) tags, keeping track of the clothing items' usage history. We describe an implementation of a mobile application prototype and its backend where the users can keep track of what items currently are in their closets, get recommendations on what to wear for today, and get recommendations on what to donate and recycle. Furthermore, we propose three different recommender approaches in the domain of fashion recommendation.

The work presented in this paper is a collaborative

[1]https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

effort between the Smartmedia program[2] at the Norwegian University of Science and Technology[3] and Accenture Norway[4]. The Smartmedia program focuses on mobile context-aware recommender systems research. The goal of this program is to present a context-aware, personalized news reading experience based on deep understanding of the textual content of news articles. Accenture is a leading management, strategy, consulting, and technology company employing approximately 384,000 people. Using technology, Accenture strives to help businesses create IT applications to deal with rapid changes caused by an increasingly digital economy. Using their *new IT strategy*, they deliver liquid, connected, and intelligent applications to their clients. In this research project, Accenture demonstrate how to realize the new IT strategy, and explore applications in their technology vision by showcasing the advantages of intelligent automation to their clients.

The contributions of this paper are: (1) the architecture and design of a smart IoT closet; and (2) three recommendation techniques for recommending items in the fashion domain.

The rest of the paper is structured as follows. Section 2 introduces the background theory. In Section 3, we give an overview of related work. In Section 4, we describe the smart closet and the user interface for the mobile application, followed by a more detailed description on how the closet is supported by recommendation technology in Section 5. We conclude with a summary, and discuss future work and possible benefits of the system.

# 2 BACKGROUND

## 2.1 Linked Data

Today, most of the data on the Web is uploaded as HTML documents, or raw dumps such as CSV. This way of uploading data sacrifices much of the Web's structure and semantics. Bizer et al. (2009) has outlined a set of rules for publishing data by using the Web to create typed links between the data. By publishing data according to these rules, computer agents can read and make sense of the data published, making it easier to gather meaningful information from the Web. Linked Data published under an open license is called LOD. An example of LOD is Wikidata[5], which is an open knowledge base that can be read or edited by any human or computer agent. Using Wikipedia, Wikivoyage, Wikisource, and others as its central storage, Wikidata contains millions of RDF triples, following a Subject, Predicate, Object structure exposed on its SPARQL endpoint.

## 2.2 Recommender Systems

The objective of a recommender system is to guide users in making choices by giving them personalized recommendations of items. Typically, recommender systems are classified into collaborative filtering and content-based (Jannach et al., 2010). Collaborative filtering generate recommendations on the idea that if some users shared the same interest on previous items, they will have similar preferences to other items as well. Content-based recommendations base its recommendations on item descriptions and a user profile. A user profile is a set of a user's preferences; the recommender system will then recommend the items that have the most similar item description to the user profile.

An interesting challenge for all recommender systems is computing accurate recommendations when few user ratings are available. This challenge is known as the sparsity problem (Jannach et al., 2010). A special case of the sparsity problem is: (a) dealing with new users who have not yet rated any items; and (b) how to recommend new items that has not been rated yet. These two problems are commonly known as the new-user and new-item cold-start problem.

The majority of today's recommender systems addresses recommendations of items in the domains of movies, books, and music. Different techniques of the approaches above are well researched and evaluated in these domains (Bobadilla et al., 2013).

## 2.3 Internet of Things

IoT is a set of Internet-connected devices embedded with hardware, software, sensor, actuators, identifiers, and network technologies. These devices collect and exchange data with each other and other components on the Internet, generating a vast amount of data every day (Gubbi et al., 2013). IoT meets the need for data-on-demand by intuitive interactions with ubiquitous computing devices. Furthermore, IoT has been identified as one of the key trends that organizations must keep track of to gain competitive advantage, and that the market adaptation is predicted to take 5–10 years.

[2]http://research.idi.ntnu.no/SmartMedia/
[3]http://www.ntnu.edu/
[4]https://www.accenture.com/no-en/
[5]https://www.wikidata.org/

The applications in the domains that will be and has been impacted by IoT devices range from control of home equipment such as refrigerators, to monitoring the water quality in cities.

## 3 RELATED WORK

Several works have been done on enabling Linked Data into recommender systems in order to improve their recommendation algorithms (Figueroa et al., 2015).

Heitmann and Hayes (2010) describe a recommender system that tries to mitigate i) the new-item problem; ii) the new user problem; and iii) the sparsity problem of recommendations of music by utilizing Linked Data. They transformed RDF graphs into a user-item matrix by using data from MySpace and data about a Wikipedia editor's homepage. Their results showed that by enabling Linked Data, the average precision increased by 14% and the average recall increased by 33%.

Di Noia et al. (2012) propose a recommender system that relies exclusively on information extracted from the Web of Data. For recommending movies they propose a content-based recommender system using the SPARQL endpoints exposed by DBpedia, LinkedMDB, and Freebase as the base of their recommender system. To compute similarities between movies they used the Vector Space Model, representing the whole RDF as a 3-dimensional matrix where each slice refers to an ontology property. Given a property, each movie is seen as a vector. For a given slice, the similarity is computed between the correlating movie vectors by calculating the cosine angle between the vectors.

Tomeo et al. (2016) generated a dataset consisting of Facebook likes of music, books, and movies. First, they mapped the likes to entities in DBpedia to enrich the item profiles in the dataset. Then, they compared the dataset on different graph-based recommender systems and matrix factorization systems. Overall, the graph-based algorithm, PathRank showed the most promising results.

Many prototypes of smart IoT closets for suggesting outfits by using RFID technology have been made in the past (Goh et al., 2011; Ling et al., 2007; Toney et al., 2006). These prototypes show some very promising techniques for IoT closets and are built on the same fundamental techniques as described in this paper. These techniques involves attaching RFID tags to the clothing items or hangers, which can be scanned by a reader in the closet, and then broadcasting a message about the state of the item to a database. Moreover, similar to our system, some of them also enable weather data or calendar integration (Schaad et al., 2016; Liu et al., 2012). As far as we know, none of them utilize LOD as we do in our system. Moreover, none of them focus on recommending clothing items that the user might want to donate or recycle. Research and evaluation of the recommender algorithms in these studies are lacking or has been scoped out.

Ingvaldsen et al. (2015) propose a recommendation technique for how personalized and location aware news recommendations can be constructed based on the users' contexts. Moreover, they show how the recommended content can be enriched by using Wikidata. In our prototype of Connected Closet, we use similar techniques to combine Wikidata with context aware user ratings to construct location aware recommendations based on the weather at the user's location.

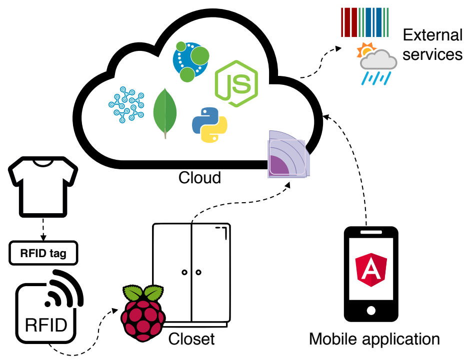



Figure 1: High level architecture.

## 4 ARCHITECTURE

The main parts of the prototype are constructed as follows. As shown in Figure 1, the closet is embedded with a Raspberry PI[6], a tiny computer, which is connected to an RFID reader. When an end-user touches the RFID tag of a clothing item onto the RFID reader, the Raspberry PI will broadcast a message to the backend of the prototype which is constructed of microservices running in the cloud, each performing their own designated task. The mobile application communicates with the microservices, providing the end-user with recommendations and inventory overview. Moreover, the high level architecture con-

[6]https://www.raspberrypi.org/

tains external services which consists of third party APIs, such as weather data and LOD.

The components of Connected Closet are connected as follows.

## 4.1 Closet

The computer embedded in the physical closet runs a Python script listening to scans of clothing items. When a scanning occurs, the script broadcasts a Message Queuing Telemetry Transport (MQTT) message containing timestamp, item id, user id of the closet owner, and the status of the clothing item (whether it is being checked in or out of the closet). Additionally, by using LED lights and speakers connected to the computer, the user receives feedback on an item scan. A red light indicates an insertion, while green light indicates extraction. This is implemented to maintain consistency between the physical clothing items in the closet and the status of the items stored by the microservices. To implement this, we used ideas from Fog Computing (Bonomi et al., 2012) and implemented a local cache database in the embedded computer that keeps track of the status of the latest item scans. Additionally, this prevents the Python script in broadcasting unnecessary messages, such as double scans.

Figure 2: Prototype of the physical closet.

Figure 2 shows a picture of an early version of the prototype, including the physical closet embedded with the Raspberry PI. The prototype is built for demonstration purposes, and to show how human interaction with the closet would work in practice.

## 4.2 Backend

The backend of the prototype (Figure 3) consists of five main components implemented as microservices: the inventory service, the history service, the catalog service, the recommender service, and the closet service. All the microservices and their databases have been implemented using container-based virtualization with Docker[7]. This type of virtualization is an operating-system-level virtualization method for running distributed applications without the need for launching an entire virtual machine. With such container-based microservices, the whole solution benefits from a horizontal scalable architecture composed of small, indepenedent, and highly coupled components communicating with each other by means of Representational State Transfer (REST) with the Hypertext Transfer Protocol (HTTP). The main components of the backend are described throughout the section.

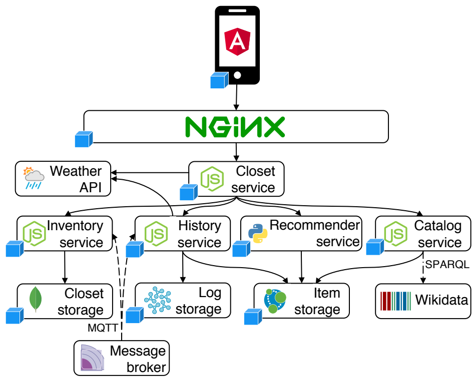



Figure 3: Detailed architecture of the backend.

### 4.2.1 The Inventory Service

The inventory service is responsible for storing data about the clothing items registered to an owner of a closet. The service stores its data in a document-oriented database. Each owner of a closet is assigned their own document in this database. The owner's clothing items are represented as a list of triples containing the unique id of the clothing item, the article number of the item, and the status of the item (inside or outside of closet). Moreover, this service stores user information such as username and favorite outfits. The favorite outfits are represented as a list of tuples containing two items, one top and one bottom.

When the service receives the MQTT message, the service will then take note of which user the scanning came from and update the user's document in the database.

[7]https://www.docker.com/

#### 4.2.2 The History Service

The history service is responsible for logging every scan that occurs in the closet. A time-series database is connected to this service for storing each log entry. For each scanning, a record containing the item id, the timestamp, status of the item, and current temperature will be saved. Furthermore, this service writes usage history and temperatures to a database shared with the recommender and the catalog service.

#### 4.2.3 The Catalog Service

The catalog service is responsible for handling data about all the different clothing items that are supported by a Connected Closet. A supported item is an item connected to an RFID tag with its article number stored in the database shared with the history and the recommender service. This service handles matching clothing items and other item properties, such as color.

Furthermore, this service is set up to communicate with Wikidata via Wikidata's SPARQL endpoint, semantically enriching the clothing items in the shared database. E.g., if one clothing items is registered to have the color 'navy' and another item is 'blue', the results from Wikidata will include similar description to both of these colors, making the similarity between the item descriptions even stronger. This is also done on other item properties where this is expedient.

#### 4.2.4 The Recommender Service

The recommender service lies in the heart of the recommendation approaches explained in Section 5. This service uses the item ratings and the descriptive item data stored in the shared database, called Item storage. This database is implemented as a graph database. Figure 4 shows a simplified example of how the data is represented in the database.

To realize the recommendation approaches, the service employs different machine learning libraries.

#### 4.2.5 The Closet Service

The closet service is responsible for providing the mobile application with meaningful information gathered from the lower level services. It generates a set of the closet overview by joining the data from the catalog service and the inventory service. Using weather data and item status from inventory as input to the recommender service, the recommender service will return a list of recommended outfits. For getting recycling recommendations from the recommender services, it uses data gathered from the history service.

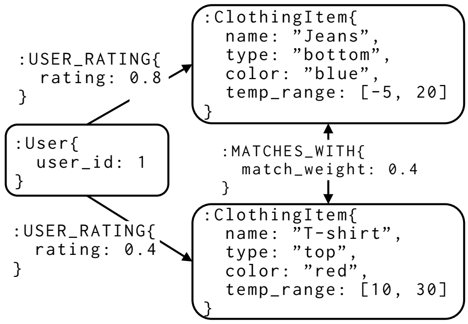



Figure 4: Example of data representation in Item storage.

### 4.3 Mobile Application

A progressive web application is developed to make the closet explorable on mobile devices. In this application, the user is allowed to view suggestions for today's outfit, view an inventory of their closet, and view suggestions on what clothing items to recycle or donate.

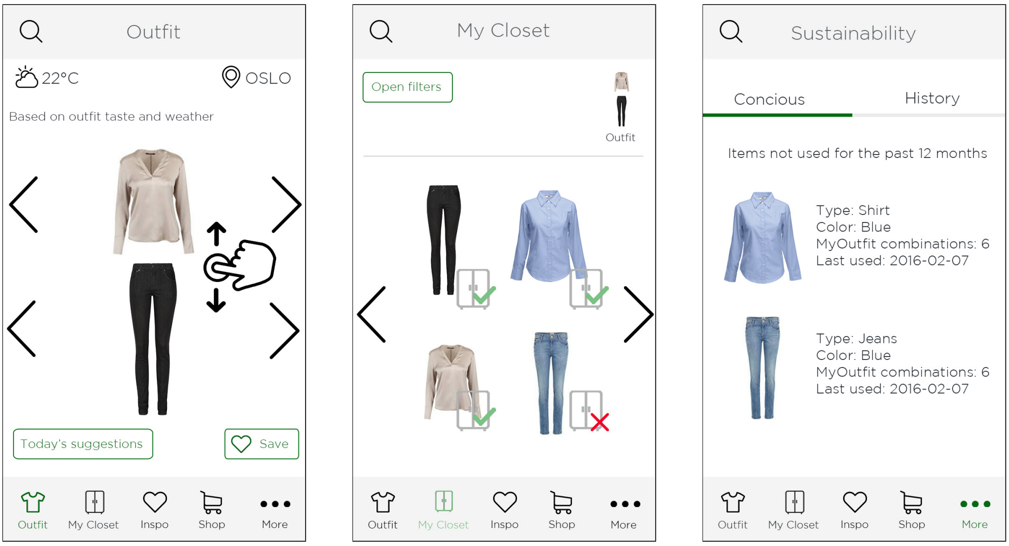
The mobile application communicates with the closet service by REST through HTTP. A web server is set up in the middle to handle traffic and connections using Nginx[8].

#### 4.3.1 User Interface

To view suggestions for today's outfit, the user chooses the Outfit button from the lower menu bar. Figure 5a shows an illustration of the outfit suggestion view. In the top of the view, the weather and location for the user is displayed. Below is the suggested outfit. The user can modify the suggested outfit by clicking on the arrows next to the clothing items of the outfit. If the user wants to go back to what the system has recommended for today, they can use the Today's suggestion button, loading the initial recommendation. The user can save the outfit displayed as a favorite by using the button next to Today's suggestion. Furthermore, the user can browse through a list of top-k outfit recommendations by swiping up and down on the screen.

Figure 5b shows an example of a closet overview. By choosing the My Closet button from the menu, the overview of the user's closet is displayed. Here, the user can browse all the clothing items registered to their closet and see item status indicated by a closet icon with a check mark. Moreover, a filtering function

[8]https://nginx.org/en/




(a) Outfit suggestion (b) Closet overview (c) Recycling suggestion

Figure 5: Screenshots from the Connected Closet prototype.

is implemented to help the user navigate through the closet overview more easily.

By choosing the More button from the menu, the user is displayed with a list of items rarely used and a suggestion to recycle these items. Figure 5c shows a suggestion for two items that have not been used for the past 6 months.

# 5 FASHION RECOMMENDATION

In this section we describe how the whole system is supported by recommender technology. We describe how the item properties needed to do fashion recommendation are determined. Furthermore, we divide fashion recommendation into outfit recommendation and recycling recommendation.

## 5.1 Ratings

The user ratings used in fashion recommendation are determined depending on several context factors. The context factors for determining the user rating of an item are:

- Usage history: How often an item has been extracted from the user's closet affects the item's rating. It is safe to assume that a frequently used item is an item that the user likes. Therefore, an item that is used on a weekly basis will have a high rating.
- Current season: Some clothing items are seasonal. E.g., a winter jacket will have higher rating during the winter, and a low rating in the summer.
- Weather: Much like season, some items are weather dependent, e.g., rain coat. The rating of such items will therefore be affected by the daily weather.
- Taste profile: As the user saves outfits as favorites, the items in the outfits will increase their user ratings.

Using these factors as input, the rating of an item is set using a 10-star rating scale.

## 5.2 Matches and Suitable Temperatures

For determining matching clothing tops and bottoms in the catalog service, we calculate a weight between the two items and assign it to their edge. Initially, all tops and bottoms match each other with a weight of 0.0. When two items are checked out of the closet during the same 2-hour time period, the weight between these items increase with $\alpha$. Furthermore, when a user saves an outfit as a favorite using the mobile application, the weight increases by $\beta$. The matching weight cannot exceed 1.0.

All clothing items are saved in the catalog service with a suitable temperature range property. The suitable temperature range is the range of temperatures in which a clothing item is comfortable to wear. This range is determined by the average temperature of all the checkouts of a clothing item, calculated by Formula 1:

$$st(i) = \frac{1}{N}(\sum_{j=1}^{N} C_{i,j,temp}) \pm \delta\ ^{\circ}\mathrm{C}, \qquad (1)$$

where $N$ is the number of checkouts of clothing item $i$ in $C_i$ and $\delta$ is a constant determining the length of the range.

## 5.3 Outfit Recommendation

In our system, two approaches for outfit recommendation are implemented. The first approach uses ideas from collaborative filtering, while the second approach is a pure content-based approach enabled with LOD.

For an item to be included in a recommended outfit it must be: (1) inside the closet; and (2) the current temperature must be inside the items suitable temperature range.

### 5.3.1 Outfit-Item Matrix

In the first approach, we transform the outfits saved as favorites by the end-users into an outfit-item matrix. In this matrix, each column represents a favorite outfit of an end-user. The rows represent every item supported by Connected Closet and that is part of a user's favorite outfit. Table 1 shows an example of an outfit-item matrix with three outfits and four items. All outfits is associated with a weight, e.g, $w_1$. These weights are based on number of likes of the outfit and are used to determine the strength of the outfit, making it easier to neglect outfits favorited by few users.

Table 1: Example of an outfit-item matrix.

| | Outfit 1<br>$w_1$ | Outfit 2<br>$w_2$ | Outfit 3<br>$w_3$ |
|---|---|---|---|
| Item 1 | × | × | |
| Item 2 | | × | |
| Item 3 | × | | × |
| Item 4 | | | × |

Using this matrix as training data, different classification algorithms are applied to classify outfits as *good* or *neutral*. A *good* outfit means an outfit that can be recommended to the user. While outfits classified as *neutral* are outfits that the users either does not like or has not been rated yet, and will therefore not be recommended to the users. In our method, we first create outfits combinations of the items that fit our inclusion criteria. Then, we input these outfits into the classification model. Outfits that are classified as *good* will then be recommended to the user.

### 5.3.2 Vector Space Model

For our content-based approach, we use a vector space model similar to the one proposed by Di Noia et al. (2012). Using the user ratings stored in the item storage, we build a user profile consisting of clothing items with a rating above $\lambda$, using Formula 2:

$$profile(u) = \{c_i \mid r_{ui} > \lambda\}, \qquad (2)$$

where $r_{ui}$ is rating of clothing item $c_i$ for user $u$.

We then generate a ranked list of all the clothing items in the user's closet using Formula 3:

$$\bar{r}(u, c_i) = \frac{\sum_{c_j \in profile(u)} sim(c_j, c_i)}{|profile(u)|}, \qquad (3)$$

where $sim(c_j, c_i)$ is a similarity measure between the vectors representing the clothing items $c_j$ and $c_i$.

We then filter out a list of top-k outfits based on our inclusion criteria and the match weight between the tops and bottoms.

## 5.4 Recycling Recommendation

For recommending items that may be of interest for recycling by the end-user, the system returns a list of the three lowest rated items that have been rarely used in the past 12 months. By using a time period of minimum 12 months it safe to recommend items that are also seasonal.

An identified problem with the proposed technique is the new user cold-start problem. The owner of a Connected Closet should be able to get relevant recycling recommendations from the day that they acquire the closet. This problem, and other techniques for recycling recommendation will be explored in later research.

# 6 CONCLUSION AND FUTURE WORK

The described prototype is an ongoing project with some development still remaining. A full evaluation and validation will be performed in later research. In this paper we have proposed a novel IoT system for doing fashion recommendation using modern technologies, such as LOD, microservices, containers, and progressive web apps. The proposed system can guide users in making daily outfit selections and efficiently organize their wardrobe in an environment-friendly way. Furthermore, if the system were to be integrated in a clothing retailer's supply chain and used by their customers, the retailer could generate targeted ads and provide relevant recommendations to their customers.

As an initial evaluation, the prototype has been showcased at various IT conferences and events. At these events, the participants have been given a demonstration of the prototype and have had the opportunity to try out the prototype for themselves. The response from the participants has been positive, and many participants have expressed that they would benefit from such a system in their everyday lives.

Future work will be devoted to gathering data for a dataset that can be used for accuracy evaluation of the recommendation approaches. Moreover, we aim to do an user-centric evaluation of the recommender system in order to evaluate the user satisfaction. Some other topics we would like to research further and include in our system are occasion-based outfit recommendations and recommendations from retailers.

## ACKNOWLEDGEMENTS

This work is composed by a research cooperation that was established subsequently of a summer internship at Accenture, where the idea and the first prototype of Connected Closet originally was developed. The authors would like to thank everyone that was involved in the internship for their contributions prior this work.

## REFERENCES

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227.

Bobadilla, J., Ortega, F., Hernando, A., and Gutirrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109 – 132.

Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, pages 13–16, New York, NY, USA. ACM.

Chavan RB (2014). Environmental Sustainability through Textile Recycling. *Journal of Textile Science & Engineering*, pages 1–6.

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., and Zanker, M. (2012). Linked Open Data to Support Content-based Recommender Systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 1–8, New York, NY, USA. ACM.

Figueroa, C., Vagliano, I., Rocha, O. R., and Morisio, M. (2015). A systematic literature review of Linked Data-based recommender systems. *Concurrency and Computation: Practice and Experience*, 27(17):4659–4684.

Goh, K. N., Chen, Y. Y., and Lin, E. S. (2011). Developing a smart wardrobe system. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 303–307.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

Heitmann, B. and Hayes, C. (2010). Using Linked Data to Build Open, Collaborative Recommender Systems. In *AAAI spring symposium: linked data meets artificial intelligence*, pages 76–81.

Ingvaldsen, J. E., Ozgobek, O., and Gulla, J. A. (2015). Context-aware user-driven news recommendation. In *3rd International Workshop on News Recommendation and Analytics (INRA 2015)*. Ceur-ws Vol-1542.

Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., and Shahabi, C. (2014). Big Data and Its Technical Challenges. *Commun. ACM*, 57(7):86–94.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press.

Klepp, I. G. and Laitala, K. (2016). Clothing consumption in Norway. Technical report 2, Oslo and Akershus University College, Oslo.

Ling, S., Indrawan, M., and Loke, S. W. (2007). RFID-Based User Profiling of Fashion Preferences: Blueprint for a Smart Wardrobe. *Int. J. Internet Protoc. Technol.*, 2(3/4):153–164.

Liu, S., Feng, J., Song, Z., Zhang, T., Lu, H., Xu, C., and Yan, S. (2012). Hi, Magic Closet, Tell Me What to Wear! In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 619–628, New York, NY, USA. ACM.

Pruit, J. C. (2015). Getting Dressed. *Popular Culture as Everyday Life*, page 155.

Schaad, P., Basler, S., Medini, M., Wissler, I., Brkle, T., and Lehmann, M. (2016). The Intelligent Wardrobe . *Studies in Health Technology and Informatics*, 225:213–217.

Tomeo, P., Fernndez-Tobas, I., Di Noia, T., and Cantador, I. (2016). Exploiting Linked Open Data in Cold-start Recommendations with Positive-only Feedback. In *Proceedings of the 4th Spanish Conference on Information Retrieval*, CERI '16, pages 11:1–11:8, New York, NY, USA. ACM.

Toney, A. P., Thomas, B. H., and Marais, W. (2006). Managing Smart Garments. In *2006 10th IEEE International Symposium on Wearable Computers*, pages 91–94.

# Chapter 5

# Paper II

Anders Kolstad, Özlem Özgöbek, Jon Atle Gulla and Simon Litlehamar. (2017). **Rethinking Conventional Collaborative Filtering for Recommending Daily Fashion Outfits**. Submitted to the *11th ACM Conference on Recommender Systems (RecSys 2017)*, 8 pages. Author's notification: June 12th 2017.

# Rethinking Conventional Collaborative Filtering for Recommending Daily Fashion Outfits

Anders Kolstad, Özlem Özgöbek, Jon Atle Gulla
Norwegian University of Science and Technology
Trondheim, Norway
andekol@stud.ntnu.no
{ozlem.ozgobek,jon.atle.gulla}@ntnu.no

Simon Litlehamar
Accenture AS
Fornebu, Norway
simon.litlehamar@accenture.com

## ABSTRACT

A conventional collaborative filtering approach using a standard utility matrix fails to capture the aspect of matching clothing items when recommending daily fashion outfits. Moreover, it is challenged by the new user cold-start problem. In this paper, we describe a novel approach for guiding users in selecting daily fashion outfits, by providing outfit recommendations from a system consisting of an Internet of Things wardrobe enabled with RFID technology and a corresponding mobile application. We show where a conventional collaborative filtering approach comes short when recommending fashion outfits, and how our novel approach—powered by machine learning algorithms—shows promising results in the domain of fashion recommendation. Evaluation of our novel approach using a real-world dataset demonstrates the system's effectiveness and its ability to provide daily outfit recommendations that are relevant to the users. A non-comparable evaluation of the conventional approach is also given.

## CCS CONCEPTS

•**Information systems** →**Evaluation of retrieval results;** *Web applications;* •**Computing methodologies** →*Classification and regression trees;*

## KEYWORDS

Recommender Systems, Machine Learning, Fashion Recommendation, Collaborative Filtering, Internet of Things

## 1 INTRODUCTION

Selecting an outfit every morning is a task that many people struggle with, often due to time constraints or the feeling of having nothing to wear. In [20], Pruit argues that our selection of an outfit influences other people's impressions of us, and that it is of high importance to our cultural lives. Moreover, the average Norwegian has 359 unique garments in their closets [15]. This suggests that people need guidance and suggestions for selecting an outfit from their clothing haystack each morning.


*Working paper, Submitted to RecSys '17*

DOI: -

Klepp and Laitala found that 20% of clothes bought by Norwegians were never or rarely used [15]. A reason for this might be that they did not actually like the item they bought or that the item did not match any existing clothing items in their closet. This information is very valuable to the clothing retailer. With such information, the retailer can map the customer's taste profile and generate targeted ads for the customer, reducing the number of unnecessary purchases, and increasing the number of satisfied customers.

Generating such outfit suggestions and targeted ads can be made a reality by recommender systems. A recommender system tries to predict the rating value of a user-item combination, where the user has indicated their ratings for other items in the past [1]. The system tracks these ratings by receiving user feedback. User feedback is classified into *explicit* and *implicit*. Explicit feedback is when the user explicitly rates an item on, e.g., a 5-star scale. Implicit feedback records other user interactions, e.g., how long a user spends on a web page on a certain topic. With the retrieved ratings by user feedback, the recommender system can predict the user's ratings of new items, and suggest the items with a high predicted rating. One of the most successful recommendation technique is called collaborative filtering (CF) [22]. CF recommends items on the assumption that users who have interacted in similar ways before, will have common interests in the future as well. Conventional CF bases its recommendations from a matrix called the *utility matrix*, which captures every rating value for the user-item combinations known to the system [17]. Table 1 shows an example of such a matrix, consisting of user-item combinations of users and movies. A known challenge in CF is called new user cold-start problem. This challenge is about how to recommend items to new users that have not rated any items yet. Suppose we were to introduce a fourth user in Table 1. The user-item combinations for this fourth user would all have '?' as a value. How to then recommend items to this user is not an easy task.

**Table 1: Example of a utility matrix.**

| | Titanic | The Godfather | Pulp Fiction | The Notebook |
|---|---|---|---|---|
| Alice | 5 | 2 | ? | ? |
| Bob | 2 | ? | 4 | 1 |
| Charlie | 4 | 1 | 5 | 4 |

Recommending individual items, such as in Table 1, is what nearly all recommender systems are focusing on. In recent years, recommendations of collections, such as music playlists [12, 13, 23], has gained a lot of attention. Hansen and Golbeck identified some key aspects that affects the recommendation of collections [10]. One aspect that especially applies to outfit recommendation is the co-occurrence interaction effect. Matching clothing items (items that go well together) will have a positive interaction effect when they co-occur together, and will therefore generate a more relevant outfit recommendation to the user.

In [16], we proposed Connected Closet, a system consisting of an Internet of Things wardrobe enabled with an RFID reader, so that clothing items with RFID tags can be checked in and out of the closet, generating implicit feedback on clothing items the user likes. Using a mobile application, the user can give explicit feedback on outfit he likes, and receive daily outfit recommendations based on outside temperature and wardrobe inventory. In this paper, we describe an implementation of the proposed system. We show where a conventional CF approach comes short in terms of the new user cold-start problem and where it fails to capture the co-occurrence effect between items. Moreover, we propose a novel CF approach that mitigate the shortcomings of the conventional approach and implement the novel approach into the proposed system. Evaluations using a real-world dataset are performed on both approaches.

The main contributions of this paper are;

(1) a novel CF approach for recommending daily fashion outfits; and
(2) an accuracy evaluation of the approach using different classification algorithms.

This work is a joint effort between the Smartmedia program[1] at NTNU[2] and Accenture Norway[3]. The Smartmedia program is researching mobile context-aware recommender systems. While, in this work, Accenture's main goal is to research modern technology for building web-based information systems and to keep track of technology key trends, such as Internet of Things.

The rest of the paper is structured as follows. In Section 2, we give an overview of related work, followed by a description of the proposed system in Section 3. Section 4 introduces the concept of outfit recommendation. The recommendation approaches are described in Section 5 and Section 6. Evaluation of the approaches is given in Section 7. We conclude with a summary and discuss future work in Section 8.

## 2 RELATED WORK

There are not many systems addressing daily outfit recommendations from either an Internet of Things wardrobe or a virtual wardrobe. In this section, we give an overview of the state of the art, identify gaps in these works, and show where our system differs from past work and how it complements previous work.

Dumeljic et al. propose a virtual wardrobe implemented as a mobile application [6]. By explicitly stating the user's current mood, the user can add clothing items that best fit the mood, to the virtual inventory. In [6], the outfit recommendation approach is not described and has not been implemented in the system. Moreover, a user study of ten people was conducted, where they concluded that mood is a motivator for selecting outfits, but that users would be more invested in the system if it also considered weather.

In [19], Limaksornkul et al. also propose a mobile application used as a virtual wardrobe. They try to solve the problem of efficiently managing closet inventory and guiding users in selecting clothes based on the user's fashion style, trends, their friends' styles, weather, and occasion. In the mobile application, the users can manage their clothes, and receive statistical-based, weather-based, and event-based clothing suggestions. The statistical-based recommendation engine is preliminary and is the only approach that takes user's preferences into account. Moreover, no evaluation of the system is given.

A smart wardrobe system is proposed by Goh et al. in [9]. Here, garments attached to RFID tags can be scanned in the user's closet. Using a system application, the user can get clothing recommendations based on the user's mood, preferred color or and occasion.

Yu-Chu et al. propose a recommendation system using a modified Bayesian network for generating outfit recommendations from the user's clothing items enabled with RFID tags stored in a smart wardrobe [24]. By taking weather, season, and occasion into consideration, the system first select a top, and then finds bottoms which match the selected top. The process of selecting a bottom depend on user feedback rating the combination. An experiment on 10 users concluded that the proposed system gave more satisfied users than a baseline using a basic Bayesian network without user feedback.

An important aspect that needs to be mentioned is that virtual wardrobes are heavily dependent on explicit user feedback, while the Internet of Things wardrobes can make use of implicit user feedback as well.

As seen in the works above, most of the recommender systems are preliminary, and does not contain clear steps for the recommendation algorithm. The ones that do have an implemented recommender system only have user studies and are lacking accuracy evaluation of their recommendations. In this paper, we describe a fully implemented prototype, using similar architecture to [9] and [24], enabled with a novel recommendation approach evaluated on a real-world dataset. To the best of our knowledge, our novel approach is a completely unique way of generating recommendations using CF. This is mostly because the majority of CF recommender systems today, are heavily based on the utility matrix [22], which is not present in our approach.

## 3 SYSTEM OVERVIEW

In this section, we describe the architecture of the smart wardrobe proposed in [16]. Moreover, we explain how the users receive recommendations through the mobile application which is a part of the architecture. We built and implemented a prototype of the whole system and created a short demonstration video available at https://goo.gl/rZBZqo.

### 3.1 Architecture

Figure 1 shows a high-level view of the architecture. The Closet is embedded with a Raspberry PI[4] connected to an RFID reader. Clothing items enabled with an RFID tag and that has their id number stored in the Cloud, are clothing items that are compatible

[1]http://research.idi.ntnu.no/SmartMedia/
[2]http://www.ntnu.edu/
[3]https://www.accenture.com/no-en
[4]A tiny computer. See https://www.raspberrypi.org/

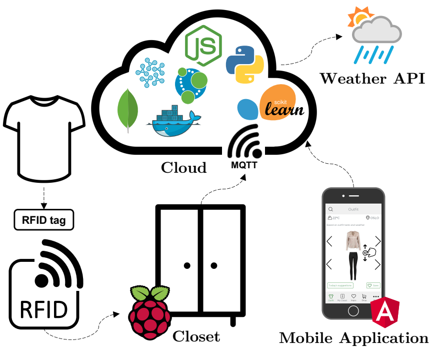



**Figure 1: High level architecture.**

with the system. Such clothing items can be manually scanned through the RFID reader. When a scanning occurs, a message gets broadcasted to multiple services deployed in the Cloud. These services include—among others—a recommender service and an inventory service. By communicating with each other and a third-party Weather API, they provide outfit recommendations to the Mobile Application.

### 3.2 Mobile Application

When the user opens the mobile application, he gets displayed a recommendation for an outfit that suits today's temperature and is inside the user's closet. By swiping through a list, the user is displayed multiple recommended outfits. Moreover, the user can modify the recommended outfit by using the arrows that corresponds to each clothing item. By clicking a Save button, the user gives an explicit positive feedback on the displayed outfit, indicating that the user has this outfit as one of his favorites.


**Figure 2: Screenshot of the mobile application.**

## 4 OUTFIT RECOMMENDATION

We define an outfit, denoted $o$, as a tuple of two items, $c_1$ and $c_2$, where $c_1$ is a top and $c_2$ is a bottom. Although clothing outfits can also contain more, or less, than two items, the current version of our system only addresses outfits of two items. This is with the assumption that most outfits comprise of one top and one bottom. Recommendation of outfits consisting of a one-piece, e.g., a dress, or with additional accessories, is planned for later research.

### 4.1 Inclusion Criteria

To ensure that the user receives outfit recommendations that are relevant for a given day, we define an inclusion criteria for the clothing items that can be part of a recommended outfit. The inclusion criteria are defined as follows:

(1) **Clothing item must be inside the closet.** The status of the item is determined by the latest RFID tag scan.
(2) **Clothing item must be suitable for current weather.** Items are stored in a database with a suitable temperature range property. This is the range of temperatures a clothing item is comfortable to wear. The outside temperature at time of recommendation, must be inside the item's suitable temperature range.

All clothing items that are owned by a user $u_i$ and fits the inclusion criteria is represented as a set $I(u_i)$. All outfit combinations that can be generated from $I(u_i)$ are added to the set $O(u_i)$.

### 4.2 User Ratings

The favored outfits indicated (explicitly or implicitly) by the user, are stored in the system using unary positive-only values. Outfits that have not been rated are outfits that the users either do not like or have not been seen or used together from the user's closet $C(u_i)$. Not rated outfits will be referred to as 'neutral' outfits in the rest of this paper.

### 4.3 Recommended Outfits

The list of recommended outfits that the user receives in the mobile application is generated by the system's recommender service that returns the set $R(u_i)$ of recommended outfits for the user.

### 4.4 Notation

All the notations defined in this section are summarized in Table 2. These notations will be used throughout the paper.

**Table 2: Notations used in this paper.**

| Notation | Description |
|---|---|
| $u_i$ | The $i$th user (owner) of a closet. |
| $c_j$ | The $j$th clothing item. |
| $o_k = (c_1, c_2)$ | An outfit of $c_1$ and $c_2$. |
| $C(u_i) = \{c_1, \ldots, c_l\}$ | Every clothing items the user owns. |
| $I(u_i) = \{c_1, \ldots, c_m\}$ | Clothes fitting the inclusion criteria. |
| $O(u_i) = \{o_1, \ldots, o_n\}$ | Outfit combinations of items in $I(u_i)$. |
| $R(u_i) = \{o_1, \ldots, o_p\}$ | Outfits recommended to the user. |

## 5 RETHINKING CONVENTIONAL CF

In this section, we introduce an approach for outfit recommendation using a conventional utility matrix for collaborative filtering. We discuss where this approach comes short, and introduce a novel approach for outfit recommendation using an outfit-item matrix.

### 5.1 Conventional CF Approach

An obvious solution to recommending fashion outfits is to map the users' favorite outfits onto a utility matrix $U$, consisting of users and outfits. Then, using a neighborhood model, one could predict new outfits for users by comparing the user's interaction pattern with users with same interaction pattern. To recommend the daily outfits $R(u_i)$, we need to match the predicted outfits with the items that fit the inclusion criteria $I(u_i)$, and filter out outfits that does not contain only such items. The approach is illustrated in Figure 6.

The first problem with this approach is that it can only recommend outfits that have been favored by other users. In other words, it cannot generate completely new outfits, and therefore fails to capture the co-occurrence effect between individual items. Another problem with this approach is that it is challenged by the new user cold-start problem. Users who have not favored any outfits or checked out any items, cannot receive recommendations. Lastly, privacy is becoming a huge concern in recommender systems [2, 3], and in this approach, we store all the users' ratings in one centralized matrix, causing a huge risk for the users' privacy.

### 5.2 Novel Outfit-Item Matrix Apprach

By basing our recommendations on the idea that users that have similar items in their closets will also have similar taste in outfits, we propose a novel approach where we rethink the conventional approach by completely transforming the utility matrix. In Figure 4, we create a matrix $Z$, where the columns represent outfits, and the rows represent the clothing items that compose the outfit. Each outfit is associated with a weight $w$. This weight is the number of users who have favored an outfit. Using $Z$ and $W$, we train a classifier using a classification model. Outfits that have been favored by users and have an associated weight above 0 will be classified as 'positive', while outfits with an associated weight of 0 will be classified as 'neutral'. When the model has been trained, we generate all the possible outfit combinations $O(u_i)$, of the items that fit the inclusion criteria for the given user $u_i$. By using the classifier, we can now recommend the outfits that are classified as 'positive' to the user $R(u_i)$.

The advantages of this approach are that it captures the co-occurrence interaction effect between two clothing items. This is because it considers the clothing items that an outfit is composed of, instead of just looking at the outfits as a whole. Moreover, it is not challenged by the new user cold-start problem because we assume that people that own similar clothing items will have same taste in outfits as well. Lastly, this approach has a huge advantage in terms of user privacy, because it does not need to store the user-item combinations in one centralized matrix.

In Figure 5, we give an example of a possible recommendation pipeline that can occur in our system using the novel approach. To the left is the set of all the clothing items owned by the user. By inputting this and the current outside temperature at the user's location, the function $f_1$ filters out and generates possible outfits for recommendation wrt. the inclusion criteria. These outfits are then inputted to $f_2$, which follows the same steps as described in Figure 4. In the end of the pipeline, we get the generated set of recommended outfits that is displayed in the mobile application.

Although not implemented in our system, this approach could be easily used by a clothing retailer to generate targeted ads by inputting clothing items from the retailer together with the user's clothing items in $C(u_i)$. Then, the clothing retailer could recommend new outfits that the users might want to buy, or individual items that would make a great outfit with clothing items already owned by the user.

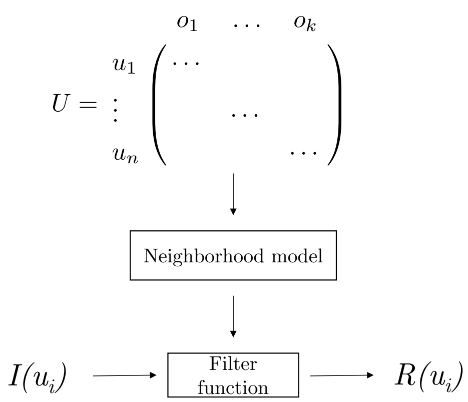



**Figure 3: Conventional CF approach using a utility matrix.**

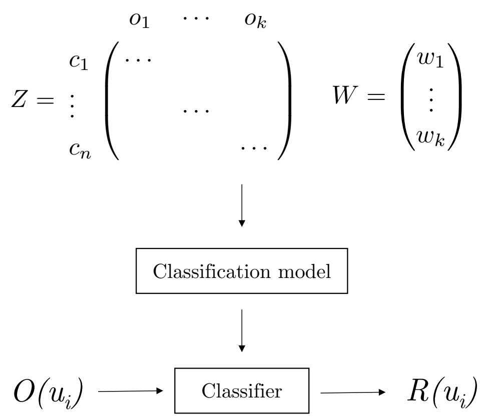



**Figure 4: Novel approach using an outfit-item matrix.**

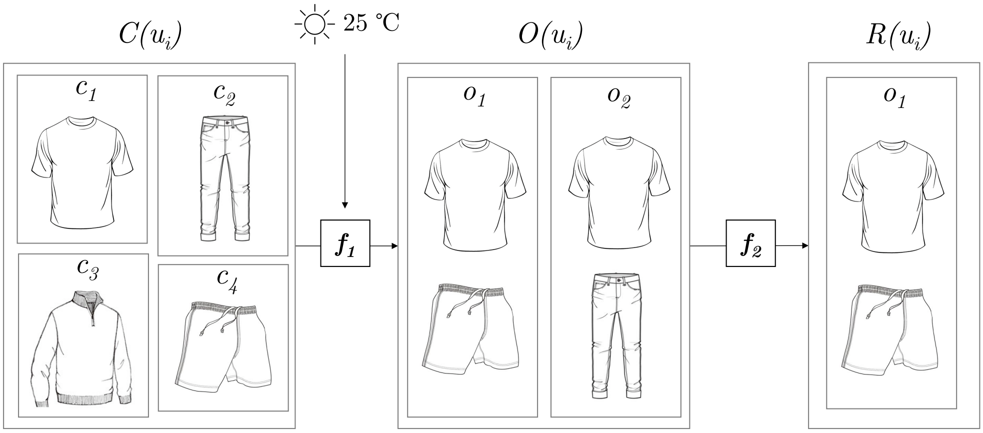



**Figure 5: Example of a possible recommendation pipeline using the novel approach.**

## 6 RECOMMENDATION MODEL

In this section, we present the recommendation model for our novel approach using different classification models. The chosen classification models are widely known and perform well in many domains [4, 5]. The classification models also include a baseline classifier. Moreover, we introduce some neighborhood models that are applied with the conventional approach.

### 6.1 Classification Models

*Naïve Bayes.* Assuming the attributes of the samples are conditionally independent and given the sample's class labels, Naïve Bayes assigns a test sample the class label $Y$ by maximizing the numerator in this equation [18]:

$$P(Y \mid \boldsymbol{X}) = \frac{P(Y)\prod_{i=1}^{d} P(X_i \mid Y)}{P(\boldsymbol{X})}, \tag{1}$$

where $\boldsymbol{X}$ is a set of $d$ attributes.

*Adaptive Boosting (AdaBoost).* Over the recent years, classification techniques known as ensemble methods have gained a lot of attention. One of the most popular ones is AdaBoost. It aggregates over a set of weak learners $h_t(x)$ that tends to perform slightly better than a random classifier. The final classifier $H(x)$ is then obtained by ensembling the weak learners by a weighted majority voting scheme using this equation [7]:

$$H(x) = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big), \tag{2}$$

where $\alpha_t$ is the assigned weight for each weak learner.

To pick the weak learners, each training sample is associated with a weight indicting its importance. AdaBoost will then pick its weak learners in a forward stage-wise manner by focusing on predicting the high-weight samples correctly.

*Gradient Boosting.* Another popular ensemble method that relies on a set of weak learners is called Gradient Boosting. It follows the same fundamental idea as AdaBoost, but instead of focusing on the sample weights when picking its weak learners, it focuses on gradients [8].

*Uniform.* As a baseline, we use a classifier that generates class predictions uniformly at random.

### 6.2 Neighborhood Models

To predict the ratings of the user-outfit combinations in the matrix $U$, given in Figure 6, we apply the user-based neighborhood model [1]. This model predicts user ratings by finding users that have rated similar outfits. To find similar users, we can apply different similarity measures. In our model, we apply Jaccard (JAC) and cosine similarity (COS) as defined by Equation 3:

$$Sim_{JAC}(\boldsymbol{A}, \boldsymbol{B}) = \frac{|\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|} \qquad Sim_{COS}(\boldsymbol{A}, \boldsymbol{B}) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{||\boldsymbol{A}||\ ||\boldsymbol{B}||} \tag{3}$$

After user similarities have been calculated we can predict the ratings $\hat{r}_{ui}$ of unrated outfits using this formula:

$$\hat{r}_{ui} = \frac{\sum_v Sim(u,v) r_{vi}}{\sum_v |Sim(u,v)|} \tag{4}$$

### 6.3 Ranking Model

To rank the outfits that are predicted to the user in $R(u_i)$, using the novel approach, we assign each prediction of an outfit $o_j$ to a ranking score equal to the classifier's probability of the class label being 'positive' $P(w_j > 0 \mid o_j)$. It should be noted that this is not a personalized ranking model, but as seen from our results, it performed well for each individual user.

The conventional approach does not use classification models, so the probability of the predicted class label is not available. Instead, the outfits are ranked according to the predicted rating calculated using the similarity measures.

## 7 EXPERIMENTS

In this section, we describe the setting for how our experiment was performed. We give a detailed description of the dataset that was used and present the results of the different models that were evaluated. The main goals of the experiments are to demonstrate the effectiveness of the system and to compare and select the best classification model for our system.

### 7.1 Dataset

The dataset is scraped from Polyvore.com[5]. Polyvore is a social media site where users can create clothing outfits by matching individual clothing items. Other users can then 'like' these outfits by a clicking a 'like button'.

From the available outfits at Polyvore, we first gathered the most liked outfits from the last 3 months. For these outfits, we filtered the outfits so that they only contained a top and a bottom. Then, we collected other outfits that these items also were a part of, and filtered them. Lastly, we gathered all the user likes for each of the outfits we had gathered. Table 3 describes the size of the dataset.

**Table 3: Data statistics on Polyvore dataset.**

| # Outfits | # Clothes | # Users | # Likes |
|---|---|---|---|
| 6,186 | 158 | 7,093 | 19,287 |
| **Positive:** 260 | **Tops:** 81 | | |
| **Neutral:** 5,917 | **Bottoms:** 87 | | |

From the gathered dataset, we have 260 outfits that are classified as 'positive' and 5,917 that are classified as 'neutral'. This means that the dataset has an imbalance approximately of 23 to 1.

In total, there are 158 individual clothing items in the dataset. This means that the feature vectors used in the classification models will be relatively sparse binary vectors of 158 dimensions.

### 7.2 Evaluation Methods

To evaluate our novel approach, we iterated through the following procedure for all users with at least 20 outfit likes: For all off the user's favorite outfits, we hide each of the user's ground-truth favorite outfits from the system by decreasing the outfits' corresponding weights in $W$ by 1. Then, we train the classification model using $Z$ and $W$. Moreover, with the assumption that a user only own items that are part of the items the user likes, we generate outfit combinations, assuming all of the items in $C(u_i)$ fit the inclusion criteria. We then compared the predicted class labels of the generated outfits combinations to the true favorite outfits of the user. We also ran the procedure a second time, but now by randomly removing 50% of the users' tops and bottoms in $C(u_i)$. This was done to simulate outfit recommendations from a half empty closet. In Table 4, we summarize some statistics for the test sets that was generated by running these methods. As seen in this table, there are—on average—quite many outfits that are being classified for each user $O(u_i)$, compared to the true number of the user's favorite outfits $O(u_i)^{TP}$.

[5]http://www.polyvore.com/

**Table 4: The average properties for the users in the test sets.**

| **Closet size** | $avg(\lvert O(u_i)\rvert)$ | $avg(\lvert O(u_i)^{TP}\rvert)$ |
|---|---|---|
| Full | 682.5 | 31.4 |
| Half empty | 164.0 | 17.6 |

To reduce the dimensionality of the samples and to detect items that are interrelated, the multivariate analysis technique called *principal component analysis* was applied to the samples before training the models [14]. The reduction is done by transforming to a new set of uncorrelated features ordered so that the first ones retain most of the original variation.

For evaluating the conventional approach using the different neighborhood models, we first randomly removed 30% of the user likes from the utility-matrix. Then, we predicted all outfit likes for each user, and filtered them out wrt. $I(u_i)$ using the same assumption above. The recommended outfits were then compared to the true outfit likes.

### 7.3 Evaluation Metrics

If we look at the task of recommending the outfits as retrieving all relevant items (outfits) from a collection of outfits separated into the two classes; relevant and not relevant, we can apply the popular accuracy metrics from information retrieval systems. In our case, we say that the relevant outfits are the ones classified as 'positive', and the not relevant are the outfits classified as 'neutral'. Then, we can use a popular metric known as Recall. It measures the ratio of relevant items retrieved to the number of all relevant items available [11]:

$$\text{Recall} = \frac{|\text{relevant items retrieved}|}{|\text{all relevant items}|} \tag{5}$$

In this paper, we also report Recall@N, which is the Recall in a ranked list just considering the N first elements. We compute Recall and Recall@N by averaging over the result for each user $u_i$.

A way to graphically display the tradeoff between the true positive rate and the false positive rate, is known as a receiver operating characteristic (ROC) curve. The true positive rate is the same as Recall, and the false positive rate is the ratio of non-relevant items retrieved to the number of all non-relevant items available. The ROC curve is great to compare the performance difference between classifiers, where the best classifiers tend to be located in the upper left corner of the diagram. The classifiers that performs best on average will have a large area under the ROC curve (AUC) [11].

To evaluate the ranking via utility, we sum the utility of an outfit $j$ to a user $u$ over a ranked recommended list of size $L$. By summing over this value for each user, we obtain the R-score as follows [1]:

$$\text{R-score} = \sum_{u=1}^{m} \sum_{j \in I_u, v_j \leq L} \frac{max\{r_{uj}, 0\}}{2^{(v_j - 1)/\alpha}}, \tag{6}$$

where $v_j$ is the rank of outfit $j$ and $r_{uj}$ is the ground-truth rating of outfit $j$. $\alpha$ is the half-life, set to 5 in our experiments. The higher the R-score is, the true favorite outfits for each user tend to appear in the top of the ranked list.

**Table 5: Results from evaluation of novel approach.**

| Model | Overall AUC | Overall Accuracy | Overall Recall | Top-L R-score | Top-L Recall@5 | Top-L Recall@10 | Top-L Recall@15 | Top-L Recall@20 |
|---|---|---|---|---|---|---|---|---|
| *Naïve Bayes* | .704 | .870 | .756 | 566.2 | .091 | .183 | .287 | .382 |
| *Gradient Boosting* | .864 | .878 | .997 | 851.9 | .111 | .228 | .337 | .448 |
| *AdaBoost* | .885 | .723 | .978 | 872.1 | .113 | .223 | .334 | .442 |
| *Uniform* | .500 | .500 | .493 | 88.0 | .024 | .052 | .077 | .095 |

## 7.4 Results and Discussion

In this section, we present our results and discuss some insight we obtained while running the experiments. By the end of this section we will have answered the following questions:

- Q1. How do the different classification models compare using our novel approach?
- Q2. How does closet size affect the recommendation results?
- Q3. To what extent can the conventional approach be used to recommend new outfits to the users?

The evaluation method for the novel approach was performed using the classification models in Section 6.1. For Naïve Bayes the best configuration was setting a prior probability for the 'neutral' class label to 0.99 and a 0.01 prior probability for the 'positive' class. This was mostly due to the 23 to 1 imbalance in the dataset. AdaBoost gave the best result using decision trees as weak learners and with a learning rate of 1.0. Gradient Boosting performed best with similar configurations.

In Table 5, we report AUC, Accuracy and Recall for the predicted class labels for all of the outfits that were tested when simulating a full closet. In the right-hand side of the table, we also report the R-score and Recall@N in a ranked list of L outfits. Because each user has different numbers of clothes in their closet, every user is recommended a ranked list of various lengths of L. The best performing model in each category is highlighted by underlining its result. As seen in the table, Gradient Boosting and AdaBoost are the dominating models in all categories. On average and overall, Gradient Boosting performs best, while in a top-L ranked list, AdaBoost performs slightly better. For N > 5, Gradient Boosting was—at maximum—only .006 points better than AdaBoost in Recall@N. In terms of the R-score, AdaBoost is superior to Gradient Boosting. Because of this, we conclude that AdaBoost is the model yielding highest utility to the users.

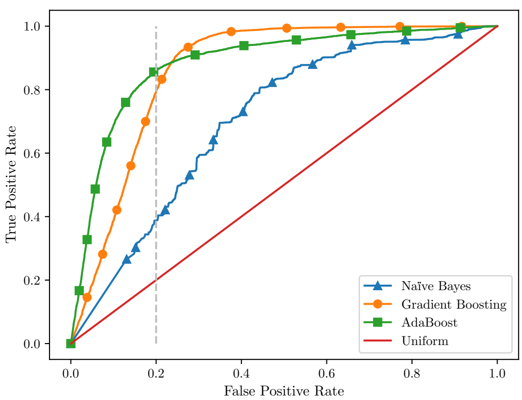



**Figure 6: Global ROC curves for recommendations from a full closet.**

In Figure 6, we plot a ROC curve for the different models used to generate a single ranked list of user-outfit pairs. This type of ROC curve is sometimes referred to as a global ROC curve [21]. As indicated by the gray dotted line, AdaBoost is the best model at a false positive rate at 20%, predicting 86% of the users' favorite outfits. As the false positive rate increase, Gradient Boosting becomes slightly superior to AdaBoost. On average, Gradient Boosting and AdaBoost dominates the two other models with an AUC of .864 and .885, respectively. Naïve Bayes yields a satisfactory AUC of .704, while from the Uniform model we got an expected AUC of .500.

The high values of AUC and R-score are a strong indication that the non-personalized ranking model performs quite well and even better than expected.

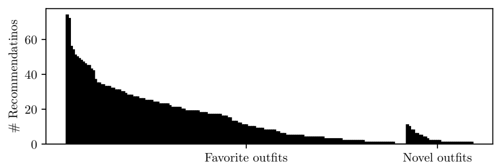



**Figure 7: Distributions of outfit recommendations using AdaBoost.**

Figure 7 shows the distributions of outfit recommendations in a top-20 list recommended to the users with at least 20 outfit likes. In total, 196 unique outfits were recommended to the users, where 33 of them were novel outfits—never favored by any users in the past. This shows that a wide range of outfits end up in the users' recommended top lists.

Experiment on a half empty closet resulted in no change in terms of overall Recall, and at most, a .005 decrease in AUC, and for this reason, we do not report any results beyond this. Besides the fact that few clothing items will result in fewer outfit recommendations, we conclude that closet size has little effect on the recommendations.

In Table 6, results from evaluation of the conventional approach is given. The table shows Recall@N in a ranked list of M outfits. Because M is much lower than L, we only report up to N = 5 (as

**Table 6: Evaluation of the conventional approach.**

| Model | Recall@1 | Recall@5 |
|---|---|---|
| *Cosine* | .077 | .366 |
| *Jaccard* | .050 | .250 |

opposed to N = 20 in evaluation of the novel approach). Note that the results are not comparable to the results in Table 5, as they are derived using an approach that is fundamentally different. The best performing model is highlighted with underlined results. As the numbers indicates, the approach generates new outfit recommendations to the users at with a satisfactory accuracy. However, these outfit recommendations are—as argued in Section 5—only outfits that have been composed and favored by other users in the past. Therefore, we conclude that this approach is insufficient when it comes to recommending novel and personalized daily outfits.

## 8 CONCLUSION AND FUTURE WORK

We have introduced a novel approach for recommending daily fashion outfits from a smart closet. Our novel approach mitigate a wide range of challenges faced by a conventional approach that tries to recommend daily fashion outfits. Evaluation of our novel approach demonstrates the method's effectiveness, and its ability to provide users with accurate and novel outfit recommendations.

The results from the evaluation helped us select which model to deploy in the system. R-score, AUC, and Recall@N are the most useful measures regarding each individual user. Since, AdaBoost achieved the highest R-score and AUC, it was chosen as the main classifier and implemented with the novel approach in the recommender service deployed in the cloud. It should be noted that Gradient Boosting achieved slightly better results in Recall@N, but we regard this difference as insignificant and conclude that AdaBoost is indeed the best fit for our system.

A non-comparable evaluation of the conventional approach was performed to see to what extent it could recommend daily outfits. The accuracy results are acceptable, but due to the approach's many challenges, it cannot be considered as an efficient method for recommending daily outfits.

Although we have demonstrated the system's performance using a real-world dataset, a full scale evaluation using data gathered from physical clothes enabled with RFID tags is planned for future work. The current state of the system should be considered as an early prototype and is premature for such a full scale evaluation. Because of this, these plans are preliminary and we consider other research topics to be more important at the current stage. These topics include content-based outfit recommendation and recommendation of garments to be recycled or donated. With these research topics, we intend to incorporate additional contextual factors such as season, user's occasion, and user's body type.

## ACKNOWLEDGMENTS

This work is an extension to a prototype of the proposed system initially developed during an internship at Accenture. The authors would like to thank everyone involved in the internship for their contributions prior this work.

The authors would also like to thank everyone at Accenture who has provided valuable feedback on this research.

## REFERENCES

[1] Charu C. Aggarwal. 2016. *Recommender Systems: The Textbook* (1st ed.). Springer Publishing Company, Incorporated.
[2] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying Differential Privacy to Matrix Factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 107–114.
[3] Smriti Bhagat, Udi Weinsberg, Stratis Ioannidis, and Nina Taft. 2014. Recommending with an Agenda: Active Learning of Private Attributes Using Matrix Factorization. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, New York, NY, USA, 65–72.
[4] Rich Caruana and Alexandru Niculescu-Mizil. 2006. An Empirical Comparison of Supervised Learning Algorithms. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*. ACM, New York, NY, USA, 161–168.
[5] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems (MCS '00)*. Springer-Verlag, London, UK, UK, 1–15.
[6] Bojana Dumeljic, Martha Larson, and Alessandro Bozzon. 2014. Moody Closet: Exploring Intriguing New Views on Wardrobe Recommendation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval (GamifIR '14)*. ACM, New York, NY, USA, 61–62.
[7] Yoav Freund and Robert E Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* 55, 1 (Aug. 1997), 119–139.
[8] Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29 (2000), 1189–1232.
[9] K. N. Goh, Y. Y. Chen, and E. S. Lin. 2011. Developing a smart wardrobe system. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. 303–307.
[10] Derek L. Hansen and Jennifer Golbeck. 2009. Mixing It Up: Recommending Collections of Items. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1217–1226.
[11] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53.
[12] Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. 2016. Music Personalization at Spotify. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 373–373.
[13] Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. 2015. Beyond "Hitting the Hits": Generating Coherent Music Playlist Continuations with the Right Tracks. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 187–194.
[14] Ian Jolliffe. 2002. *Principal component analysis*. Wiley Online Library.
[15] Ingunn Grimstad Klepp and Kirsi Laitala. 2016. *Clothing consumption in Norway*. Technical report 2. Oslo and Akershus University College, Oslo. In Norwegian.
[16] Anders Kolstad, Özlem Özgöbek, Jon Atle Gulla, and Simon Litlehamar. 2017. Connected Closet - A Semantically Enriched Mobile Recommender System for Smart Closets. In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*. 298–305.
[17] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of Massive Datasets* (2nd ed.). Cambridge University Press, New York, NY, USA.
[18] David D. Lewis. 1998. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proceedings of the 10th European Conference on Machine Learning (ECML '98)*. Springer-Verlag, London, UK, UK, 4–15.
[19] Chantima Limaksornkul, Duangkamol Na Nakorn, Onidta Rakmanee, and Wantanee Viriyasitavat. 2014. Smart Closet: Statistical-based apparel recommendation system. In *Student Project Conference (ICT-ISPC), 2014 Third ICT International*. IEEE, 155–158.
[20] John C Pruit. 2015. Getting Dressed. *Popular Culture as Everyday Life* (2015).
[21] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and Metrics for Cold-start Recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02)*. ACM, New York, NY, USA, 253–260.
[22] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. in Artif. Intell.* 2009, Article 4 (Jan. 2009).
[23] Andreu Vall. 2015. Listener-Inspired Automated Music Playlist Generation. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 387–390.
[24] Lin Yu-Chu, Yuusuke Kawakita, Etsuko Suzuki, and Haruhisa Ichikawa. 2012. Personalized Clothing-Recommendation System Based on a Modified Bayesian Network. In *Proceedings of the 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet (SAINT '12)*. IEEE Computer Society, Washington, DC, USA, 414–417.

# Chapter 6

# Paper III

Anders Kolstad, Özlem Özgöbek, Jon Atle Gulla and Simon Litlehamar. (2017). **Content-Based Recommendations for Sustainable Wardrobes using Linked Open Data**. Submitted to the *13th International Conference on Semantic Systems (SEMANTiCS 2017)*, 4 pages. Author's notification: July 3rd 2017.

# Content-Based Recommendations for Sustainable Wardrobes using Linked Open Data

Anders Kolstad, Özlem Özgöbek, Jon Atle Gulla
Norwegian University of Science and Technology
Trondheim, Norway
andekol@stud.ntnu.no
{ozlem.ozgobek,jon.atle.gulla}@ntnu.no

Simon Litlehamar
Accenture AS
Fornebu, Norway
simon.litlehamar@accenture.com

## ABSTRACT

Through recycling textile waste, greenhouse gas emissions can drastically be reduced. Many clothing retailers today offer checkpoints where people can recycle their clothing items. Moreover, people today are often challenged by overloaded wardrobes and store many clothing items that they never use. In this paper, we describe a system that creates incentives for the users to recycle their clothes, benefiting the environmental sustainability. We propose a content-based recommendation approach that utilizes semantic web technologies and that leverages a set of context signals obtained from the system's architecture, to recommend clothing items that might be relevant for the user to recycle. Experiments on a real-world dataset shows that our proposed approach outperforms a baseline which does not utilize semantic web technologies.

## CCS CONCEPTS

**•Applied computing →Consumer health; •Information systems →***Evaluation of retrieval results;* Data mining;

## KEYWORDS

Recommender Systems, Content-Based Recommendation, Textile Recycling, Linked Open Data, Bag of Concepts

## 1 INTRODUCTION

In today's world, overconsumption is becoming a huge concern. The equivalent for fast food called 'fast fashion' is becoming a phenomenon, and people continue to purchase large quantities of clothes. Because of this, textile recycling has become a pressing issue. Textile recycling has old history, but was earlier often just concerned with the economic benefits. As of now, the environment has become the number one factor for doing textile recycling. For every pound of recycled textile, more greenhouse gas emissions are prevented than for every pound of glass, plastic, and paper—combined [9]. Moreover, Klepp and Laitala found that 20% of the clothes in people's wardrobes were never or rarely used [4]. Because of this, clothing retailers are now starting to offer checkpoints where people can recycle their clothes in order to enhance the environmental sustainability.

It is important that people get incentives to use these recycling checkpoints. Such incentives can be generated by recommender systems and Internet of Things technology. Traditionally, recommender systems try to predict items that might be of interest to the users. A popular technique for recommender systems is known as *content-based* [6]. Content-based recommendations builds a user profile of item properties that the user has shown interest of in the past, and then compute item similarities with other items that the user has not seen yet. In previous studies, content-based recommender systems enabled with semantic web technologies have shown promising results by increasing the accuracy of the recommendations [2]. Linked Open Data (LOD) is a semantic web technology that forms a set of rules for publishing data so that the data become machine-readable and free of use for anyone [1].

In [5], we proposed a system called Connected Closet—a smart closet where clothing items enabled with RFID tags can be scanned in the user's closet, tracking the usage history of the clothes. In this paper, we propose a semantic content-based recommender system that leverages a set of context signals obtained from the system's architecture to provide recycling suggestions to users of the system. By utilizing semantic web technology, the recommender system's accuracy improves. Moreover, it can improve the system's transparency and increase the user's trust and confidence in the system. The main contributions of this paper may be summarized as follows:

- We propose a content-based recommender system that utilizes LOD to recommend clothing items to be recycled.
- We evaluate our proposed recommender system on a real-world dataset and compare it to a baseline that does not utilize semantic web technology.

The remainder of this paper is structured as follows. In Section 2, we describe the contextual models addressed when computing the recommendations. Then, we describe our recommendation algorithm in Section 3. In Section 4, we evaluate our proposed approach and discuss related work in Section 5. We conclude with a summary and discuss future work in Section 6.

## 2 CONTEXTUAL MODELS

In this section, we describe the contextual signals obtained from the proposed system's architecture depicted in Figure 1. We describe how they are obtained and how they affect the recommendation algorithm for recycling recommendations.


*Working paper, Submitted to SEMANTiCS '17*

DOI: -

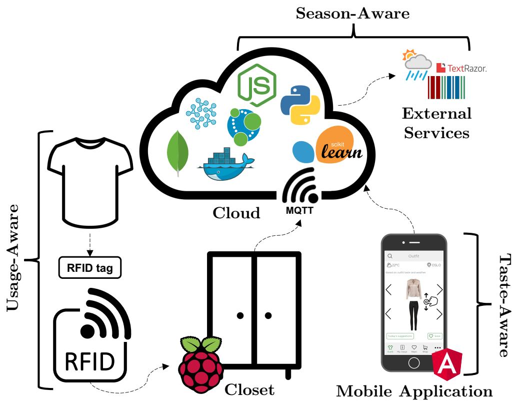



**Figure 1: Architecture for a sustainable wardrobe.** The figure indicates where in the system the contextual signals are obtained.

### 2.1 Usage-Aware

The user's clothing items are enabled with RFID tags. These can be manually scanned through an RFID reader connected to a tiny computer embedded in the user's Closet. The computer will broadcast a message containing information about the scan to a set of services deployed in the Cloud. Each scan is added to the set $S$.

Clothing items that are often checked out of the user's closet will achieve a higher user rating. More formally, based on item usage, we calculate the user's rating of a clothing item as follows:

$$\hat{r}_\mu(u,i) = \frac{1}{2}|S_{u,i}|, \tag{1}$$

where $S_{u,i}$ is the set of all scans of clothing item $i$ done by user $u$.

### 2.2 Taste-Aware

In the Mobile Application, the user can save his favorite outfits. An outfit is represented as a tuple (one top and one bottom). The user's favorite outfits are added to the set $O_u$. Items that occurs in many outfit combinations will achieve a high rating. More formally, the user rating based on the user's favorite outfits is defined as follows:

$$\hat{r}_\tau(u,i) = |\{(j,k) \in O_u \mid j = i \vee k = i\}| \tag{2}$$

### 2.3 Season-Aware

The usage pattern of some items might only occur during a season, e.g., a winter coat will only be used in the winter. Such seasonal clothing items are assigned to a season {winter|spring|summer|fall}. If a seasonal clothing item is recommended for recycling, we check if the item was used during the last assigned season. This is done by looking up the latest item scans in $S$. If the item was used during the season, the item is removed from the recommended list and not displayed to the user.

## 3 SEMANTIC CONTENT-BASED RECOMMENDER SYSTEM

In [5], we proposed a model for recycling recommendations that recommended the lowest rated items. Although, considering how the ratings are obtained from the context signals, newly bought clothing items would always be recommended for recycling when using this model. In this paper, we propose a content-based recommender system that recommends the items that are as least similar as possible to the user profile.

### 3.1 Vector Space Model

In our recommendation approach, we adopt the Vector Space Model, where we represent each clothing items as a vector. We then use the *Bag of Concepts* [10] approach to create the vectors using entities from `Wikidata`[1], which is a Knowledge Base published as LOD. As a weighting scheme, the Concept Frequency (CF) is used. Here, the term concept refers to the `Wikidata` entities.

The user profile is represented as a set of clothing item vectors:

$$profile(u) = \{\vec{i} \mid \hat{r}(u,i) > \lambda\}, \tag{3}$$

where $\hat{r}(u,i)$ is an aggregate of $\hat{r}_\mu(u,i)$ and $\hat{r}_\tau(u,i)$, and $\vec{i}$ is the CF vector representing clothing item $i$.

The user's clothing items are then sorted ascending in a ranked list using this scoring function:

$$\bar{r}(u,i) = \frac{\sum\limits_{\vec{j} \in profile(u)} dist(\vec{i},\vec{j})}{|profile(u)|}, \tag{4}$$

where $dist(\vec{i},\vec{j})$ is a distance measure between the vectors representing the clothing items $i$ and $j$. In our approach we use the *Euclidean distance* defined as follows:

$$dist(\mathbf{q},\mathbf{p}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}, \tag{5}$$

where $\mathbf{q}$ and $\mathbf{p}$ are both vectors of $n$ dimensions.

### 3.2 Semantic Item Representation

Figure 2 shows an excerpt of the services deployed in the Cloud. When new clothing items are fed into the system, they are inputted with a free text description to the Catalog service. To represent the clothing items as vectors using `Wikidata` entities with CF, the following process is performed on the item's text description (the location of where in the system each step is performed is depicted in Figure 2):

*(1) Entity extraction.* We extract `Wikidata` entities by using the `TextRazor API`[2]. For disambiguating entities, each entity is ranked with a confidence score based on multiple signals in the text.

*(2) Weighting.* We generate vectors using the CF weighting scheme from the entities returned by Step 1.

*(3) Storing.* The vectors are then stored in the system's graph database called Item storage.

In the system's current stage, removal of stop entities and generic entities, is not addressed and will be included in the process in future research.

[1] https://www.wikidata.org/
[2] https://www.textrazor.com/

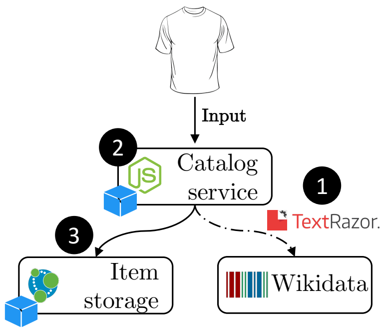



**Figure 2: Excerpt of the system's services deployed in the Cloud.** An illustration of the process of a new item being stored in the system.

## 4 EXPERIMENTS

To demonstrate the validity of our approach, we perform an evaluation on a dataset collected from the Web.

### 4.1 Dataset

In order to evaluate our approach, we desired a dataset consisting of users with outfit ratings, the user's usage history on clothing items, and information on which clothing items that have been recycled. Because our system is not yet in full scale production and there seems to be little available data in this domain, we could only obtain a dataset addressing outfit ratings.

The dataset was collected from the social media site Polyvore[3]. Polyvore is a site where users can create fashion outfits. Other people can rate these outfits using a 'like' button on a unary rating scale. This mirrors the functionality found in the smart closet's mobile app. The collected dataset consists of 260 rated outfits composed by 158 clothing items, 7093 users and 19287 outfit ratings.

Figure 3 shows an example from one of the items in the collected dataset. The top shows the representation of the item using the classic *Bag of Words* approach with a Term Frequency (TF) vectors. Below, is the representation of the item after it has gone through the process described in Section 3.2. The blue node (`item`) represent the clothing item, while the green nodes (labels starting with 'Q') represent the `Wikidata` entities describing the clothing item. To visualize the context signals, we have included one user's interactions with the item. In the figure, $\hat{r}_\mu(u,i)$ (USAGE_COUNT) and $\hat{r}_\tau(u,i)$ (OUTFIT_LIKES) are represented as relations from the `user` node to the `item` node.

### 4.2 Evaluation Method

Due to the nature of the dataset, we can only consider the ratings obtained by the Taste-Aware context signal. For this reason, the other signals are neglected in this experiment. Moreover, the dataset does not contain clothing items that have been recycled by the users. This means that we need to make an assumption for when a clothing item is relevant for recycling. In this experiment, a clothing item

[3]http://www.polyvore.com/

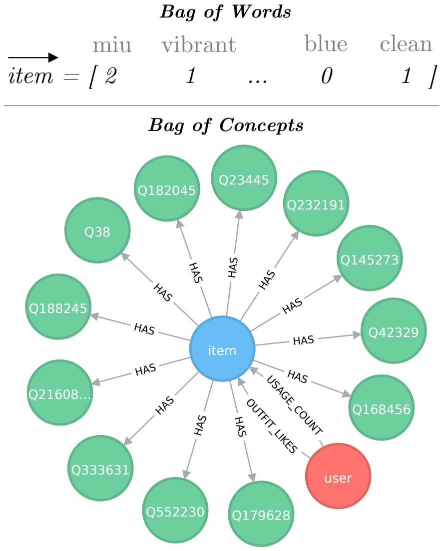



**Figure 3: Representation of a sample item.** Text description: *"Miu Miu's vibrant Resort '17 collection is inspired by the '90s rave scene. Knitted in a kaleidoscope of hues, this cropped sweater has sumptuous touches of wool, mohair and [...] Dry clean.. Made in Italy.."*. Given this description, the `Wikidata` entities such as `Q42329` (wool), `Q232191` (sweater), and `Q552230` (Miu Miu) are extracted.

that is relevant for recycling is a clothing item that occurs only once in the user's favorite outfits, i.e., $\hat{r}_\tau(u,i) = 1$.

#### 4.2.1 *Evaluation protocol.*

We evaluate all the users in the dataset that has at least one item $i$ such that $\hat{r}_\tau(u,i) > 3$. Moreover, we set $\lambda = 2$ in Equation 3. We then assume that the user only owns items that occurs at least once in his favorite outfits, i.e., $\hat{r}_\tau(u,i) > 0$. For these items, we generate a recommended list to the users using Equation 4.

#### 4.2.2 *Evaluation metrics.*

To assess the quality of the recommendations we apply the traditional evaluation metrics Recall and Precision defined as follows:

$$\text{Recall} = \frac{tp}{tp + fn} \qquad \text{Precision} = \frac{tp}{tp + fp}, \tag{6}$$

where $tp$ is the number of correctly recommended relevant items, $fn$ is the number of wrongly recommended relevant items, and $fp$ is the number of wrongly recommended non-relevant items. We report Recall@N and Precision@N which is the Recall and Precision in a ranked list just considering the first N items.

#### 4.2.3 *Baseline method.*

As a baseline, we use the classic *Bag of Words* approach and represent the free text descriptions of the clothing items as TF vectors as opposed to CF in our proposed approach.

### 4.3 Results

We report the results in Figure 4. For the evaluation metrics, we focused on N = 5, since our system will display 5 recycling recommendations to the user. From the figure, we note that our proposed approach using the CF weighting scheme outperforms the baseline in both categories.

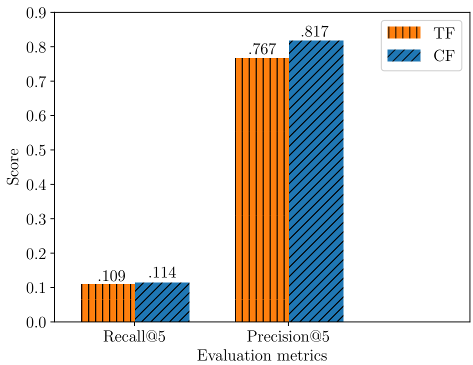



**Figure 4: Experiment results of Recall@5 and Precision@5 for the baseline (TF) and our proposed approach (CF).** The reported results are the average of Recall@5 and Precision@5 for each user evaluated in the experiment.

*CF vs. TF.* In the following, we highlight the advantages our proposed CF approach has over the TF baseline. These highlights should give an idea of why the CF outperforms TF. A drawback of *Bag of Words* is that it considers all words in the text descriptions as equally important. To use the item in Figure 3 as an example, it represents the item as an item containing the word 'miu' two times, when—in fact—'Miu Miu' is the brand of the item and is of vital importance to the representation of the item. For clothing items with brand names of common terms, such as 'Jean Shop', the *Bag of Words* approach is in deep trouble. In advantage, the *Bag of Concepts* approach is able to extract the 'Miu Miu' entity (`Q552230`) and able to capture the semantic context of the item. Moreover, the *Bag of Words* approach will describe the item using words such as 'clean', which does not characterize the item in any way.

## 5 RELATED WORK

In recent years, LOD in recommender systems has been frequently researched and various applications have been proposed [2, 3]. The most common application is to calculate semantic similarity of items based on the item's relationships found in datasets published as LOD. Using concepts from a LOD Knowledge Base to model the user profile have shown promising results in past work, e.g., [8]. Many of these past works use `DBpedia`[4] as a LOD Knowledge Base and focus on recommendation in the traditional domains where large datasets are available, such as movies, music, and books.

Our recommender system addresses a relatively unexplored domain and exploits a LOD Knowledge Base lacking previous research. To the best of our knowledge, recommending items that is no longer of interest to the user is a quite recent idea. Moreover, with the proposed approach built into the architecture of the Internet of Things wardrobe, this paper's proposed recommender system in the fashion domain is the first of its kind.

[4] http://wiki.dbpedia.org/

## 6 CONCLUSION AND FUTURE WORK

In this paper, we describe an Internet of Things wardrobe enabled with a proposed semantic content-based recommendation approach to recommend clothing items for recycling. We describe a set of context signals obtained from the wardrobe's architecture and how they affect the recommended list displayed to the user. Evaluation of our approach shows that our approach outperforms a baseline in terms of accuracy. Moreover, previous research has shown that LOD can increase recommender system's transparency and increase the user's trust in the system by computing convincing explanations to the recommendations [7]. The proposed approach facilitates opportunities for this and is planned for later research. Hence, the approach poses as a promising fit for the system.

Future work will also be devoted to improving the recommendation approach with a Concept Frequency - Inverse Document Frequency weighting scheme, and to develop further steps in system for the users to act on the recommendations.

## ACKNOWLEDGMENTS

The authors wish to thank Accenture Norway for making their implementation of their smart closet prototype available for this research project.

## REFERENCES


[1] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. Linked Data-The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)* 5, 3 (2009), 1–22.

[2] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. 2012. Linked Open Data to Support Content-based Recommender Systems. In *Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS '12)*. ACM, New York, NY, USA, 1–8.

[3] Cristhian Figueroa, Iacopo Vagliano, Oscar Rodríguez Rocha, and Maurizio Morisio. 2015. A Systematic Literature Review of Linked Data-based Recommender Systems. *Concurr. Comput. : Pract. Exper.* 27, 17 (Dec. 2015), 4659–4684.

[4] Ingun Grimstad Klepp and Kirsi Laitala. 2016. *Clothing consumption in Norway.* Technical Report. Consumption Research Norway (SIFO), In Norwegian. 124 pages.

[5] Anders Kolstad, Özlem Özgöbek, Jon Atle Gulla, and Simon Litlehamar. 2017. Connected Closet - A Semantically Enriched Mobile Recommender System for Smart Closets. In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*. 298–305.

[6] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2011. *Content-based Recommender Systems: State of the Art and Trends.* Springer US, Boston, MA, 73–105.

[7] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2016. ExpLOD: A Framework for Explaining Recommendations Based on the Linked Open Data Cloud. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 151–154.

[8] Guangyuan Piao and John G. Breslin. 2016. Exploring Dynamics and Semantics of User Interests for User Modeling on Twitter for Link Recommendations. In *Proceedings of the 12th International Conference on Semantic Systems (SEMANTiCS 2016)*. ACM, New York, NY, USA, 81–88.

[9] Chavan RB. 2014. Environmental Sustainability through Textile Recycling. *Journal of Textile Science & Engineering* (2014), 1–6.

[10] Magnus Sahlgren and Rickard Cöster. 2004. Using Bag-of-concepts to Improve the Performance of Support Vector Machines in Text Categorization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*. Association for Computational Linguistics, Stroudsburg, PA, USA, Article 487.

# Part III

# Summary and Outlook

# Chapter 7

# Discussion

While conducting the work on this thesis, a lot of important insight was obtained. Due to taking on a relatively new domain in recommender systems, acquiring suitable data (Section 7.1) and comparable research (Section 7.3) proved to be challenging. Moreover, due to the complexity of the system, a lot of technical issues (Section 7.2) were experienced. In addition, some insight was also obtained while conducting the experiments (Section 7.4) described in the papers.

## 7.1 Domain and Dataset

As always, acquiring a suitable dataset is challenging. For this specific project, the offline experiment required data about a set of clothing items that were put together in outfits and rated by users. Moreover, the user's usage history on the items should be included with information on which items that had been recycled or donated.

In a domain with already very little available data suitable for recommender systems evaluations, a complete dataset as described above turned out to be impossible to automatically extract. At the system's current stage of being an early prototype, the system was too immature to be used for retrieving data from a set of test subjects. Ultimately, some social media sites posed as good candidates for extracting some data that could be used in a dataset for offline experiments. This data did not cover usage history or recycling information but was extensive enough for complete accuracy evaluations in Paper II and with some preliminary accuracy evaluations in Paper III.

## 7.2 Technical Issues

The first IoT demonstrator developed as part of the prototype from the summer internship was not made available for this research project. Because of this, a new IoT demonstrator had to be built for this project. With little experience in hardware development, this turned out to be quite of a technical challenge. First, the acquisition of the same type of RFID reader used in the first demonstrator, was not possible. This resulted in that the source code for the computer embedded in the closet was useless for the new demonstrator—requiring that the IoT demonstrator had to be built from scratch. Secondly, with little available documentation on connecting the acquired RFID reader to a Raspberry PI, this whole process became more time consuming than initially expected.

The source code developed during the internship was made available for this research project. The codebase provided basic functionality, such as inventory overview. In order to extend the existing architecture with recommender and semantic web technology, a lot of changes in the original codebase had to be made. This included fixing bugs originating from the internship (a lot of the bugs were to be blamed on the thesis author). Moreover, the codebase's many frameworks, libraries, and tools had been deprecated and the process of migrating to newer versions were time consuming and posed technical difficulties.

However, due to the system's use of new and innovating technology (especially the microservices), further extension of the architecture with recommender and semantic web technology, was a quite painless process. The collection of available machine learning libraries, natural language processing tools, and knowledge bases available, has become quite extensive in recent years. All providing good documentation and allows for simple integration methods. This aided the system in using state-of-the-art algorithms and techniques.

## 7.3 Related Research

The low number of related previous studies also contributed to the challenge of data collection. Moreover, the previous studies lacked extensive descriptions of how to recommend clothes from smart closets and was often not developed or evaluated in any way. This proved it difficult to compare the results of this work to similar works in the literature. On the other hand, this also proves that the work of this thesis is unique and will hopefully encourage future research and innovation of similar systems.

## 7.4 Experiments

While conducting the experiment reported in Paper II, it was expected that the AdaBoost algorithm and the Gradient Boosting algorithm would outperform the much simpler Naïve Bayes algorithm. Ultimately, AdaBoost was selected for the system because it yielded the best result on the collected dataset. However, Gradient Boosting's performance was pretty close to AdaBoost, and it is possible that Gradient Boosting might perform better on a different dataset. Moreover, from the ROC curves, it shown from the Naïve Bayes algorithm's more choppy line, that it is generating much fewer recommendations than AdaBoost and Gradient Boosting. Due to time constraints and Naïve Bayes' much weaker performance, the reason for this was left unexplored.

An interesting fact in the same experiment is that closet size did not affect the accuracy of the recommendations. However, the only other closet size than 'full' that was tested was a 50% reduction of all clothing items. In hindsight, it is possible that the accuracy can be affected by different distributions of clothing types, for example, twice as many tops as bottoms.

The experiment performed in Paper III showed that the proposed Bag of Concepts approach outperformed the baseline using Bag of Words. The improvement was expected to be much larger, but with the proposed approach's use of semantic web technology, it is still considered as a promising fit for the system. It should also be mentioned that both approaches were pretty naïve implementations, and with more tuning and optimization, the improvement could possibly be greater.

# Chapter 8

# Conclusion

To conclude the thesis, this final chapter summarizes the contributions (Section 8.1) of the thesis and gives concluding remarks to the research questions and the goals (Section 8.2). Moreover, a discussion of future work (Section 8.3) is given.

## 8.1 Summary of Contributions

Chapter 1 stated the main contributions of this thesis. This section sheds light on how each of these contributions are provided in the papers.

**C1** *The architecture and design of a smart closet.*

Paper I describes the whole architecture of the system. It gives detailed descriptions of the task for each of the components in the architecture. Figure 1, Figure 3, and Figure 5 are the most important figures in the paper that summarize this contribution.

**C2** *A novel collaborative filtering approach for daily outfit recommendations.*

The novel collaborative filtering approach is best summarized by Figure 4 and Figure 5 in Paper II. The novelty of the approach is the transformation of the utility matrix into an outfit-item matrix.

**C3** *A novel semantic content-based approach for garment recycling recommendations.*

The novel semantic content-based approach is best summarized by Figure 2 and Figure 3 in Paper III. The novelty lies in recommendation of items that are no longer of interest to the user.

**C4** *Thorough evaluations of the approaches using a real-world dataset.*

A thorough evaluation of the novel collaborative filtering approach is given in Paper II – Section 7. The semantic content-based approach is evaluated in Paper III – Section 4.

## 8.2 Concluding Remarks

To summarize the key findings of this thesis, this section gives answers to the research questions. The answers are concise and meant to emphasize the extensive descriptions from the papers. Moreover, for each goal, a brief summary is given, meant as a measure of how the goals were fulfilled.

### 8.2.1 Research Questions

**RQ1** *What are the core components and the functionality of a smart closet and how can this be supported by recommendation technology?*

Paper I describes the system of a smart closet where clothing items enabled with RFID tags can be manually scanned in and out of the closet using a tiny computer embedded in the user's physical closet. In a mobile application, the user is displayed the closet inventory. Moreover, the mobile application provides daily outfit recommendations using an approach that is based on collaborative filtering leveraging machine learning algorithms. Paper III describes how the mobile application can receive recycling recommendations by a content-based recommender system leveraging Linked Open Data. A detailed architectural view is available in Appendix A.

**RQ2** *How can we automatically extract datasets that help us build and evaluate content-based recommendation and collaborative filtering for smart closets?*

Paper II and Paper III describes a social media site called Polyvore that mirrors some of the same functionality as the smart closet's mobile application. Paper II describes how this site can be scraped in order to extract a dataset of fashion outfits that is rated by a set of users.

Figure 8.1 illustrates an excerpt of some items in the collected dataset. Paper III includes a figure describing how each of these items are semantically enriched for the content-based approach.

**RQ3** *To what extent (accuracy) can a recommender system help users choose clothes from a smart closet?*

Paper II and Paper III conducts experiments on the collected dataset with accuracy evaluation using traditional evaluation metrics from information retrieval. The results (Paper II – Section 7.4) from Paper II supported the selection of the best performing machine learning algorithm that recommended outfits most accurately. Moreover, it shows how a conventional approach is inferior to the novel approach. Paper III showed that a content-based recommender system leveraging Linked Open Data for recycling recommendations, outperformed a baseline without Linked Open Data, in terms of accuracy.

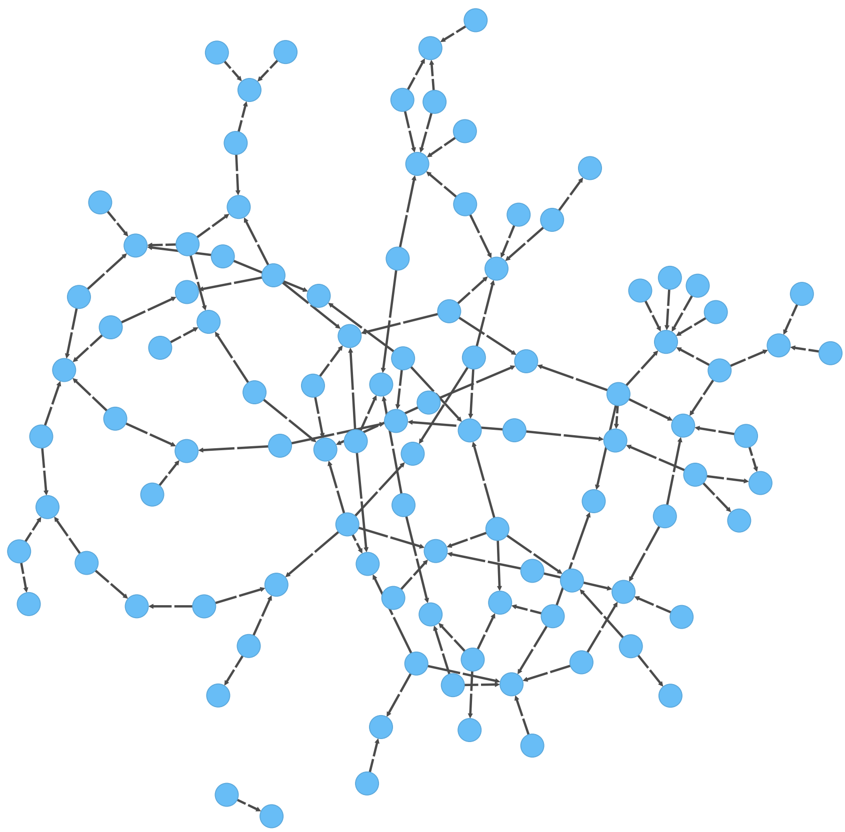


Figure 8.1: **Excerpt of the dataset as stored in the system's graph database.** *The nodes represent the clothing items while the edge between two nodes represent the outfit of the two nodes. The edge stores a weight corresponding to the number of outfit likes from the users.*

### 8.2.2 Goals

**G1** *The 'New IT' strategy*

Paper I described the system's architecture and the set of microservices implemented using the most suitable platform for their designated tasks. Improving the modularity of the system. The microservices and databases were implemented using container-based virtualization with Docker. This enabled automation of testing, building, and deployment—ensuring that the system's software is able to go into production at continuous delivery.

The paper showed how a tiny computer could be integrated with the architecture, using the lightweight communication protocol MQTT, and how the Fog Computing architecture handles the IoT device's substantial amount of data created by the device's sensors.

Paper II and Paper III showed how to build 'intelligent applications' using recommender systems, machine learning, and semantic web.

**G2** *IoT demonstrator*

Figure 2 in Paper I depicts the IoT demonstrator built as a small open closet. It was built using a Raspberry PI connected to an RFID reader. Some pictures of clothing items were printed and attached to RFID tags. With the system's mobile application, this provided an apt little demonstrator of the whole system that was easy to set up and to transport. Paper II included a short video (https://goo.gl/rZBZqo) of the IoT demonstrator in action.

**G3** *Show off at conferences*

During the work on this thesis, the IoT demonstrator was showcased at itDAGENE[1] and JavaZone Academy[2]. itDAGENE is an annual career fair where businesses get to meet IT students at NTNU. The IoT demonstrator was displayed at Accenture's booth at the event venue. JavaZone Academy is a miniature version of Scandinavia's largest IT conference JavaZone[3]. JavaZone Academy brings the best from JavaZone and present it to students. Accenture participated with a booth in the conference's exhibition area where the IoT demonstrator was displayed.

Moreover, with a paper acceptance in the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017), the paper authors were awarded a presentation slot at the conference where the work was presented as an oral presentation.

[1] https://www.itdagene.no/
[2] https://2017.javazone.no/academy
[3] https://www.javazone.no

**G4** *Publicity*

Together with the thesis author and some Accenture representatives, the project was promoted with the IoT demonstrator on display at the conferences and events mentioned above. Moreover, Paper I was published in the proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017). In addition, Paper II and Paper III are awaiting author's notifications in high-profile conferences.

## 8.3 Future Work

The system described in this thesis should be considered as an early prototype and a lot of development and research remains before a possible full scale production of smart closets. The future work that is directly related to the scope of this thesis, is the improvement and the additional applications regarding the daily outfit recommendations and the recycling recommendations.

### 8.3.1 Daily Outfit Recommendations

#### Granularity

Since outfits can also comprise of more clothing items than what is researched in this thesis (one top and one bottom), further research should be devoted to how to add additional accessories to the recommended outfits computed using the outfit-item matrix approach proposed in Paper II.

#### Occassion-based recommendations

Different occasions require different dress codes, e.g., business meeting vs. dinner party. This could be addressed by integrating the user's calendar with the system, and the recommendations could be achieved, for example, by using an inclusion criteria similar to the one proposed in Paper I and Paper II.

#### Retail recommendations

The number one benefiting factor a fashion retailer could achieve from a system as described in this thesis, is targeted advertisement. Paper II briefly mentions that the outfit-item matrix approach can be used to generate targeted advertisement of clothing items that match clothing items already owned by the users.

### 8.3.2 Recycling Recommendations

#### Feedback

In the current version of the system, recycling recommendations are just displayed to the user. In case of recycling, the next user steps, such as removal from the user's inventory and registration at a recycling checkpoint, needs to be developed. Moreover, if the user does not want to recycle the recommended item, the user should be able to give the system feedback on this, so that the item will not be recommended again.

#### Explanations

Utilization of Linked Open Data in the recycling recommendations facilitates for a computational model for generating convincing explanations to the recommendations. Explanations to the recycling recommendations should be considered as a highly important feature. For a user to recycle a clothing item, it is safe to assume that the user requires a pretty sound justification for the recommendation.

#### Additional incentives

A recommendation for recycling with a good justification for the recommendation, provides some incentives for people to recycle their clothes. This could be improved with applications such as gamification [68]. A measure of the user's carbon footprint [69] or some awarding of discounts at clothing retailers, are some things that could strengthen the incentives for recycling.

# Bibliography


[1] Martin Braaten Røise. Accenture har stor tro på prosjektet de unge studentene utviklet på bare seks uker. https://www.digi.no/artikler/da-de-bestemte-seg-for-utvikle-i-typescript-dukket-det-opp-problemer-de-ikke-hadde-sett-for-seg/351187, September 2017. In Norwegian. [Online; posted 6-September-2016].

[2] John Gantz and David Reinsel. Extracting Value from Chaos. *IDC iView*, 1142(2011):1–12, 2011.

[3] Kevin Ashton. That 'Internet of Things' Thing. *RFiD Journal*, 22(7):97–114, 2009.

[4] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805, 2010.

[5] Zeinab Tavakoli and Taghi Javdani Gandomani. Big Data and Its Challenges. *Majlesi Journal of Multimedia Processing*, 4(1), 2015.

[6] Ingun Grimstad Klepp and Kirsi Laitala. Clothing consumption in Norway. Technical report, Oslo and Akershus University College of Applied Sciences - SIFO, 2016. In Norwegian.

[7] John C Pruit. Getting Dressed. *Popular Culture as Everyday Life*, page 155, 2015.

[8] RB Chavan. Environmental Sustainability through Textile Recycling. *Journal of Textile Science & Engineering*, (S2):1, 2014.

[9] Chris M Roberts. Radio Frequency Identification (RFID). *Computers & Security*, 25(1):18–26, 2006.

[10] Accenture PLC. https://www.accenture.com/. Accessed: 2017-05-24.

[11] SmartMedia Program. https://research.idi.ntnu.no/SmartMedia/. Accessed: 2017-05-24.

[12] Accenture Technology. The Future of Applications. http://www.accenture.com/applicationstrategies, 2015.

[13] Dmitry Namiot and Manfred Sneps-Sneppe. On Micro-Services Architecture. *International Journal of Open Information Technologies*, 2(9), 2014.

[14] A. Balalaie, A. Heydarnoori, and P. Jamshidi. Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. *IEEE Software*, 33(3):42–52, May 2016.

[15] J. Rufino, M. Alam, J. Ferreira, A. Rehman, and K. F. Tsang. Orchestration of containerized microservices for IIoT using Docker. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 1532–1536, March 2017.

[16] Harry Stockman. Communication by Means of Reflected Power. In *Proceedings of the IRE*, volume 36, pages 1196–1204. IEEE, 1948.

[17] MQ Telemetry Transport. http://mqtt.org/. Accessed: 2017-05-24.

[18] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, pages 13–16, New York, NY, USA, 2012. ACM.

[19] Charu C. Aggarwal. *Recommender Systems: The Textbook.* Springer Publishing Company, Incorporated, 1st edition, 2016.

[20] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets.* Cambridge University Press, New York, NY, USA, 2011.

[21] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction.* Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[22] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[23] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[24] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.

[25] Steffen Staab and Rudi Studer. *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

[26] Ora Lassila, Ralph R. Swick, World Wide, and Web Consortium. Resource Description Framework (RDF) Model and Syntax Specification. https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, 1998.

[27] Tim Berners-Lee. Linked Data - Design Issues. http://www.w3.org/DesignIssues/LinkedData.html, 2006.

[28] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data-The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[29] Wikidata. https://www.wikidata.org/. Accessed: 2017-05-24.

[30] Wikimedia Foundation. https://wikimediafoundation.org/. Accessed: 2017-05-24.

[31] Lisa F. Rau. Extracting Company Names from Text. In *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, volume i, pages 29–32, Feb 1991.

[32] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[33] Magnus Sahlgren and Rickard Cöster. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proceedings of the 20th international conference on Computational Linguistics*, page 487. Association for Computational Linguistics, 2004.

[34] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Semantic enrichment of twitter posts for user profile construction on the social web. In *Extended Semantic Web Conference*, pages 375–389. Springer, 2011.

[35] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[36] David D. Lewis. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 4–15, London, UK, UK, 1998. Springer-Verlag.

[37] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[38] Yoav Freund and Robert E. Schapire. Experiments with a New Boosting Algorithm. In *Proceedings of the 13th Internatinal Conference On Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[39] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001.

[40] Richard Bellman. Adaptive control processes: A guided tour. (A RAND Corporation Research Study). Princeton, N. J.: Princeton University Press, XVI, 255 p. (1961)., 1961.

[41] Ian Jolliffe. *Principal component analysis*. Springer Verlag, New York, 2002.

[42] P. D. Francesco, I. Malavolta, and P. Lago. Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 21–30, April 2017.

[43] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000. AAI9980887.

[44] Node.js. https://nodejs.org/. Accessed: 2017-05-24.

[45] Express. https://expressjs.com/. Accessed: 2017-05-24.

[46] Gavin Bierman, Martín Abadi, and Mads Torgersen. *Understanding TypeScript*, pages 257–281. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[47] Flask. http://flask.pocoo.org/. Accessed: 2017-05-24.

[48] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[49] TextRazor. https://www.textrazor.com/. Accessed: 2017-05-24.

[50] The Weather Company. http://www.theweathercompany.com/. Accessed: 2017-05-24.

[51] Alex Russel. Progressive Web Apps: Escaping Tabs Without Losing Our Soul. https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/, 2015.

[52] Andreas Biørn-Hansen, Tim A. Majchrzak, and Tor-Morten Grønli. Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 344–351. INSTICC, ScitePress, 2017.

[53] Angular. `https://angular.io/`. Accessed: 2017-05-24.

[54] Neal Leavitt. Will NoSQL databases live up to their promise? *Computer*, 43(2), 2010.

[55] MongoDB. `https://www.mongodb.com/`. Accessed: 2017-05-24.

[56] Neo4j. `https://neo4j.com/`. Accessed: 2017-05-24.

[57] InfluxDB. `https://www.influxdata.com/`. Accessed: 2017-05-24.

[58] M. Shahin, M. Ali Babar, and L. Zhu. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5:3909–3943, 2017.

[59] Docker. `https://www.docker.com/`. Accessed: 2017-05-24.

[60] Kim Nee Goh, Yoke Yie Chen, and Elvina Syn Lin. Developing a Smart Wardrobe System. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 303–307. IEEE, 2011.

[61] Lin Yu-Chu, Yuusuke Kawakita, Etsuko Suzuki, and Haruhisa Ichikawa. Personalized Clothing-Recommendation System Based on a Modified Bayesian Network. In *Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on*, pages 414–417. IEEE, 2012.

[62] Aaron P Toney, Bruce H Thomas, and Wynand Marais. Managing Smart Garments. In *Wearable Computers, 2006 10th IEEE International Symposium on*, pages 91–94. IEEE, 2006.

[63] Sea Ling, Maria Indrawan, and Seng W Loke. RFID-Based User Profiling of Fashion Preferences: Blueprint for a Smart Wardrobe. *International Journal of Internet Protocol Technology*, 2(3-4):153–164, 2007.

[64] Si Liu, Jiashi Feng, Zheng Song, Tianzhu Zhang, Hanqing Lu, Changsheng Xu, and Shuicheng Yan. Hi, Magic Closet, Tell Me What to Wear! In *Proceedings of the 20th ACM international conference on Multimedia*, pages 619–628. ACM, 2012.

[65] Bojana Dumeljic, Martha Larson, and Alessandro Bozzon. Moody Closet: Exploring Intriguing New Views on Wardrobe Recommendation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, pages 61–62. ACM, 2014.

[66] Chantima Limaksornkul, Duangkamol Na Nakorn, Onidta Rakmanee, and Wantanee Viriyasitavat. Smart Closet: Statistical-based Apparel Recommendation System. In *Student Project Conference (ICT-ISPC), 2014 Third ICT International*, pages 155–158. IEEE, 2014.

[67] Lei Xie, Yafeng Yin, Xiang Lu, Bo Sheng, and Sanglu Lu. iFridge: An Intelligent Fridge for Food Management Based on RFID Technology. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 291–294. ACM, 2013.

[68] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From Game Design Elements To Gamefulness: Defining Gamification. In *Proceedings of the 15t International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15. ACM, 2011.

[69] Thomas Wiedmann and Jan Minx. A Definition of 'Carbon Footprint'. *Ecological Economics Research Trends*, 1:1–11, 2008.

# Appendix

# Appendix A. System Architecture