NTNU
Norwegian University of Science and Technology

Recommending activities for refugees in Norway

Henriette Vanvik Kopstad

Master of Science in Computer Science
Submission date: June 2017
Supervisor: Herindrasana Ramampiaro, IDI
Co-supervisor: Katrien De Moor, IIK

Norwegian University of Science and Technology
Department of Computer Science

TDT4900 - Datateknologi, Masteroppgave

NTNU

Spring 2017

Recommending activities to refugees

Author:
Henriette Kopstad

Supervisors:
Heri Ramampiaro
Katrien De Moor

Abstract

The influx of asylum seekers and refugees, especially in recent times, has caused a bottleneck in Utlendingsdirektoratet (UDI)'s application process, leading to long waiting periods for their work and residency permits. During this time refugees have difficulties finding pastime activities, and instead of having possibilities of integration and inclusion, they suffer from boredom, passivization and reduced quality of life. This paper researches possibilities for using techniques from recommender systems to aid in solving this problem of finding activities and integrating refugees into Norwegian society. After researching state of the art tools and techniques and current existing systems pertaining to similar areas, a definite trade-off between simplicity of the system, and the accuracy and effectiveness of its recommendations was observed in existing solutions. This resulted in a content-based recommender system, Refugenious, being developed and tested, where the idea is to recommend activities without requiring vast amounts of user data. The initial results of its evaluation are promising and show that such a recommender system can provide good recommendations with only a small amount of information. The recommender system also proves to aid with lowering the threshold for finding activities, increase participation and widen refugees' specter of interests. Refugenious is planned to be deployed on the web portal InterConnect within the end of 2017.

Contents

1 Introduction

This chapter gives an introduction into the basic premises this thesis is built upon. In section 1.1, the motivation for the research and work is presented. The following sections define the research questions and goals of the thesis, as well as the scope of the work and structure of the paper itself.

1.1 Motivation

According to the UN Refugee Agency there are an estimated 21.3 million refugees[1] in the world today, and 31150 have asked for protection in Norway alone[2]. Due to their sheer volume and the long processing time for applications, refugees that arrive in Norway are in for a long waiting period before being accepted into the country. While they wait for their work and residence permit, they are restricted from having a job and many have a hard time finding other activities to fill their days with[3].

Using the internet to find information has become commonplace for most people, and it is a natural source for refugees to seek information as well. There are several web pages by organizations such as Røde Kors and Flyktninghjelpen, that offer activities and events refugees *are* eligible to participate in, but the information is spread on numerous web sites, often with user interfaces not made with refugees in mind as the target audience.

Recommender systems have since the 90's offered a way to aid people in finding the right items and information in the vast sea of information online. Because there is a need among refugees to gather information in one place and filter out what is relevant and what is not, an idea of creating an online web portal that deals with these needs was developed. The portal, which is currently being developed under the working title InterConnect, aims to provide an intuitive and simple to use system that works as a gathering point for all activities the refugees are eligible for, regardless of their current status. An integrated back-end recommender system will sort through the available information, and return a ranked set of activities that is tailored for the individual user. It is the development and evaluation of this back-end system, Refugenious, that is the focus of this project thesis.

Having a portal that gathers this information and focuses on offering the items that are most relevant to the individual refugee in a simple and intuitive way could be a great resource for an otherwise deferred group of people. It could lower the threshold to participate and use the web portal, making more refugees take advantage of the services. This would not only provide opportunities for fast and free integration into their new communities and learning the language; it could also help the refugees fill their time with *meaningful* activities where they obtain valuable experience. This experience could help them start their new life once their application is approved.

1.2 Objective and research questions

Based on the information from the previous section, it is clear that it is ideal for refugees to have a single point of access to information pertaining to their situation, such as activities like courses, volunteer work, and social events they can attend. Organizations providing such activities should have access to and be able to post events in a single portal, and the threshold for refugees finding activities relevant for themselves should be as low as possible. A theory this is based on is that a web-based solution is suited to function as such a gathering point for information, and a recommender system could aid in simplifying the activity-finding process. Investigating a way to develop a suitable recommender system, the main research question to answer is:

> How can recommender system technologies and techniques be used as a tool for refugee integration?

[1] http://www.unhcr.org/figures-at-a-glance.html

[2] https://www.udi.no/statistikk-og-analyse/statistikk/asylsoknader-etter-statsborgerskap-aldersgruppe-og-kjonn/

[3] Information gathered from recent results of Tankesmien Agenda: http://www.tankesmienagenda.no/wp-content/uploads/Perspektivnotat-Ti-bud-for-bedre-integrering-1.pdf and through qualitative studies by my associate Ida Marie Støp Meland in Bergen and Trondheim

This question poses several sub-questions for my research as well:

RQ1: What is the current situation pertaining to aiding refugees with finding activities?

RQ2: How can a recommender system accurately find activities suitable for the individual refugee?

RQ3: What information is needed for modeling accurate user and item profiles to use in a recommender system?

RQ4 Can enough information be gathered to make accurate predictions without compromising the arguably mandatory low threshold for using the system?

1.3 Scope and limitations

The scope of this project is to do extensive research on tools and techniques of recommender systems to implement a suitable recommender system for the InterConnect web portal. In addition, the system should be tested both when it comes to utility and feasibility for real world application, and the quality of its recommendations. The report will work as an overview of the research process, as well as an introduction into the development and testing of the final system Refugenious.

Although the main focus of the thesis will be on Refugenious, it will include aspects of InterConnect as well, as the two systems are unavoidably linked and the requirements of each affect the other. InterConnect will also be the component responsible for presenting the recommendations to the user, and will therefore be included in the evaluation strategy as well.

In order for it to be realistic to do research, gather data and test the system, Refugenious' intended user group will be limited to refugees in Norway. However, maintaining the system's ability to scale internationally will be considered in the decision making processes.

1.4 Structure of report

Chapter 2 gives an overview of the theory and background of recommender systems, explaining the most prominent techniques and approaches. Chapter 3 delves into state of the art research within the area of recommender systems, their proposed systems and solutions, as well as how they can be related to this project. In chapter 4, an overview of the implemented system is shown, in addition to its architecture and relationship with the web portal InterConnect. Chapter 5 deals with the evaluation of the system, both technical and non-technical, before the entire research process and focus is evaluated in Chapter 6. Chapter 7 gives an overview of the conclusions drawn from the research and evaluations, as well as future work.

2 Background

Recommender systems are software tools that belong to a subgroup of information filtering technology, whose goal is to present their users with easily accessible, high-quality predictions of items they are likely to be interested in [1]. Prevalent web services today, such as Netflix and Spotify, use recommender systems to suggest movies and music for their users respectively, a service that adds great value to their product.

Recommender systems rose as an independent research area in the mid 1990's and have since had a dramatic increase in interest as online services and electronic commerce have emerged and grown. Today, recommender systems have become the most powerful and popular tools of e-commerce, where the total turnover exceeds a trillion dollars, making advances within item suggestions a lucrative business.

The systems can base their predictions on different pieces of information, alone or in combination, and typically process this information in one of two ways; with content-based or collaborative filtering (CF). Less common methods such as demographic and knowledge based systems have also been used, and an emerging trend in many situations is to choose a hybrid system, which incorporates techniques of two or more of the aforementioned approaches.

2.1 Content-based systems

In content-based filtering, the predictions are made primarily using information about the items themselves. The system provides item suggestions to the user similar to the ones he or she has liked in the past, or that match the user's profile by some other metrics, such as predefined preferences.

> *Example*: Consider someone who is looking for a job and uses an online marketplace for job listings. He might specify beforehand that he is looking for a job within web development, and a recommender system should then be able to recommend jobs listings that match this category. However, because finding a job can take time, the user might also show an interest in smaller jobs to earn some money in the meantime, such as painting fences etc. Although this is not user specified, a recommender system should be able to register this as well, and suggest similar jobs to the user as they are posted.

To determine the likeness of items, many strategies have been researched, but the most widely used is the cosine similarity measure. Here, the items are represented as a uniform set of attributes, such as genre, actors and director for movie items. A user profile is then created, represented with the same attributes and built up by analyzing the content of items which have been deemed interesting by the user. New items are considered relevant or non-relevant by how they suit the user profile by finding the similarity of their two m-dimensional vectors using the formula:

$$sim(\vec{Item1}, \vec{Item2}) = \frac{\vec{Item1} \cdot \vec{Item2}}{||\vec{Item1}|| \cdot ||\vec{Item2}||} = \frac{\sum_{n=1}^{m} Item1_n Item2_n}{\sqrt{\sum_{n=1}^{m} Item1_n^2}\sqrt{\sum_{n=1}^{m} Item2_n^2}} \tag{1}$$

> *Example:* Let us again consider the man in search for a job within IT. In a content-based recommender system the user profile is built by registering information about the jobs he shows interest in, which might be jobs posted by an IT-company, or jobs that have "Web Developer" in their title. Any new job listing that is posted would then be checked for similar attributes as the ones in his profile, which in turn would result in high probability of interest if its title includes "Web Developer" or it is posted by an IT-company.

There are several advantages to this approach to recommender systems. The system is fairly user independent, as it does not require users to explicitly rate items; the system can infer it itself. In addition, whenever a new item is introduced into the system, it can be recommended to users at once, by comparing it with items they have already deemed relevant or not.

However, there are some significant drawbacks to the approach as well. The number of attributes the items can be represented with must be limited, and do not necessarily provide a complete picture of what information is needed to deem the item as relevant or not. The system can also result in overspecialization in the ranked list of recommendations given to the user. Items that might be of interest can be ignored because they do not resemble items already liked by the user. In addition, even though new items are handled well in content-based systems, new users are not. The system requires the user to provide enough items marked as relevant before the user profile can be built and suggestions for new items can be made.

2.2 Collaborative filtering systems

Contrary to content-based, recommender systems that utilize collaborative filtering do not need information about the items themselves, and are rather user dependent. The idea is to exploit information about the rating behavior and opinions of a user community for suggesting interesting items to the individual user. A central assumption in collaborative filtering is that users that have shared interests in the past will have similar tastes in the future. Collaborative filtering is considered the most popular and widely used technique in recommender systems.

There are two prominent approaches to CF; user-based and item-based collaborative filtering. In user-based systems, the rating vectors of each user are compared. If two users' rating vectors are similar, the users themselves are deemed similar by the system, and items one user has rated high is recommended to the other, and vice versa. The similarity of taste between two users is therefore calculated based on the similarity of their rating histories.

> *Example*: Using the job seeker as an example again, he might have viewed and marked several job listings as relevant. Another user looking for a job similar to him might have viewed many of the same listings and marked them as interesting as well. The system does not need to know the content of these listings, but assumes the two users are looking for the same type of items. Therefore, any item the first user has ranked high that is not yet seen by the second, will likely be suggested to the latter, and vice versa.

Oftentimes, the amount of items supersede the number of user ratings, making user rating vectors sparse. One way to better this problem, is through item-based CF. Here, it is the rating vectors of each *item* that are compared to each other, as items often have more ratings than a user has given ratings. The system looks at the ratings of each item to determine its similarity to other items, and recommends items to the user that have similar rating patterns to the ones he/she has already rated.

> *Example*: Let us say the job seeker rates Job A as relevant. The system will then recommend jobs that have similar rating vectors, essentially saying other users who tend to rate Job A high, also tend to rate the recommended jobs favorably as well. A well know example is Amazon's recommendations that say "People who have bought this item also tend to buy...".

As with content-based, there are several similarity measures available to compute the similarities of user tastes and item ratings, including cosine similarity. The arguably most prominent method that has shown to be statistically better than cosine in CF is the Pearson correlation coefficient.

$$Pearson(u,v) = \frac{\sum\limits_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum\limits_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum\limits_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{2}$$

This formula shows the correlation between user u and user v over items i, where I_{uv} denotes the items both u and v have rated. r_{ui} and $\bar{r}_u$ express user u's rating of item i, and user u's mean rating respectfully. For item-based CF, the equation for computing the similarity between item i and item j becomes:

$$Pearson(i, j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{\mathrm{r}}_i)(r_{uj} - \bar{\mathrm{r}}_j)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{\mathrm{r}}_u)^2 \sum_{u \in U_{ij}} (r_{ui} - \bar{\mathrm{r}}_v)^2}} \quad (3)$$

As mentioned before, collaborative filtering systems do not need to store or process information about users or items in order to make its predictions. Another advantage is that it can produce highly personalized recommendations as its suggestions are based on users' personal experiences. Contrary to content-based, suggesting serendipitous items is possible as it considers users' like-mindedness in rating behavior instead of focusing on the similarity of the rated items.

The primary disadvantage to collaborative filtering systems is know as the Cold-Start problem, related to the data sparsity that often occurs, especially of newly deployed systems. The collaborative-filtering technique is dependent on a significant amount of ratings to be available to it in order for it to make its suggestions, which both new and established systems can lack. The number of items typically far outnumber the amount of users, making the data sparse even with many ratings.

Also, the system lacks support for heterophilous diffusion, meaning that users might seek the recommendations of people more advanced than themselves, but as the system would likely identify them as dissimilar to the user, their superior opinions are disregarded.

Another problem is that of scalability. As the system grows and involves thousands or even millions of users and items, the computational costs become high. Simply comparing each user to every other user in the system to find those that are similar does not scale well.

2.3 Other methods

A *demographic* recommender system provides recommendations based on a demographic profile of the user [2]. It relies on information about users' gender, age, occupation, geographical location and more. Based on this information, users are clustered together into different demographic niches, and recommendations are given within each niche based on its users' ratings and interests.

When a recommender system is *knowledge-based* it suggests items based on inferences about a user's needs and preferences [3]. The system uses knowledge about items' features and infer how they meet the user's requirements. For example, a user that buys a new computer might also be interested in a computer bag. Its recommendations do not depend on a base of user ratings, nor does it have to gather information about a particular user as its judgments are independent of individual tastes.

Hybrid recommender systems are based on a combination of two or more of the aforementioned approaches. Hybrid systems typically incorporate techniques from one approach to fix the disadvantages of another, and vice versa. An example would be a system that uses collaborative filtering for item prediction and recommendation, but utilizes content based techniques for new items and users to avoid the cold start problem.

2.4 Evaluating a recommender system

To measure the quality of recommendations a system makes, it is common to use data sets that contain ground truth [1], or a blueprint of what items within the set are actually right to recommend or not. There are many measures comparing the actual interesting items with the ones the recommender system returns in order to infer something about its quality. For this thesis, the focus will be on three such measures; precision, recall and Normalized Discounted Cumulative Gain.

Precision describes the fraction of relevant items the system recommends to the user. It is calculated using the formula:

$$P = \frac{tp}{tp + fp}$$

where *tp* is the number of recommended items that are deemed relevant by the test user, while *fp* denotes the amount of recommended items that the test user marked as irrelevant.

Recall describes the fraction of relevant activities that were recommended to the user. It is computed as follows:

$$R = \frac{tp}{tp + fn}$$

where *fn* denotes the activities the test user found relevant but were not recommended by the system.

> *Example*: Given 10 activities, a recommender system returns the five recommendations with the highest score, while the remaining 5 are considered irrelevant. The test user rates all documents with a value between 0 and 2 as previously stated, where 1 and 2 are denoted with R, and the rating position given to the activity is denoted by i:

Activity	i	$Rank_{user}$	
A_1	1	1 R	
A_2	2	1 R	
A_3	3	2 R	
A_4	4	0 NR	
A_5	5	1 R	only top 5
A_6	6	0 NR	returned
A_7	7	0 NR	
A_8	8	2 R	
A_9	9	2 R	
A_{10}	10	1 R	

Table 1: Rating of activities by the system (i) and a test user ($Rank_{user}$)

> Out of the five activities that were recommended to the user, only one was not relevant. However, there were several relevant activities that were not included by the system. The precision and recall scores are computed as follows:

$$P = \frac{4}{4+1} = \frac{4}{5} = 0.8$$

$$R = \frac{4}{4+3} = \frac{4}{7} = 0.57$$

> Precision is high, but the recall score indicates that a little under half the total amount of relevant activities were recommended to the user. this could indicate that increasing k, here at 5, could be considered.

Normalized Discounted Cumulative Gain is an evaluation technique often used for web search engines [4]. The premise behind the algorithm as that the *order* of the recommended activities is crucial, as a relevant item that has a low position is less likely to be found and investigated. Higher values of NDCG indicate better ranked activities and will therefore be used to evaluate the ranking algorithm's correctness. How NDCG is calculated is best shown though the computation of its denormalized version, DCG:

$$DCG = \sum_{i=1}^{m} \frac{rel_i}{log_2(i+1)}$$

where rel_i is the relevance score given by the test user to the activity ranked i^{th} by the system, m is the total number of rated items, and the logarithmic function punishes wrongly positioned items. Normalized DCG results in a value between 0 an 1 and is defined by:

$$NDCG = \frac{DCG}{iDCG}$$

where iDCG denotes the ideal score that could be obtained with the same activities, meaning the DCG score they would obtain when sorted in a descending order. To further illustrate how the algorithm works, we will examine the same example of activities and rankings as before:

Example: The relevance classification of the previous example is no longer important, only proper positioning, that is to say *ranking.* The following table shows the calculation of the components of DCG. Note that $Rank_{user}$ is equivalent to rel_i and has thus been replaced in the table.

Activity	i	rel_i	$log_2(i+1)$	$\frac{rel_i}{log_2(i+1)}$
A_1	1	1	1	1
A_2	2	1	1.585	0.631
A_3	3	2	2	1
A_4	4	0	2.322	0
A_5	5	1	2.585	0.387
A_6	6	0	2.807	0
A_7	7	0	3	0
A_8	8	2	3.17	0.631
A_9	9	2	3.322	0.602
A_{10}	10	1	3.459	0.289

Table 2: Computing the components of DCG

With the components of DCG calculated, DCG itself is computed by summing the values of $\frac{rel_i}{log_2(i+1)}$:

$$DCG = 1 + 0.631 + 1 + 0.387 + 0.631 + 0.602 + 0.289 = 4.54$$

As previously stated, in order to normalize DCG and provide the final NDCG score, iDCG must be calculated. It is the DCG score the system would have received, had it ranked the activities the same as the user. The following table shows the activities and rankings rearranged.

Activity	i	rel_i	$log_2(i+1)$	$\frac{rel_i}{log_2(i+1)}$
A_3	1	2	1	2
A_8	2	2	1.585	1.261
A_9	3	2	2	1
A_1	4	1	2.322	0.431
A_2	5	1	2.585	0.387
A_5	6	1	2.807	0.356
A_{10}	7	1	3	0.333
A_4	8	0	3.17	0
A_6	9	0	3.322	0
A_7	10	0	3.459	0

Table 3: Computing the components of iDCG

Adding together the values gives an iDCG = 5.768, which results in the following expression for NDCG:

$$NDCG = \frac{4.54}{5.768} = 0.787$$

The example above was provided more for illustrating the workings of NDCG, but is limited in its data to show the real effect of the algorithm. A score of 0.787 may be a decent score in itself, but considering $\frac{7}{10}$ of the activities were relevant, the score should be higher. Table 4 reveals that several irrelevant documents were ranked higher than others that were relevant, explaining the relatively low score. However, NDCG has proven to yield good results with its logarithmic reduction factor [5]. Based on the relatively low score, and the values obtained from precision and recall computation, one might conclude that increasing k would not necessarily improve the system.

Not all aspects of a recommender system that should be evaluated are technical. In initial stages of developing such a system, a more user-centric evaluation of the system is needed. A proof of concept (POC) is a demonstration, the purpose of which is to verify that certain concepts or theories have the potential for real-world application. POC is therefore a prototype that is designed to determine feasibility, but does not represent deliverables.

3 Related work

This chapter covers some recent research that has been done regarding recommender systems that could be relevant for Refugenious, as well as selected existing systems that relate to the project. The paper mainly focuses on research and systems that pertain to job recommendations as there is no research done, to the best of our knowledge, specifically for the aspects of this paper's main problem. Job recommendation was chosen as the main research domain as parallels can be drawn from finding the right job and the right activity for a given user, especially as activities include volunteer work.

3.1 State of the art

The research revolves around solving problems with existing solutions for job recommendation, and their proposed fixes will be assessed as candidate methodologies for designing Refugenious with the scope of this paper in mind.

3.1.1 Collaborative filtering

In research done by Zhang et al. [6], a CF approach to recommending jobs to users using *implicit* positive feedback is introduced. The final system interprets a user applying for a job as interest from the user for that job, and rating history of users is constructed from information about the jobs the user has applied to in the past. All other implicit feedback is disregarded for simplicity. To further improve the system's predictions, the users' resumes and details of recruiting information is considered in the recommendation algorithm.

First, the paper weighs the two main approaches of CF, user-based and item-based, against each other. Two systems are developed, each following the methodology of one of the approaches. Experiments with three different similarity measures, and comparison of their precision, recall and F1 scores leads the paper to conclude that the item-based system performs better for this task.

First, the system finds all other users that have applied to the same jobs as the current user. For each of these users, all other items they have applied to, that the current user has not yet seen, are gathered into a candidate job pool. These are the jobs the system will predict the active user's relevance rating for. The item-based system makes its prediction of candidate job $Item_j$ for active user U_i based on the following formula:

$$Pref(U_i, Item_j) = pref_0(U_i, Item_j) \cdot w_h(Item_j) \cdot w_c(U_i) \tag{4}$$

Here, $pref_0(U_i, Item_j)$ is the *sum* of, while $w_h(Item_j)$ is the *average* of a candidate job item's similarity to jobs the user U_i has already applied for. $w_c(U_i)$ denotes the similarity between the active user and all other users that have also applied for $Item_j$. Jobs are represented as vectors such as {job name, location, job type, field, category name}, while user vectors have the form {college, major, degree, home place, gender} etc. and the similarity between to users or items is computed using Cosine similarity.

Demanding the user to disclose detailed information about a their past experience in this way, increases the complexity and work needed to create a profile and begin using the system, and for Refugenious it is important to keep this threshold low. In addition, the final system may have avoided the need for explicit feedback, but it is clear it is still much dependent on users having rated several items. Therefore, the system's lack of support for the cold start problem causes new items and users not to be handled well in the system. As a main goal of Refugenious is to give fast and immediate recommendations of any item to any user, this proposed system is insufficient for the scope of this project.

In research done for movie recommendations by Choi et al. [7], a CF system is implemented which seeks to avoid the cold start problem and requires less intricate information from the user. The system requires the user to submit their genre preferences, here called category preferences, before using the system, and represents movie items as vectors of its categories. Instead of computing groups of similar users based on their preferences, the system uses information about rating and category correlation of all the items and compares them to users' genre preferences.

The data set used in the paper contains 18 movie genres and close to 3900 movies. The correlation between all genres are computed, resulting in a percentage that reflects the co-occurrence of the genres. The system then calculates predictions of a user U's rating of a movie M using the formula

$$Rating(U, M) = \frac{\sum\limits_{i \in U_{gp}} (\sum\limits_{j \in M_g} C_{i_n, j_m} \cdot \mu R_M)}{n(U_{pg})} \tag{5}$$

where U_{pg} denotes the user's set of preferred genres, M_g is the set of genres categorizing the movie, C_{i_n, j_m} is the correlation between the genres i_n and j_m. The movie's overall mean rating μR_M is also taken into account in the calculation. Testing the recommender system against a simpler calculation method that does not include genre correlation, the paper's proposed system performed better.

The choice to use overarching categories for the item descriptions seems suitable for recommending *activities* as well. Choosing different domains like *sports, technology, elderly care* etc. are not only well suited for categorizing any activity (volunteer work, events and courses), but could also work as a unified way of describing the content of the activities as well.

However, Choi et al attempted to get around the cold start problem, but only dealt with new **users**. The calculation still needs the items to have been rated, preferably a great deal, and Refugenious must be able to recommend any item to any user at once. Unlike the paper by Zhang et al., the system depends on explicit ratings which is in general not suited for the task at hand. It would decreases Refugenious' simplicity of use for the user as well as present the problem of having to handle rating ambiguity. Users' positive or negative ratings of activities and volunteer work could be a result of any number of factors, from the organizer's execution of the event to the actual content of it, making it difficult feedback for a recommender system to interpret.

3.1.2 Content-based approach

To avoid the problem of recommending newly introduced items in a recommender system, one can look toward content-based approaches.

In research by Diaby et al. [8] the focus is on recommending jobs to users of Facebook and LinkedIn by processing parts of their profile data and data from job listings. A user's social media profile data is twofold; interaction data, which is information the user has submitted, and a user's social connections' user data. The system proposed in the paper uses only the interaction data.

In their data set, the job listings as well as the user profiles are made up of textual elements, called fields, such as job title, job description and profile summary. Diaby et al. explores two methods for using this information to make its system's prediction, choosing to opt for the second.

Here, each user and job is represented as a vector of key term elements from their textual profile or description, where each element value represents the importance of the associated term for the document. The vector is constructed using "bag-of-words" strategy and TF-IDF weighting. Contrary to the paper's first proposed method of simply calculating a job's relevance to a user by calculating the Cosine similarity of its vector and the user's vector, this second attempt involved using trained statistical models, Support Vector Machines (SVM). SVM's are known to yield good performance in text categorization [9] and the method proved to increase performance over cosine similarity measures when trained with larger data sets. Without the training, however, the method performs significantly poorer on smaller data sets, even those that were comprised solely of jobs and users that were known to be a match.

Because of the significant amount of training the system needs *prior* to deployment in order perform well, the process is too complex and time consuming for the scope of this project. Furthermore, basing the system on the organization users and refugee users to submit a significant amount of free text information about jobs, past experience etc, is not ideal. For a system that aims to be as simple as possible, it not only requires a lot of work by the users,it also introduces the language barrier problem. InterConnect's, and therefore also Refugenious', intent to support a multilingual user group and several languages adds too much to the complexity for such a scheme.

3.1.3 User and item representation

A paper by Almalis et al. [10] selects a less complicated route to constructing the vector representations of jobs and user. All job listings are vectors where each element is a requirement for the job, and user vector element values reflect how the user measures up to those requirements. The focus of this research is on on the way qualification requirements of a job listing are defined and how it affects the process of matching jobs to eligible candidates.

The paper addresses the inflexibility that results from recommender systems normally assuming a one-dimensional approach to requirement matching; an exact match between a job attribute's required value and a user's actual value. A goal for the research was to support the various *types* of requirements through their system FoDRA, a Four Dimensions Recommendation Algorithm for matching.

In addition to exact match, the paper deals with upper bounds (applicant must be maximum 45 years old), lower bounds (applicant must have at least 2 years of experience within a certain area) and a combination of the two (applicant must be between 18 and 35 years old).

In their experiments, the data set was solely comprised of jobs within IT. The job listings came from various categories such as project management, software engineering etc., and they identified a finite set of possible skills, such as programming knowledge, past experience, language skill etc. For all job listings, the attributes *could* have bounded restraints or not, which must be specified by the company posting the listing.

Although the system requires less complex and easily processed data, it infers that a significant specificity of requirements is necessary, in order to reap the rewards of various bounding possibilities. For example, a company looking for a seasoned web developer, will likely require a lower bound of some years experience within web development, not just in IT in general. This granularity and requirement of specificity when creating a job listing might be reasonable when focusing on a single domain, such as IT, but would be cumbersome to scale for the various domains that must be represented in Refugenious.

Additionally, the fact that each user and job listing must be represented as a vector of all the possible requirements any of the job listings in the collection may have, does not scale well for this project's scope. Implementing this technique for several domains at such fine granularity would lead to large and mostly empty user and job vectors.

3.1.4 Hybrid system

Lu et al. [11] attempt to combine the simplicity of content-based recommender systems with the performance benefits of collaborative filtering. They propose a hybrid recommender system, HYRED, that uses both the content of job listings and user profiles, as well as recorded user interactions, to provide personalized recommendations.

The system is created to make two-way recommendations, meaning both recommending candidates for jobs and jobs to candidates. Employers themselves are also an entity, and can be recommended to a user as a suitable employer and vice versa. The content and interactions data from and between the entities, are modeled using a directed, weighted and multi-relational graph.

The content of user profiles and job listings are exploited in two ways to by the recommender system. The first is profile *matching*, creating a bidirectional relation between job and user identities. This is based on the premise that if a user's CV content matches that of a job's description, the two entities are probably interested in each other. The second way is to find entities of the same type that are *similar*, and create a bidirectional relation based on users with similar profiles tending to have similar interests and opportunities.

Interactions build relations between entities in a different manner. Employers *post* jobs, and users can *apply* for, *favorite*, and *like* these jobs, creating relations between the entities. In addition, jobs that users *visit* are registered and considered in the graph.

The various inter-relations of entities are summed up in the table below:

	Candidate	**Employer**	**Job**
Candidate	Similar	Visit, Like Match, Favorite Apply	Visit, Like Match, Favorite Apply
Employer	Visit Favorite Match	Similar Visit	Post Visit
Job	Match	Posted	Similar

Table 4: Inter-relations of entities in HYRED

Relations are weighted as follows: $apply > favorite > post = like > similar = match = visit$. The graph is used to predict several recommendation cases, but those relevant to this paper are the ones regarding recommending jobs to candidates. A sample of recommendation cases are illustrated in the following example.

> *Examples*: The system will likely recommend Job C to $User_1$ because $User_2$ liked Job C, and $User_1$ and $User_2$ have similar profiles. The system might also recommend Job A to $User_1$ because $User_1$ *liked* Job A, and Job A and Job B are similar. Finally, $User_1$ is recommended $Employer_1$ because $User_1$ applied for Job C which was posted by $Employer_1$.

Their experiments show that HYRED outperforms both content-based profile matching and collaborative filtering on recommendation precision and user coverage. However, the offline data used for the experiments, they selected only 39.6% of the users, comprised of the users that had adequately complete profiles and/or generated interaction data. The results of the experiments might therefore be skewed due to the virtually ideal data subset.

To outperform the purely content-based approach, HYRED relies a great deal on interactions data. However, only 9% of the users in the data set had interactions data. This is explained in the paper by the features being relatively new. However, there is no guarantee that users are likely to find the functionality useful, or be prone to using it. In addition, the ample ways for a user to express sentiment towards other entities, such as liking and favoring, seems excessive and not in line with the simplicity desired for Refugenious.

Aside from the interactions, the system relies on the profile similarity calculations. Profile content of jobs and users are parsed into plain text files and analyzed using Latent Semantic Analysis. Using this approach for calculations leads to less precise results than comparing each entity's characteristics directly.

In addition, profiles do not have a rigid form. It may be more practical for users and employers to enter information about experience and job requirements in whatever form they wish, but it poses problems for a recommender system. Resumes with similar content might be overlooked, due to differing vocabularies. Keywords must precisely match resumes' items, and substrings such as "program" and "programs", can cause false positive matches.

The approach of drawing recommendation conclusions based on relations drawn between entities is interesting. However, too many relation types are introduced to be practical for Refugenious, and some entity recommendations do not apply. For instance, recommending an employer to a user in the context of InterConnect and Refugenious, as the organizations posting activities do not necessarily only post activities within the same domain.

3.2 Related systems

There are several web sites that provide information on available volunteer work, such as frivillig.no and www.frivilligtrondheim.no, but there does not seem to be any kind of recommender system implemented in any of them. Some of them do not even require a user to create a profile at all. Consequently, this section will focus on systems for finding paid work, as there are more relevant systems to look into as well as inherent parallels between how systems recommend paid jobs and volunteer work, and even activities, to users. The section also includes reflections on advantages and disadvantages of the systems' chosen techniques relative to how they would work for InterConnect and Refugenious.

3.2.1 Finn.no

Finn.no is an online market place which conveys various online services and listings, including job listings. Finn.no uses a pure collaborative filtering model for recommending jobs to users, where only past behavior, in this case page views, is used as a similarity measure between users to predict jobs of interest [4]. The system pays no attention to the content of the job listings or user profiles themselves, only the pattern of interest for certain jobs found among the users.

The website collects the user id of every page view of a job listing, and registers the page view as implicit positive feedback, meaning interest shown by this user for this job. A matrix of all job listings as rows and users as columns is made which conveys all registered interest between a user and a job. Consider an example of three active job listings for jobs 1, 2 and 3. Three users, Alice, Bob and Carl have viewed job 1 and 2, 2 and 3, and 1, respectively, which results in the matrix illustrated in Table 5. This aforementioned matrix is then used as input to ALS, the Alternating Least Squares algorithm, which estimates the missing values, i.e. the items that have not yet been rated by the user, based on the existing values in the matrix.

	Alice	Bob	Carl
Job1	1		1
Job2	1	1	
Job3		1	

Table 5: User-Item matrix of Finn.no

[4] Information Collected from investigating Finn.no and through communication with FINN's Product Manager Per Erik Marton and Data Scientist Simen Eide

This solution is effective for large data sets and is easy to implement. Disregarding the content of job listings and/or user profiles entirely saves a lot of effort when it comes to data extraction, analysis and comparison. However, the disregard of content and explicit user preferences, such as desired work domain, skews the system to recommend generally popular jobs, such as "Uber driver", to most users, even though they may be unsuitable. In addition, Finn.no has no support for the cold start problem, meaning that before a user has actively searched for and viewed several job listings, no recommendations will be given to her. For Refugenious, recommendation functionality is especially crucial for new users, as having to do manual searches on an unfamiliar website weakens the system's desired helping effect.

3.2.2 LinkedIn.com

LinkedIn is a social networking service with a professional focus and job search tool. Its goal is to connect the world's professionals to make them more productive and successful. Members create an online resume by uploading information about their skills and experience etc, and can browse and apply for the different jobs posted by businesses and organizations. LinkedIn provides suggestions for jobs a member might be interested in based on how features extracted from the user's profile and the job listings match, making its system content based [5].

Both job listings and member profiles are largely free form text, apart from "Skills and expertise" where a user may select several out of predefined skills, such as "Photoshop" or "Customer Service". Some fields are specific, such as "Job title" or "Company", but are still free text. Because of this, a challenge for LinkedIn's system is the need for extensive textual analysis in order to match the right applicants to the right jobs.

From the text in the profiles and job listings, feature vectors, $v_{j,m}$, are created for {job-member} pairs. Each feature is the result of a similarity function for corresponding field texts, for example the similarity between a member's profile summary and the listing's job description. The similarity is found by creating a vector, $v_{w,f}$, of term weights for each field, using term frequency and inverse document frequency,

$$w_{t,f} = tf_{t,f} idf_{t,f}$$

and then computing the cosine similarity between the fields' vectors. In addition to the ones extracted from the profiles of users, features may also include network and company related information, like how many of a member's connections work at a given company, or if a member has previously worked there.

Because it is not feasible to compute every pairwise comparison of jobs to members, LinkedIn treats the recommendation problem as a search problem in order to provide a ranked list of jobs to each user. The search of selected terms in certain fields result in a list per term that consists of the job listings that contain the term, along with the frequency of the term in that listing.

The lists are sorted by the job ID's in an increasing order. The scoring of the items, in this case; job listings, then becomes the task of merging these lists into one, which will usually only consist of the top 10 items at a time, where the items that appear in the most lists and with the highest term frequency are ranked the highest. Some filters are also applied in order to eliminate irrelevant jobs from the recommendations, such as jobs the member has already applied to.

LinkedIn's system has proven to be very effective, but its complexity seems beyond the scope for this paper's proposed system. Largely basing user profile and job listing representations on free text analysis would also be inconvenient, as the proposed system will likely include several languages, adding to the complexity.

3.2.3 NAV.no

NAV.no [6] is the web page that contains the information and services provided by the Norwegian labor and welfare administration. It includes possibilities for looking for work, where anyone can

[5] Information collected from investigating LinkedIn.com and lecture material by Paul Ogilvie, Staff Software Engineer at LinkedIn 2012 at http://techtalks.tv/talks/recommender-systems-at-linkedin/57914/

[6] All information was collected by investigating NAV's web page NAV.no

create user, upload their CV which can be made public for employers to find, and search for jobs.

As of today, there is no recommendation system implemented in this service to help users navigate through their comprehensive amount of job listings. It is, however, possible to narrow down the list of prospects by selecting very specific domain criteria. Users must manually search for jobs, either by area or work domain, such as health, farming or IT. For registered users, it is possible to store search criteria, relieving the user of having to specify them manually with every visit to NAV.no, and have jobs that match the search criteria sent to their e-mail address.

The CV is uploaded in a structured manner, with many of the fields, such as work experience and relevant competence, strictly selectable by predefined categories. This way, requirements in job listings can be easily and uniformly matched to a user's CV. However, the amount of categories and subcategories a user must go through can make the process cumbersome and confusing. Also, the categories and subcategories may be incomplete or misguiding when it comes to representing a user's experience and competence.

Although NAV.no allows having job listings that match certain criteria sent directly to your e-mail, it does not make up for the lack of automation and support in its job search process which a recommender system could provide. Filtering of the search results is also limited to work domain or area, even though the website collects a lot more data from the users and job listings that could be used for narrowing down the listings to relevant ones. The system makes it the job of the users and the employers to find suitable candidates for themselves, and merely acts as a gathering point for them to search.

3.2.4 Other systems

Other known systems are Mojob and JobbNorge.no. Neither have implemented a recommender system, but are nonetheless noteworthy when discussing systems for finding jobs, and both have other interesting features [7].

JobbNorge.no lets job seekers as well as businesses create profiles and offer extensive functionality for the applicants. In place for a recommender system, JobbNorge adds a twist to the the functionality for saving search criteria. Users can create what is called job agents, that are in charge of notifying the user of new hits that match the criteria. Each job agent can be given a name and various criteria to look out for, such as desired domains, locations and key terms.

Creating job agents is just one of many services JobbNorge offers to aid its users in the search for a job. In addition, the web site has an integrated mail service, users can create several types of CV's or import CV's from files of certain format, and create a general application text which is stored in the system for later use. Figure 1 depicts the landing page for a logged in user, depicting the numerous actions a user can take.

[7] All information gathered from investigating JobbNorge.no and the Mojob mobile application, as well as interviews with Tobias Nervik, CEO and Co Founder of Mojob

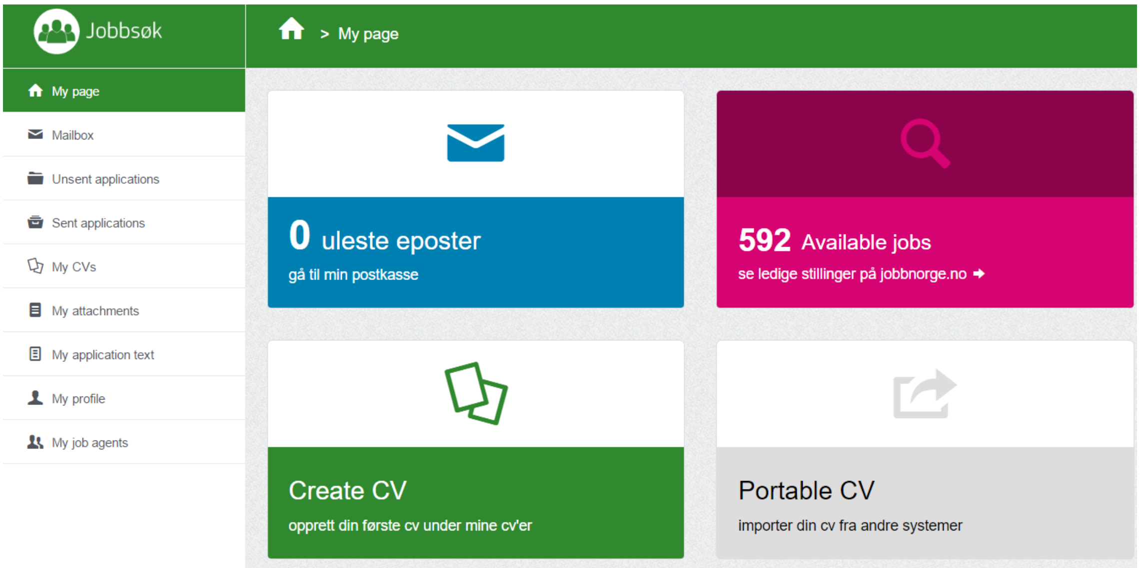

Figure 1: Landing page of Jobbnorge.no

Although perhaps of great support for the avid and technologically adept job seeker, the amount of functionality and arguably excessive number of possible actions shown in the user interface may add too much complexity for this paper's intended target users.

Mojob is the only of the mentioned systems that is solely available as an app for smartphones and tablets. Users create a profile and can auto-generate a CV from their LinkedIn or Facebook profiles, although there seems to be trouble with data extraction from LinkedIn. The application filters the job listings by the user's specified domains and sorts them by distance from his or her location. The maximum distance for the jobs he or she is searching for can be adjusted manually. One applies for a job by sending a 200 character "pitch" to the employer, and the CV is automatically sent with it. If the employers approve, a chat is created for the two parties as a path of communication. This chat functionality chiefly puts the responsibility of the recruitment process on the users ability to market themselves. This is not necessary for Refugenious, and assuming language skills of the user is also not ideal.

4 Approach

In Chapter 3 we saw that the main issue is that recommender systems delivering that yield quality recommendations today require extensive data from the user, explicit or implicit. This is not an option for a system with refugees as the target user, as it increases the threshold of understanding and use. A challenge for Refugenious is therefore to be a system that can produce instant recommendations that are of interest with limited information.

This chapter is an overview of the approach to developing the proposed solution to this thesis' main research objective based on the theory from chapter 2, and the discoveries from the research in chapter 3. The first section addresses InterConnect's role in the planning and development of Refugenious, while section 4.2 deals with the implementation of the recommender system. Section 4.3 to 4.6 give details about the various components of Refugenious, revealing the algorithms intended for similarity measures, activity recommendation and user modeling.

4.1 InterConnect

As previously stated, Refugenious will be an integrated part of the web portal Interconnect. Therefore, this section provides an overview of how the two systems relate to each other, as well as the requirements of the web portal, which need to be considered in the development of Refugenious. InterConnect is currently being developed as this thesis is written, and this section also includes some mock ups of the design to illustrate the functionality of the portal.

4.1.1 Architecture of InterConnect and Refugenious

Refugenious and InterConnect roughly represent the back-end and front-end of the composite service respectively, but some of their parts are intertwined. How InterConnect and Refugenious relate to each other is illustrated in the figure below:

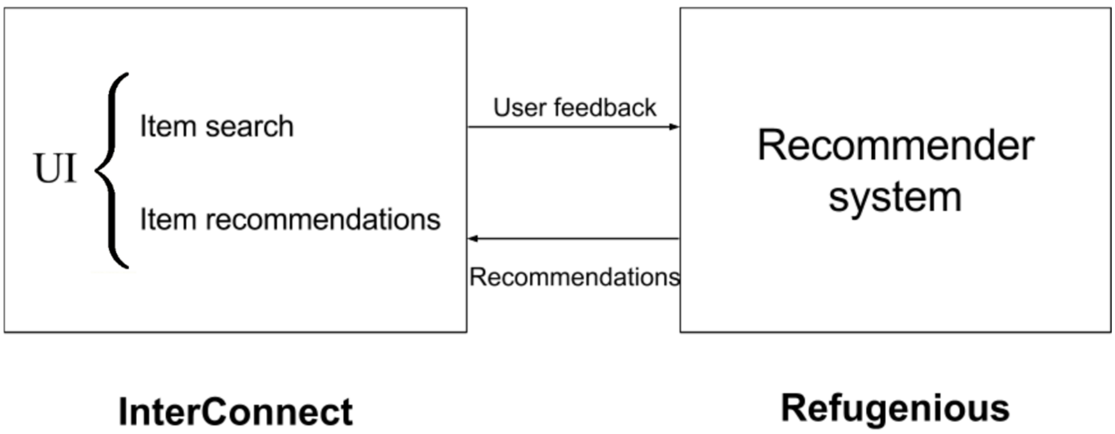

Figure 2: Overarching architecture of InterConnect and Refugenious

4.1.2 System requirements

This section focuses solely on the *functional requirements* of InterConnect, which also affect Refugenious. Although non-functional requirements like availability and and response time are important, they are not included in the scope of this project. Functional requirements are the tasks which the system must be able to perform in order to fulfill its main objective: to help refugees find and participate in activities that will help them build competencies, connections and experience.

As the system must accommodate two types of users, refugees and the organizations posting activities, the requirements must reflect the concerns of them both. Based on research done regarding what functionality and information the users want and need, the following set of requirements have been formulated:

The system in general must:

- *allow users to sign up, sign in and sign out*
- *require certain information from users in order to allow them access, such as organization number or e-mail address*
- *allow authenticated users to set and update information pertaining to their user profile*

Considering refugee users(RUs), the system must:

- *provide simple and intuitive use of the web portal through a user interface tailored/customized for refugee users*
- *allow authenticated RUs to view activities they are eligible for*
- *provide a set of activities that are especially suited to the current RU*
- *allow authenticated RUs to apply for a participant slot on an activity*
- *notify authenticated RUs if their application is approved or denied*
- *allow authenticated RUs to view their resumé, which is an overview of personal information and activities they have participated in.*

Considering organization users(OUs), the system must:

- *allow authenticated OUs to post activities, with or without participant requirements such as age limitation*
- *allow OUs to view profile of RUs that have applied for a slot in their activity*
- *allow OUs to approve or deny RUs applications to participate in activities posted by their organization*

4.1.3 User interface concept

InterConnect is the portal the users interact with Refugenious. It is responsible for generating the data Refugenious uses, through organizations posting activities, and users creating profiles and providing feedback to the system. The user interface is being developed using React[8]. Below is a is a mock-up of the user interface, showing the process of users setting up an account.

[8]https://facebook.github.io/react/

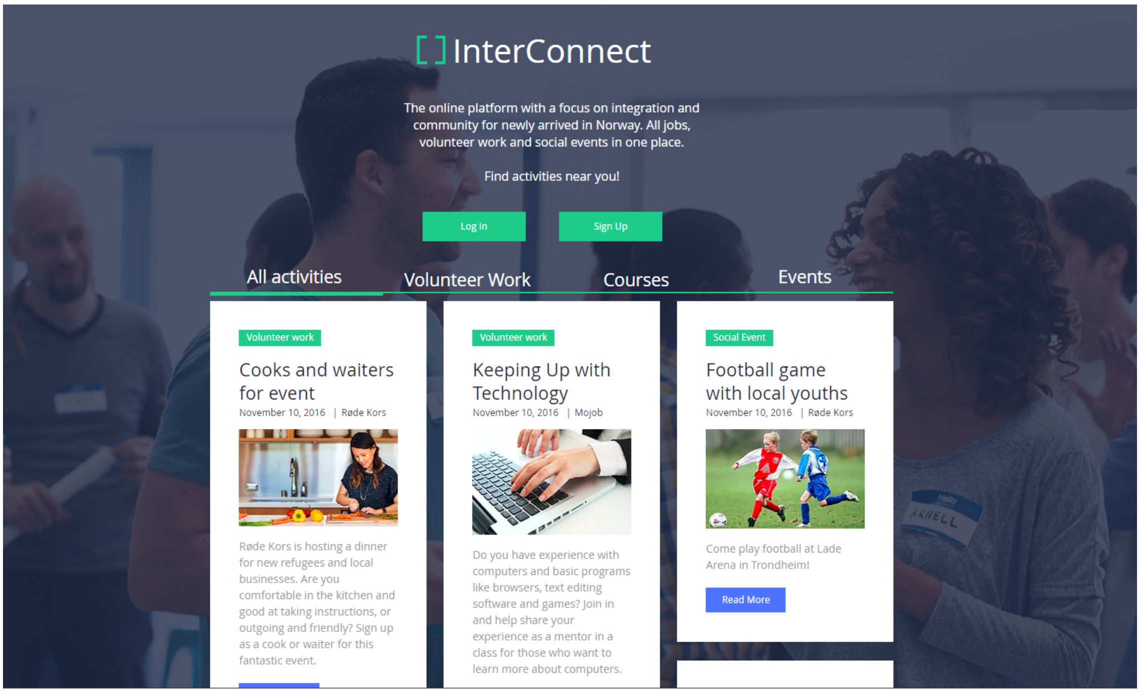

Figure 3: Landing page for users without an account. Users can sign up or sign in, and view activities, but not participate.

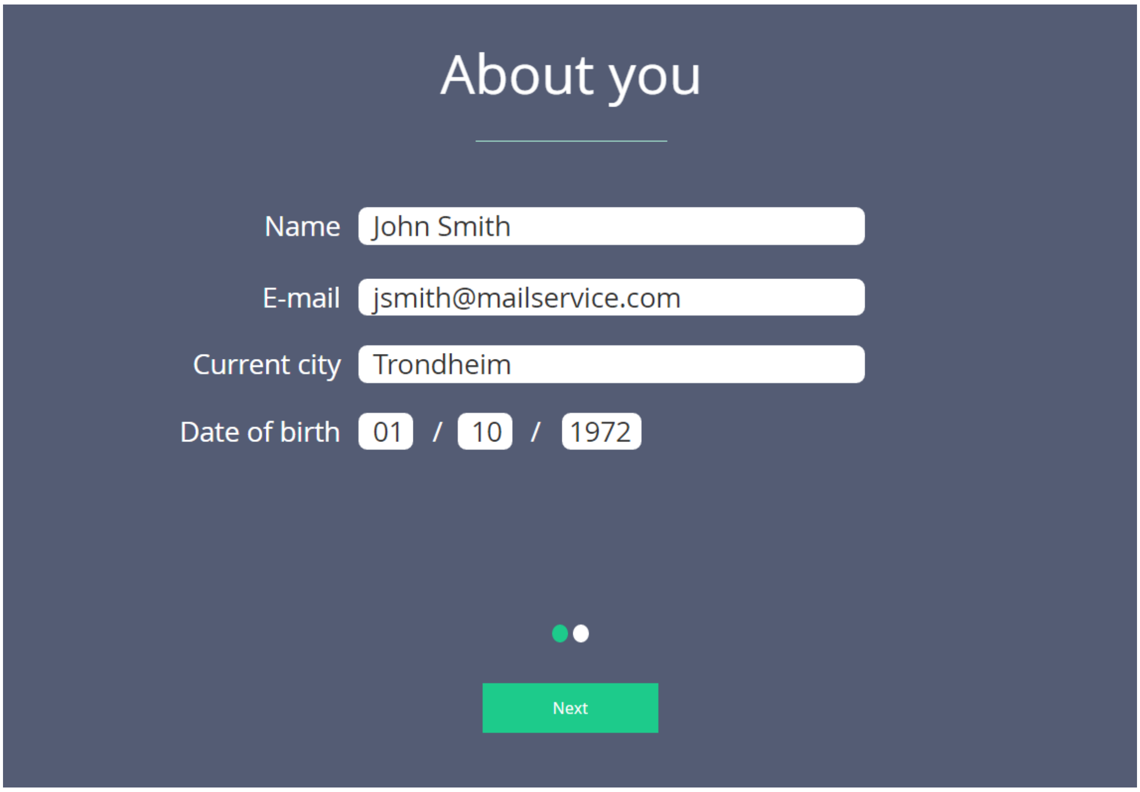

Figure 4: Sign up - The user must provide basic personal information

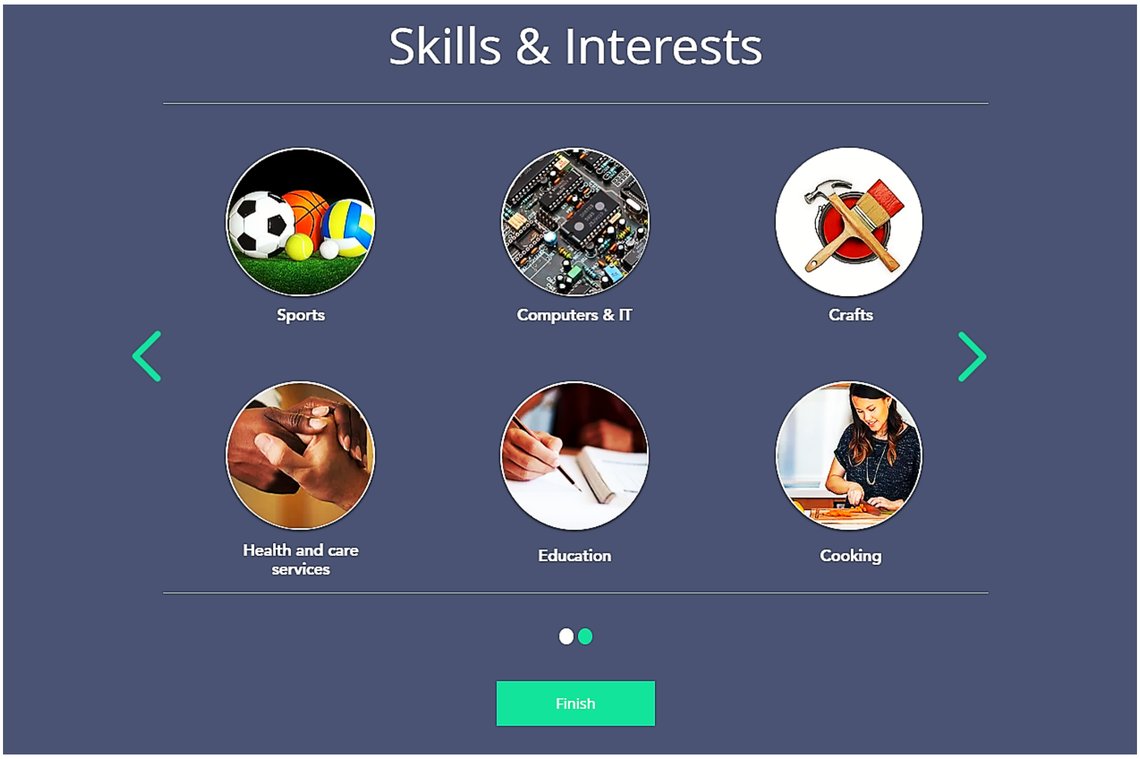

Figure 5: Sign up - The user chooses preferred domain categories based on their skills and interest.

While Refugenious is responsible for using the information given by the users and make recommendations to them, it is InterConnect which presents these findings to the individual user.

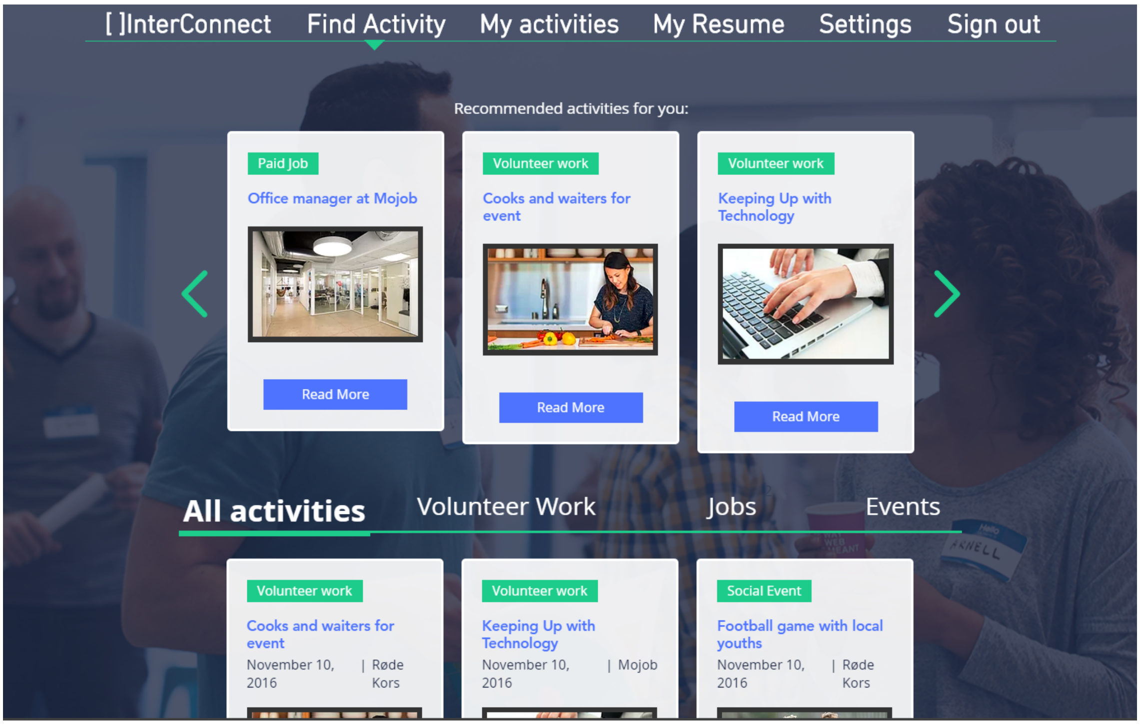

Figure 6: Landing page for a user that is signed in. Recommended activities will be clearly displayed and easily accessible to the user. Note that InterConnect will also include offers of paid work, such as odd jobs, but they are not included in the scope of Refugenious.

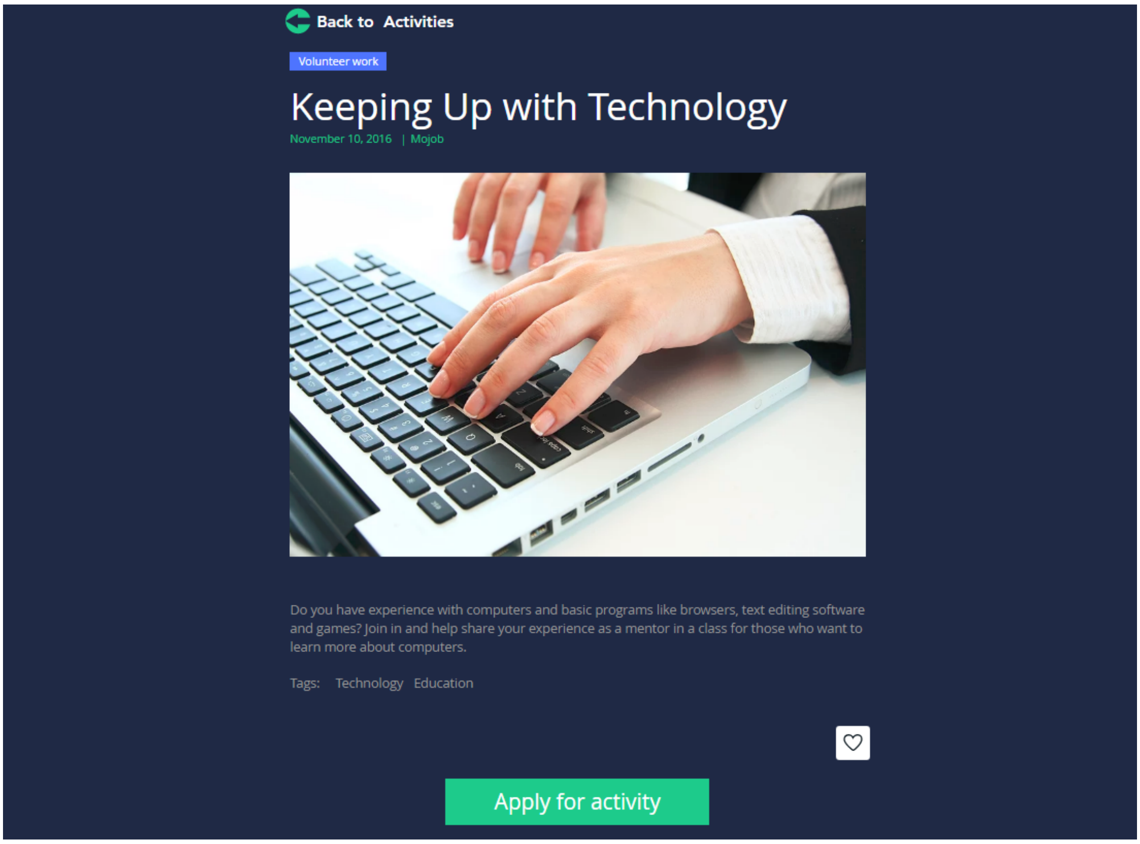

Figure 7: Users can click on activities to read more and apply to participate in them. The organizers of the activities are notified of new applicants and can peruse their profile before accepting their application.

4.2 Refugenious: System implementation

Based on the theory presented in Chapter 2, and the research examined in Chapter 3, a content-based approach to creating a recommender system was chosen. Its ability to instantly recommend new items to users is crucial to the system's purpose of fast and easy recommendation. The problem of users having to rate a certain amount of items before a user profile can be established is solved by requiring the users to provide domain categories of interest upon signing up to use InterConnect. As organization users have no direct impact on the recommendation system, the *user* in this section will exclusively refer to refugee users.

4.2.1 Design process

When implementing a content-based recommender system, there are several aspects to consider and design decisions to make. The following list shows the central steps of the process:

- Building item representation (item profiles) from a set of attributes
- Building user representation (user profile) using the same attributes
- Determine and prepare data to use as input for recommendation engine
- Devising similarity measure to compute likeness between items and user profiles
- Ranking the items and sorting them by relevance
- Presenting the recommendations to user
- Devising method for refining user profiles based on feedback from the users

4.2.2 Architecture and technology

This section gives an overview of Refugenious' proposed architecture and how its components interact with each other, as well as the various tools and technologies selected for implementing the system.

Proposed architecture

Refugenious will mainly be comprised of three central components; the filtering module, the recommender engine and the user modeling module. The data flow and interaction between the components is shown in Figure 8.

The *filtering module* takes in all the available activities and filters out the ones that are known to be ineligible for the active user. Some organizations have age or language restrictions on their activities, or restrictions based on how long the refugee has been in the country. The filtering component was added to handle these restrictions and filter out unfit activities before the recommendation process to reduce the load. In addition, the filtering module will sift out activities that are outside a set distance from the user as well as activities the user has already rated.

The set of filtered items (activities) is then sent to the *recommender engine* for processing. This module is in charge of calculating the degree of relevance of the items in regards to the active user and return activities that a user will likely be interested in. When the computation is done, the items are sorted into a list by decreasing relevance which is presented to the user.

When a user marks an item as relevant, this feedback is communicated to the *user modeling module*. This component is responsible for updating the user profiles based on their feedback, and thus refining the system to hone in on the users' interests. The details of the user modeling module, as well as the other components is explained further in Section 4.3 to 4.6.

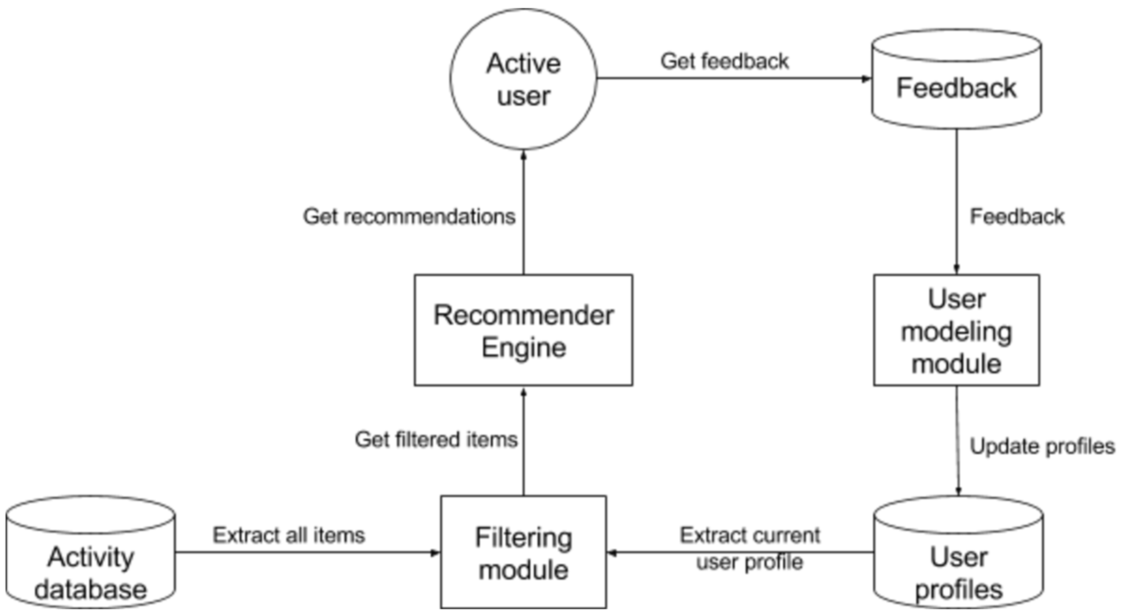

Figure 8: Proposed architecture of the recommender system

Technology & Frameworks

As mentioned in section 4.1, the JavaScript library React will be used for developing the user interface. For creating and handling user and activity models, the web framework Django will be used, which is accessed by React components through a REST API.

Refugenious itself will be written in Python, which is widely used within development of recommendation systems and therefore also extensively supported through recommender-engine frameworks, such as Crab [12]. This is also compatible with the models, as Django is a Python framework.

For the database, PostgreSQL [9]is used on a Docker[10] droplet. PostgreSQL is an open source database with support for virtually all SQL constructs and has a strong reputation for reliability, data integrity, and correctness. It can also run on any major operating system. Docker provides support for any operating system and their droplet sizes are easy to scale up or down, depending on need, which makes it suitable for a project under development.

4.3 Representation of users and items

The representation of users and items in a recommender system determines much of its functionality. The representation method chosen for this project is a variant of the Vector Space Model, where users and activities will be represented as vectors. Instead of one m-dimensional vector, however, both items and users will be represented as *two* vectors; the *content vector* and the *restriction vector*.

The first will only reflect the content of the activity, while the second deals with the possible restrictions that might be imposed on an activity. Only the first will be used by the recommendation engine, while the second is used by the filtering module which will be explained in the next subsections. The content vector consist of the high level domain categories the activities typically revolve around, such as *sports*, *computers* & *IT*, and *cooking*. For cases where there are three possible activity restrictions, a user's age, and whether or not he/she speaks English or Norwegian, and four possible domain categories, the restriction vector and content vector are defined as follows:

$$\vec{v_{res}} = \{age\_restriction, speaks\_English, speaks\_Norwegian\}$$

$$\vec{v_{cont}} = \{category1, category2, category3, category4\}$$

An example of how the user and activity vectors correlate is shown below:

$$vector(activity1) = \begin{cases} v_{res} = \{18, yes, no, \} \\ v_{cont} = \{1, 0, 0, 1\} \end{cases}$$

$$vector(user) = \begin{cases} v_{res} = \{24, yes, no, \} \\ v_{cont} = \{1, 0, 0, 0\} \end{cases}$$

Here, the activity requires a user to be 18 years of age and be able to speak English, and belongs to category1 and category4. User profiles consists of the same two vectors with the same attributes. The values of the restriction vector attributes will reflect how the user would measure up to the restriction. The user in this example is 24 years old and speaks English. The restrictions are therefore met, and because the activity also falls under a category the user prefers (category1), it is likely activity1 will be recommended for this user.

The domain categories are the aspects Refugenious considers as an item's content. Consequently, the content vector's elements are comprised of an attribute pointing to one of the domain categories, and a corresponding value. The value is either 0 or 1, where 1 indicates that the activity belongs to this category. When an activity is created, the creator must specify its categories, and refugee users must provide what categories they prefer upon signing in to the system. This gives Refugenious the information it needs to recommend any item to any user as soon as it is introduced to the system.

Domain categories were adopted to represent content because they are high level enough to describe all the activity types (events, courses and volunteer work) while still specific enough to reflect the activities' essence. The complexity of this scheme is suitably uncomplicated, as it requires little work from the user and eliminates the need for complex processing. A user profile could be inferred from their social media account activities and profiles, or through more specific information given by the users when they sign up to InterConnect. However, this would require complex processing by the system, and is beyond the scope of this project.

[9]https://www.postgresql.org/
[10]https://www.docker.com/

Example: An activity is created that involves teaching seniors basic computer programs, such as Microsoft Word and simple web browsers. The organizer might mark this under several domain categories, such as *education*, *computers and IT*, and *elderly care*, which would set their corresponding attribute values to 1 in the content vector representation. This activity might require the volunteers to speak at least some Norwegian, setting the value of the attribute corresponding to *speaks Norwegian* to 1. If the organizer decides that any participant must be at least 18, the *age* attribute of the activity will be set to 18. If there is no value set for the restriction attribute, it will be set to -1, a default value the filtering module knows how to process.

As previously stated, the users will be prompted to specify domain categories they are interested in when setting up their account. With values of 0 and 1 in the user's content vector indicating no interest and high interest respectively, all preferred domain categories provided by the user upon signing in will get a value of 0.65 in his or her content vector. Dividing the scale between 0 and 1 into six, which is a classic rating scale, gives 0.65 an equivalent rating of a little under 4/6. Such a rating is considered to reflect definite interest, while also leaving room to let the recommender system determine what domain categories are of the *most* interest to the user. Further explanation of how this mechanism works is provided in Section 4.6.

4.4 Filtering module: Preparing the data

Not all activities need to be considered by the recommendation engine. There are several aspects of an activity that explicitly lets the system know it is not suitable to recommend to a user. The filtering module is in charge of sifting out these unsuitable activities in two phases.

In the first, the module takes in all activities in the data base and filters out the ones the user has already applied to. It also disregards activities outside of the user's city of residence. As InterConnect also intends to support finding paid jobs for users who have a work permit, and Refugenious does not, these items will be filtered out as well.

In the second phase, the filtering module investigates the restriction vectors of the remaining activities and checks if any are imposed. The module recognizes this by observing that the vector contains an attribute value not equal to -1, a default value set for all restriction elements not given a restriction value by the organizer. If there are any restrictions imposed, the filtering module performs checks for the different attributes to make sure the user meets the requirements. Only if the user requirements are met will the filtering module pass the item along to the recommendation engine. Below is a pseudo code excerpt showing how the filtering module sifts out ineligible activities.

```
def filter_out_restricted_activities(user, activities):

    eligible_activities = {}

    for each activity in activities:

        #Ensures activities are not yet rated or outside the user's residence area
        if user not rated activity & user.residence == activity.location:

          #Checks the vectors and makes sure the activity restrictions are met
          if user.res_vector meets activity.res_vector:

            eligible_activities.add(activity)


    send_to_rec_engine(eligible_activities)
```

4.5 Recommendation engine: similarity measures and ranking

The recommendation engine measures the various items (activities) against the user profile of the active user to see if the item might be of interest to him or her. It does so by comparing the content vectors of the user and the individual items, and determining the similarity between them.

The higher the similarity between the vectors is, the higher the chances are of the user finding that item relevant. The restriction vectors are not used in this part of the process.

The similarity measure that will be used in the system is cosine similarity as shown in equation 2.1. It yields good results for comparing vector representations when the set of terms, or in this case attributes, are the same [13]. The result of the computation will be a number in the range [-1, 1], where 1 means identical match, 0 means maximally dissimilar, and -1 indicates that the two vectors are inversely similar. Here, we are only interested in whether the items are directly similar to the user profile or not, and will therefore only focus on scores between [0, 1].

These similarity scores will be ranked by sorting the values by descending scores. Only the top k most similar items will be offered to the user. Depending on the final user interface of InterConnect, k will be set to a number within the range [5, 10].

Another possible strategy for selecting items to recommend is to return all items with a score above a certain threshold. The reason the k items with the highest score were chosen instead, is because there is no strictly binary classification problem, where there is a clear threshold indicating relevance. The items will not be categorized as being either strictly relevant or completely irrelevant, but rather sorted by *how relevant* they are. The threshold would also be difficult to determine, as comparing user preference vectors and item content vectors often inherently yields low scores even though they are relevant.

> *Example*: The vectors of a user that has specified four out of six preferred domain categories, and an activity belonging to one of them look as follows:
>
> $$pref(user) = \vec{v_{cont_u}} = \{1, 1, 0, 1, 0, 1\}$$
>
> $$cont(activity) = \vec{v_{cont_a}} = \{1, 0, 0, 0, 0, 0\}$$
>
> When comparing the two vectors using cosine similarity, the activity scores only 0.354. However, an activity that only belongs to one of the user's preferred categories may be just as worthy of being recommended as one that belongs to several. Preferring domains like *sports* and *cooking* might indicate that a user would like activities falling under either category, not necessarily under both.

4.6 Feedback and user modeling

Feedback is given implicitly by users through their interaction data. Implicit feedback can be noisy and require a lot of interpretation and processing, so for the scope of this project the focus will be on two types of implicit feedback. The first is positive feedback being registered when a user applies to participate in an activity. The second is from a UI feature where users can "X out" a recommended activity, which is registered as negative feedback.

Explicit feedback was disregarded for this project, as textual feedback would require Natural Language Processing and sentiment analysis, perhaps of more languages than Norwegian and English. The rating mechanisms of CF could falter because ratings in this context would not necessarily reflect a user's displeasure with the domain categories of the activity, but rather the execution of the activity by its organizers, or any number of other possible factors.

The feedback is used by the user modeling component continuously to learn the user's preferences. The module updates their content vectors accordingly. As previously stated, all domain category preferences the user explicitly states upon signing in to the system have their corresponding vector values set to 0.65. The remaining values are set to 0, indicating no interest for these categories.

When a user applies to participate in an activity, the feedback will be stored and sent to the user modeling module, which processes and updates the content vectors once every 24 hours. The domain categories of the activity are interpreted as desirable to the user, and their corresponding vector values are incremented by a value v. As the vector has values ranging from 0 to 1, v must be set accordingly. The adjustment must not be too high, such that a single positive feedback on a category can impact the preference representation too significantly. Nor must the adjustment be too slant resulting in the system not picking up on user preferences as they appear fast enough. If a user has given implicit positive feedback on 10 activities that fall under the same category that was originally set to 0, that category should be deemed interesting and receive a score of 0.65 or higher. Therefore, v is set to 0.07. If adding v to an attribute value of the vector results in it exceeding 1, it is simply set to 1.

> *Example*: A new user has applied to participate in an activity under the domain category *Education* and *Computers* & *IT*. The possible categories are {Education, Sports, Cooking, Computers & IT}, making the item's content vector:
>
> $$cont(activity) = v_{\vec{cont_a}} = \{1, 0, 0, 1\}$$
>
> The user specified *Education* and *Cooking* resulting in a content vector on the form:
>
> $$pref(user) = v_{\vec{cont_u}} = \{0.65, 0, 0.65, 0\}$$
>
> The profile learner user modeling component updates the user's content vector:
>
> $$pref(user) = \{(0.65 + v), 0, 0.65, (0 + v)\} = \{0.72, 0, 0.65, 0.07\}$$

In order to combat the overspecialization that typically arises in content-based recommender system, a decrementation mechanism was considered. User's interest may vary depending on many factors, such as the season of the year, or after developing new skills or interest. The thought behind this mechanism is that users may lose interest in certain activities over time, and the system should therefore penalize domain categories that have not been given implicit positive feedback from the user in a while. However, lack of implicit feedback does not necessarily indicate lack of interest; it may simply be a lack of posted activities under that category. This could be clarified by making the system only penalize the categories which a user is known to have been recommended and not applied to, but this increases complexity of the system and relies further on noisy implicit feedback.

Instead, a mechanism for users to state lack of interest for an activity will be introduced in the user interface. A user should be able to "X out" a recommended activity, removing it from the recommendation bar shown in Figure 6. The activity next in line for recommendation will fill the void.

When this negative feedback is registered, a heavy penalty befalls that activity's domain categories in the user's content vector. The values will be decremented by 0.3, significantly lowering the value, without making irreparable damage to categories that might still be of interest.

5 Evaluation of recommender system

This chapter deals with evaluating the developed recommender system Refugenious. Testing is important to ensure high quality and utility of the system before launching it on the intended web portal. The following sections show the test plans, execution and results of both a technical and non-technical evaluation of Refugenious.

5.1 Test plan

As previously stated, the evaluation of Refugenious will be twofold. In addition to a technical evaluation of the system's accuracy in predictions, there will be a non-technical evaluation to ascertain the recommender system's concept and utility. The reason for this added evaluation is to sufficiently answer this thesis' research questions, that not only regard the accuracy of predictions made by the system, but whether or not this system and its predictions are perceived as helpful tools for integration and the refugees' everyday life.

5.1.1 Non-technical: Proof of concept and interviews

Before any technical evaluation of Refugenious, a user-centric evaluation of the system's concept and utility is needed. This will be done using focus groups consisting of refugees. Focus groups were chosen as they are a great way of getting high quality feedback on concepts and products. Although data has been gathered previously about the state of refugees in regards to integration and activity levels for this thesis, focus groups are a great way to get in *direct* contact with the refugees themselves, who are after all the target audience of the system. This evaluation will provide more thorough answers to the research questions regarding the system's ability to aid with immersion and integration.

The following test plan concerns the steps to test the system's concept and utility through a *Proof of Concept* (PoC) of the web portal InterConnect and its associated recommender system Refugenious. A PoC is a demonstration, the purpose of which is to verify that certain concepts or theories have the potential for real-world application.

The goal of the test is to inquire with the intended target audience of the web portal about their initial reactions to such a concept and gain insights into possible improvements and alterations that could better the user experience and quality of the product, resulting in qualitative data regarding the concept and system. The entire process and corresponding interviews will be recorded, to better obtain all the feedback from the focus groups.

Test subjects:

Ideally, each focus group will consist of 6-8 people, divided into five focus groups consisting of the following desired groups of people:

1. Adults (age 35 ->)
2. Young adults (age 20 to 35)
3. Men only
4. Women only
5. Mixed group (all ages and genders)

The difference of participants within the various groups is done with the intention of discovering any anomalies in the data, for example if the activity level of women is significantly lower or higher than that of the men.

Test plan:

1. Before the demonstration
    - Introductions and getting to know the individual participants
        - Age, gender
        - Status of work and residence permits, duration of stay in Norway
    - Record their current habits and activity levels
    - What are the tools/strategies they use to initiate or join activities today?
    - Are they comfortable using a computer?
2. Presentation of web portal and recommender system. Walk-through of premade demo web portal (as shown in Figures 3 to 7 in Chapter 4), with the participants observing the processes, but not interacting with the portal themselves.
    - Explanation of concept: What it is for and basic functionality
    - Landing page with activities and filters
    - Signing up (3 step process)
    - Home page with overview of available activities and search feature
    - Signing on for an activity
3. Initial reactions
    - Did they understand the concept and uses of the system?
    - Have them explain the concept in their own words.
    - Are they positive or negative to the concept?
4. More in-depth feedback
    - Would they use it? Why/why not?
    - What are the obstacles that could stop them from using it?
    - Is the portal intuitive?
        - Are the *recommendations* intuitive/ easy to find/ helpful?
    - What activities would they want on the portal? (Odd jobs/ volunteer/ courses/ social)
    - Have they seen a similar concept? What other resources do they use now to get the same information?
    - What other resources would they still prefer to use of my portal if launched?
5. About the recommender system
    - Explaining what information the recommender system uses to create its predictions. (Areas of interest and location etc.)
    - What do they think of the sign up process?
        - Too many steps? Complicated? Intuitive?
        - Is the information required considered sensitive by the participants?
        - What information would they *not* be willing to give to such a portal?
    - What are their thoughts on the portal's recommender system? Is it helpful vs manual search?
6. Data gathering through surveys and activity preferences (See plan for technical testing in next section)
7. General feedback and final thoughts

- What are their thoughs on the web portal now vs. before the demo?
- What could be done better?
- Other feedback?

A goal of InterConnect and Refugenious is to provide refugees with a new and better solution for finding activities than the ones that are currently available to them, which makes mapping existing solutions and getting feedback on the proposed solution imperative for evaluation. The system must be regarded as viable for real-world application.

Post-Study Usability Questionnaire Because the goal of the proposed system is to simplify the process of participation in activities as much as possible for refugees, an important evaluation parameter is how easy it is to use, as well as how the users perceive its recommendations. An evaluation of this must also include InterConnect, as it and Refugenious are perceived as the same by the users.

Because it is desired to get as much user feedback as possible, another form of quantitative research was planned for the evaluation, but was not implemented due to time limitations and language barriers. The use of a Post-Study Usability Questionnaire (PSSUQ) is a common and relatively straightforward way of testing ease of use. A PSSUQ is a survey with predefined statements for the user to agree or disagree with after completing test case tasks. The test user would have been given a set of tasks to complete on InterConnect depending on the finished functionality of the web portal. The most important test case for Refugenius would be for the user to find activities that are relevant for themselves. Figure 9 shows a standard PSSUQ survey form.

The Post-Study Usability Questionnaire
Version 3

		Strongly agree 1	2	3	4	5	6	Strongly disagree 7	NA
1	Overall, I am satisfied with how easy it is to use this system.	O	O	O	O	O	O	O	O
2	It was simple to use this system.	O	O	O	O	O	O	O	O
3	I was able to complete the tasks and scenarios quickly using this system.	O	O	O	O	O	O	O	O
4	I felt comfortable using this system.	O	O	O	O	O	O	O	O
5	It was easy to learn to use this system.	O	O	O	O	O	O	O	O
6	I believe I could become productive quickly using this system.	O	O	O	O	O	O	O	O
7	The system gave error messages that clearly told me how to fix problems.	O	O	O	O	O	O	O	O
8	Whenever I made a mistake using the system, I could recover easily and quickly.	O	O	O	O	O	O	O	O
9	The information (such as online help, on-screen messages and other documentation) provided with this system was clear.	O	O	O	O	O	O	O	O
10	It was easy to find the information I needed.	O	O	O	O	O	O	O	O
11	The information was effective in helping me complete the tasks and scenarios.	O	O	O	O	O	O	O	O
12	The organization of information on the system screens was clear.	O	O	O	O	O	O	O	O
13	The interface* of this system was pleasant.	O	O	O	O	O	O	O	O
14	I liked using the interface of this system.	O	O	O	O	O	O	O	O
15	This system has all the functions and capabilities I expect it to have.	O	O	O	O	O	O	O	O
16	Overall, I am satisfied with this system.	O	O	O	O	O	O	O	O

**The "interface" includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the microphone, and the screens (including their graphics and language).*

Figure 9: Standard PSSUQ survey

5.1.2 Technical: Precision, recall and NDCG

Although implementation and execution of a Proof of Concept and user interviews are important, they only refer to a prototype of the system and focuses on determining Refugenious' feasibility.

They do not, however, represent or evaluate the actual deliverables, which in this case are the recommendations of the system. To ensure the quality of the recommendations, measures must be taken to test the suitability of the algorithms utilized in the system, as well as other implementation choices. This will be done by gathering data which will be used as a *ground truth* for the recommendation evaluation algorithms. The data for this will be gathered explicitly from the participants of the focus groups, hereby referred to as *test users*.

In order to conduct a technical evaluation of Refugenious, its recommendations must be compared to actual results. As there are no datasets to our knowledge that contain recorded activity preferences of refugees, this data will be gathered directly from the test users. They will be asked to fill out information regarding their age, gender etc., as well as to specify which of a given set of activity categories they are interested in on a form as shown in Figure 10.

The choice to include age, gender and information about their residency status was included, not only to obtain as detailed information as possible, but to discover any noteworthy patterns or correlations between the data the refugees provide and their activity preferences.

The activity categories were chosen carefully. It was challenging to find categories that would be comprehensive enough to be able to include all possible activities, yet descriptive and specific enough to be intuitive and contained to a suitably low number. To obtain the right categories, the web sites mentioned in section 3.2 were crawled for their categories. These were then merged to fewer categories when appropriate, or removed if found irrelevant to the domain, before several dozen activities, mainly from Frivillig.no and Røde Kors, were manually traversed to see which categories they would fit under. Finally, 14 categories remained, which will hopefully be suitable:

1. Outdoor activities
2. Local community
3. Social activities
4. Art, Literature and Culture
5. Sports
6. Care and Aid
7. Family Friendly
8. Education
9. Politics and debate
10. Environment and animals
11. Faith and Spirituality
12. Technology
13. Food and cooking
14. Arts and crafts

After filling out the forms with their personal information and category preferences, the test users will be given a set of 25 activities and asked to rank the items by relevance on a scale from 0 to 2; 0 meaning irrelevant, 1 meaning somewhat relevant, and 2 indicating definite relevance. The reason for not choosing a binary relevance scale that both precision and recall depend on, is due to the latter evaluation method, NDCG. For precision and recall both scores of 1 and 2 will be considered as relevant, and 0 as irrelevant.

The activity rankings made by the test users will be recorded and used as ground truth and compared to Refugenious' own rankings and recommendations. As mentioned in Chapter 2, several methods have been developed chiefly to perform this evaluation of recommender systems. Of the variations possible, the chosen evaluation methods for Refugenious are the widely used *precision* and *recall*, as well as Normalized Discount Cumulative Gain (NDCG).

Refugenious does not classify every activity as either relevant or irrelevant. Instead, the system ranks all items and returns the top k items with the highest score. This is not a problem for calculating precision, but recall will need to be evaluated somewhat differently, by considering all activities with a test user rank of 1 or 2 as relevant, and 0 as irrelevant.

Precision is central to Refugenious. It describes the fraction of relevant items the system recommends to the user, and the goal of the system is to provide users with the simplicity of having relevant activities clearly displayed and easily accessible. It will not do to recommend irrelevant activities as it is not only irksome for the user, but also means they must search through the activities manually. Precision is calculated as follows:

$$P = \frac{tp}{tp + fp}$$

where tp is the number of recommended items that are deemed relevant by the test user, while fp denotes the amount of recommended items that the test user marked as irrelevant.

Personal information:

Gender: ☐ Male ☐ Female

Age: ☐

Do you have a work permit? (Arbeidstillatelse) ☐ Yes ☐ No

How long have you been in Norway? ☐

Areas of interest:

Please check the boxes of those that you consider yourself interested in.

☐ Outdoor activities / Friluftsliv	☐ Education / Utdannelse
☐ Local community/ Lokalmiljø	☐ Politics and debate/ Politikk og debatt
☐ Social activities / Sosialt	☐ Environment and animals / Miljø og dyr
☐ Art, literature and culture / Kunst, litteratur og kultur	☐ Faith and spirituality / Tro og livssyn
☐ Sports / Idrett	☐ Technology/ teknologi
☐ Care and aid/ Omsorg og hjelp	☐ Food and cooking/ Mat og matlaging
☐ Family friendly/ Familievennlig	☐ Arts and crafts/ Håndtverk

Figure 10: Form given to the focus groups participants for data gathering

As there is no relevance score threshold defining whether or not an activity is relevant or irrelevant in Refugenious, an activity's relevance becomes relative to the number of other activities with a higher relevance score than itself. To compute recall, we define all items not within the top k items as irrelevant by the system. The recall score will then be a measure to check the suitability of the chosen k-value. Recall is computed as follows:

$$R = \frac{tp}{tp + fn}$$

where fn denotes the activities the test user found relevant but were not recommended by the system.

If precision is high, but recall is low, this might indicate that k is too small, while a low precision paired with high recall might imply that k is too high.

Although powerful, precision and recall alone do not necessarily provide sufficient evaluation of the system. Therefore, the NDCG measure will be used to measure the quality of the ranking algorithm itself further, and check whether the premises which guide the scoring algorithm are legitimate.

The premise behind the NDCG is that the *order* of the recommended activities is crucial, as a relevant item that has a low position is less likely to be found and investigated. This is arguably the case for Refugenious and InterConnect. NDCG is computed as follows, using its denormalized version:

$$DCG = \sum_{i=1}^{m} \frac{rel_i}{log_2(i+1)}$$

where rel_i is the relevance score given by the test user to the activity ranked i^{th} by the system, m is the total number of rated items, and the logarithmic function punishes wrongly positioned items. Normalized DCG results in a value between 0 an 1 and is defined by:

$$NDCG = \frac{DCG}{iDCG}$$

5.2 Execution and results

Though the plan was kept in mind, the execution of both the technical and non-technical tests were altered slightly to accommodate the actual test environment. This was due to the challenges that arose during the organization of the tests, as well as to simplify the process for the refugees that were selected as test users. The simplifications were needed due to language barriers and technological knowledge.

5.2.1 Proof of Concept and interviews

In order to assemble and meet with a large enough group of refugees to conduct the survey and interviews, we were dependent on organizations that work and have direct contact with refugees on a daily basis. Their network and contacts were imperative, as attempting to contact refugees personally would be too time consuming, and organizing a meeting in which they could all attend, even more so. Communication with several organizations that personally deal with refugees was maintained throughout the course of this research, and after inquiring about possibilities of organizing such focus groups, INNsatssenteret in Trondheim was seen as best fit. They had the schedule and resources which seemed most likely to allow for the focus groups to be carried out.

Focus groups:

Even with INNsatssenteret's help, organizing the focus groups proved challenging. Because of the nature of their work, which was also the case for all the other organizations we contacted, there were no guarantees of how many participants could be assembled on any given day. This also meant the plan for the various focus groups (men only, women only and mixed group etc.) could not be followed.

In total, four focus groups were interviewed, over the course of three days, with a total of 20 people. Because of the uncertainty of who would be available as focus group participants within a day, all groups became mixed, instead of divided into the ideal focus groups described in the test plan.

The members of the focus groups were from various countries of origin, a mix of male and female and ranged in age from 22 to 43 years old. In total, there were four women and 16 men. The duration of their stay in Norway was on average between 1 and 3 years. Two participants retained work permits, although none had acquired paid jobs.

Getting to know the participants

Before any demonstration of the system was carried out, a short introduction to the system's intent was given to the focus group. Then, time was allotted to get to know the participants and inquire about their current life situation; how they spend their time, what activities they participate in and what tools, if any, they use to find these activities.

All the participants stated that they spent much time on education, being in school to learn Norwegian up to three days a week. A few were involved in work placement two days a week, a sort of unpaid internship. The amount and span of leisure activities the participants partook in varied to a great degree. Those who had children spent most of their free time with them, or doing family related chores. Others spent much of their free time advancing their studies. Other activities were mostly minor and social, such as walks and hikes, arranged with friends face to face or over the phone.

Some, however, were active in trying to find volunteer work, festivals and other happenings in the local community. Usually, they found out about activities through social media, such as Facebook, or even found sites that announce volunteer work or other activities, such as Frivillig.no and trdevents.no. There seemed to be a strong correlation between the participants who were comfortable using technology (computers and smartphones), and those who actively and successfully searched for activities. The majority of the participants were either not comfortable with or skilled enough to use a computer regularly.

These questions were asked to gain a realistic and current insight into the daily life of a refugee in Norway as well as map what tools they were aware of and already comfortable with using. It is important to map their activity level, desire to partake in activities, and their current means of finding such activities in order to properly assess InterConnect's quality and utility.

It was clear among the participants that the goal of their education was to obtain a spot within work placement, a step they have been told is the natural predecessor of paid work. This led to discussions about learning Norwegian, as knowing the language is a requirement for getting one. In addition to there being a limited amount of opportunities for work placement, one was not considered eligible at all without significant and sufficient Norwegian reading and speaking abilities, which several of the participants stated was a factor holding them back from acquiring work. It became clear that the consensus among them was that the education they had access to only got them so far, and that the best way to learn the language is through immersion. They stated that language cafés, where they sit together and speak Norwegian with others, including Norwegians, was currently the best option for this. These are arranged often and frequently, but there was an evident desire for more varied opportunities to speak and interact with Norwegians.

When asked about what tools they currently used to find and join activities, the results were mostly uniform throughout the groups. Most made plans directly with friends by getting in contact with them on the phone, face to face or through social media such as Facebook. Many also used Facebook's event function to find activities in their area. Most of the participants did not seem to attend many events outside their own social circle, such as concerts and exhibitions, and a few even stated that they did not partake in activities outside their own family. This was not due to lack of desire to, but seemed more to be because of time consuming family life and lack of knowledge as to what activities were out there. Few of the activities that were mentioned by the participants when asked to provide examples were under the integrative category, meaning they did not involve speaking Norwegian or meeting new people. Two of the younger participants stated that they used web sites outside of Facebook such as trdevents.no (a site that tries to convey events in and around Trondheim) sporadically to find out what local events were available. Of course, the participants all stated that they found some activities through INNsatssenteret, but as previously mentioned, they are very often the same, such as language cafés.

It became clear through the inquiries of preferred tools for finding activities, that the use of computers is markedly less widespread among the refugees than it is for the average Norwegian. Less than a handful stated that they used a computer (or smartpad etc.) on a regular basis. Even among the younger participants there was a clear discomfort around computer technology. They expressed frustration at how nearly all processes in Norway seem to be centered around and carried out online, such as paying bills, or getting into contact with organizations. Some did not use the computer as a result of difficulty to understand the content of the sites in Norwegian or English, while others seemed simply too unaccustomed to the machines that they stayed away from them entirely. Smartphones, however, seemed to be frequently used by all participants.

Demonstration of InterConnect

After getting to know the participants within the focus groups, a demonstration of InterConnect and its most relevant functionality was executed. As InterConnect is not yet a fully functioning web portal, a demo-site was used which was made with an online demo builder tool called Wix[11], whose layout and user interface can be seen in the Figures 3 to 7 in Chapter 4. The processes that were completed in the demo were as follows:

- Explanation of concept: What it is for and basic functionality
- Examining the initial landing page
- Signing up and creating a user (3 step process)
  - Gathering personal information and activity category preferences
- Home page with overview of available activities and search feature
  - Pointing out the activities that Refugenious has marked as recommended for the user
- Signing up for an activity.

Initial reactions

After the demonstration was concluded, the participants were asked to give their initial reactions to the web portal. As answers, most talked about how they would rate the user interface of the portal, and seemed to agree that it looked good and the design was satisfactory. Some immediately offered to help with the translation work, to get the portal in several languages. The demonstration showed a portal with English at the main language, which after conferring with the participants turned out to be of little consequence as a refugee is as likely to know Norwegian than English.

Were they positive or negative to the concept?

The impression of the web portal across the group was positive. They liked the way it looked, and stated that they thought it could be of use.

More in-depth feedback

Would they use the web portal?

Although generally positive to the idea, the feedback on whether or not they themselves would in fact utilize the web portal varied. Only a few stated that they would definitively use it for finding activities, and most felt it was unlikely. For some, it was due to the fact that they did not like to use computers at all, but even the few computer enthusiasts who also used sites like trdevents.no stated that they probably wouldn't use it. For them, they saw little advantage InterConnect could give them that they could not already find elsewhere.

Then, the recommender system was explained. The participants were told that the web portal's goal was to recommend activities especially for every single user. After some questions from the participants as to how this would work, and explanations of what information would be used to make these recommendations, the majority of the participants found the prospect of access to such a web portal to be of greater interest.

Finally, the participants were asked to assess a set of activities according to their level of interest for them, the results of which will be presented in the next section. After reading through the activities, that were actual activities gathered from various sources, even one participant who initially stated she does not like to use the computer and would not be interested in using such a web portal was exited at the prospect. It seemed as if seeing the kind of activities that are out there and accessible was a new experience for the refugees, and in some cases even ignited new interests in them which they wanted to explore.

[11] http://henriettekopstad.wixsite.com/jibedemo

In the end, the feedback was highly promising. The results of the focus groups bode well for the feasibility and utility of InterConnect as a web portal, and establishes that the integration of a recommender system like Refugenious plays an integral part in contributing to its success.

5.2.2 Technical evaluation

As stated in the plan in Section 5.1, the participants of the focus groups were selected as the test users, and given a set of activities to explicitly state their interest in. However, due to the time limitations and the level of understanding about the test processes among the refugees, the tests had to be simplified slightly, which will be explained in the following sections. Explaining the intended processes and giving explicit instructions to several groups of people with various levels of language skills was time consuming and sometimes confusing for the participants. This also led to some of the data that was gathered being unusable in some or all test cases, because the test user had misinterpreted what information to give.

Activity set

The test users were given a set of 26 activities to deem interesting or not. They were given out in paper form, both in Norwegian and English. The activities were all actual activities that have been gathered from frivillig.no, Røde Kors and Finn.no. Locations and times were omitted for privacy and convenience, as not all of the activities were originally located in Trondheim. It was important to use real activities, as opposed to ideal or fabricated ones. They would not represent real world situations, and it is important for the integrity of the data that the interest shown for the activities is based in reality.

Test users

As stated previously, 20 refugees participated in the focus group. However, due to their own schedules not allowing them to stay for the activity rankings, or because they specified unusable data, activity preference data was ultimately gathered from 13 test users, three women and 10 men.

Execution According to the plan, the test users were supposed to rate all activities on a scale of 0 to 2, where 0 meant not interesting, 1 meant interesting and 2 meant very interesting. Due to the ambiguity of such a scale, the scheme was discarded and replaced with recording only the activities that the test users would want to attend, and asking them to mark three of them as their top choices. This way, the complexity was reduced without removing the possibility of evaluating the system using NDCG.

Results

The test users were given forms to fill out their activity category preferences, which can be seen in Table 6. The leftmost column shows the categories which the users could select from, while the rest of the columns each represent a test user's category preference vector, denoted by the user's ID at the top. For instance one can see that User 1 selected Outdoor activities, Social activities, Art, Literature and Culture, Education, Environment and animals and Technology, while User 12 only selected Sports.

Category	User ID:	1	2	3	4	5	6	7	8	9	10	11	12	13
Outdoor activities		1	1	1	0	0	1	0	0	1	1	0	0	1
Local Community		0	1	1	0	0	0	1	0	0	0	0	0	0
Social activities		1	1	1	1	1	1	0	0	1	1	0	0	1
Art, literature and culture		1	0	1	0	0	1	0	0	0	0	0	0	0
Sports		0	1	1	0	0	1	1	1	1	1	0	0	1
Care and aid		0	1	0	0	0	1	0	0	1	1	0	1	1
Family friendly		0	1	0	1	1	1	1	0	1	1	0	0	1
Education		1	1	1	1	0	1	1	0	1	1	1	0	1
Politics and debate		0	0	0	0	0	0	0	0	1	0	0	0	1
Environment and animals		1	1	0	0	0	1	0	0	1	0	0	0	0
Faith and spirituality		0	0	0	0	1	1	0	0	1	1	0	0	1
Technology		1	0	1	1	0	1	0	0	1	0	1	0	1
Food and cooking		0	1	0	0	1	1	0	0	1	1	0	1	1
Arts and crafts		0	0	0	1	0	0	0	0	0	0	0	0	1

Table 6: Category preferences for each user

After filling out their preferences, the test users were given the 26 activities and asked to mark each one as interesting or not. After reviewing the activities, they were asked to specify which of the chosen activities were their top three choices.

The information about each test user's category preferences and personal information was fed into Refugenious' recommender engine which resulted in the rankings seen in the tables below. The tables, one for every user, show the score of each activity determined by the recommender system, while the rightmost column show the ground truth data, namely the test users actual interest in the activity. Interesting activities are marked with a R for "relevant", and the top 3 activities are marked with an *-symbol by its activity ID. The red line shows the line between the top five activities deemed relevant and returned by Refugenious.

Rank	User 1 ActivityID	Score	
1	1	0.577	R
2	17*	0.548	R
3	11	0.471	R
4	15	0.471	
5	19	0.471	
6	7*	0.408	R
7	10	0.408	
8	14	0.408	
9	22	0.408	R
10	2	0.289	
11	3	0.289	R
12	4	0.289	R
13	5	0.289	
14	0	0.289	R
15	12	0.289	
16	13	0.289	
17	18	0.289	
18	24*	0.289	R
19	25	0.289	
20	26	0.289	R
21	6	0.237	R
22	16	0.236	
23	8	0	
24	20	0	R
25	21	0	R
26	23	0	

Rank	User 2 ActivityID	Score	
1	7	0.667	
2	6	0.577	R
3	11*	0.577	R
4	2	0.471	R
5	3	0.471	R
6	9	0.471	
7	12	0.471	R
8	21	0.471	R
9	24	0.471	
10	24	0.471	R
11	26	0.471	
12	15	0.385	R
13	16	0.385	
14	23	0.333	
15	17	0.298	
16	1*	0.236	R
17	4	0.236	R
18	5	0.236	
19	8	0.236	
20	13	0.236	
21	19	0.193	R
22	10	0	R
23	14	0	R
24	18	0	R
25	20	0	
26	22*	0	R

Rank	User 3 ActivityID	Score	
1	6	0.655	R
2	15*	0.655	R
3	7	0.567	R
4	1	0.535	
5	3	0.535	R
6	9	0.535	R
7	12	0.535	
8	17	0.507	R
9	11	0.436	R
10	16*	0.436	R
11	19	0.436	
12	10	0.378	
13	14	0.378	R
14	22	0.378	
15	2	0.267	
16	4*	0.267	R
17	8	0.267	
18	13	0.267	
19	18	0.267	R
20	21	0.267	
21	24	0.267	R
22	25	0.267	R
23	26	0.267	
24	5	0	
25	20	0	
26	23	0	R

	User 4		
Rank	ActivityID	Score	
1	17	0.8	
2	7	0.671	R
3	1*	0.633	R
4	2	0.633	
5	13	0.633	
6	11	0.516	
7	19	0.516	
8	3	0.316	R
9	4	0.316	
10	5	0.316	
11	9	0.316	R
12	12	0.316	
13	24	0.316	
14	26	0.316	
15	6	0.258	R
16	15*	0.258	R
17	16	0.258	
18	8*	0	R
19	10	0	
20	14	0	R
21	18	0	
22	20	0	R
23	21	0	R
24	22	0	R
25	23	0	
26	25	0	

	User 5		
Rank	ActivityID	Score	
1	2	0.707	R
2	11	0.577	
3	7*	0.5R	
4	3	0.354	
5	4	0.354	
6	9*	0.354	R
7	12	0.354	
8	13	0.354	
9	6*	0.287	R
10	16	0.287	
11	19	0.287	
12	17	0.224	
13	1	0	
14	5	0	
15	8	0	
16	10	0	
17	14	0	
18	15	0	
19	18	0	
20	20	0	
21	21	0	
22	22	0	
23	23	0	
24	24	0	R
25	25	0	
26	26	0	R

	User 6		
Rank	ActivityID	Score	
1	17	0.539	R
2	11*	0.522	R
3	15	0.522	R
4	7	0.452	R
5	1	0.426	R
6	2	0.426	
7	4	0.426	
8	24	0.426	
9	26	0.426	
10	6	0.348	
11	16	0.348	R
12	19	0.348	R
13	10	0.302	
14	14	0.302	
15	22	0.302	
16	23	0.302	
17	3	0.213	R
18	5	0.213	
19	9	0.213	
20	12	0.213	R
21	13	0.213	R
22	18	0.213	R
23	21	0.213	
24	25	0.213	R
25	8	0	
26	20	0	

	User 7		
Rank	ActivityID	Score	
1	7	0.75	
2	6*	0.577	R
3	15	0.577	
4	16	0.577	
5	17	0.447	
6	1	0.354	
7	2	0.354	
8	3	0.354	
9	4	0.354	
10	8	0.354	
11	9	0.354	
12	12	0.354	
13	21	0.354	
14	24	0.354	
15	25	0.354	
16	26	0.354	
17	11	0.289	
18	5	0	
19	10*	0	R
20	13	0	
21	14*	0	R
22	18	0	
23	19	0	
24	20	0	
25	22	0	
26	23	0	

	User 8		
Rank	ActivityID	Score	
1	6	0.	
2	15	0.	
3	1	0.	
4	2	0.	
5	3	0.	
6	4	0.577	
7	5	0.577	
8	7	0	R
9	8	0	
10	9	0	
11	10	0	
12	11	0	
13	12	0	
14	13	0	
15	14	0	
16	16	0	
17	17	0	
18	18	0	
19	19	0	
20	20	0	
21	21	0	
22	22	0	
23	23*	0	R
24	24	0	R
25	25	0	
26	26	0	R

	User 9		
Rank	ActivityID	Score	
1	11	0.522	R
2	7	0.452	R
3	1*	0.426	R
4	2	0.426	
5	24	0.426	R
6	26	0.426	R
7	17	0.405	R
8	6	0.348	R
9	15	0.348	
10	20	0.302	
11	23	0.302	R
12	3*	0.213	R
13	4	0.213	R
14	5	0.213	
15	8	0.213	
16	9	0.213	R
17	12	0.213	
18	13	0.213	
19	18	0.213	
20	21*	0.213	R
21	25	0.213	R
22	16	0.174	
23	19	0.174	
24	10	0	
25	14	0	R
26	22	0	R

Rank	User 10 ActivityID	Score	
1	11*	0.612	R
2	7	0.530	R
3	2	0.5	
4	24	0.5	R
5	26	0.5	
6	6	0.	
7	15*	0.408	R
8	23	0.354	
9	17	0.316	
10	1	0.25	R
11	3	0.25	R
12	4	0.25	R
13	9*	0.25	R
14	12	0.25	
15	13	0.25	
16	21	0.25	R
17	16	0.204	
18	19	0.204	
19	5	0	
20	8	0	
21	10	0	
22	14	0	
23	18	0	
24	20	0	
25	22	0	
26	25	0	R

Rank	User 11 ActivityID	Score	
1	1	1	
2	17	0.633	
3	24*	0.5	R
4	26	0.5	R
5	15	0.408	
6	7	0.354	
7	2	0	
8	3	0	
9	4*	0	R
10	5	0	
11	6*	0	R
12	8	0	R
13	9	0	R
14	10	0	R
15	11	0	R
16	12	0	
17	13	0	
18	14	0	R
19	16	0	
20	18	0	
21	19	0	
22	20	0	
23	21	0	
24	22	0	
25	23	0	
26	25	0	

Rank	User 12 ActivityID	Score	
1	23	0.707	
2	21	0.5	
3	24	0.5	
4	26	0.5	
5	1	0	
6	2	0	
7	3	0	
8	4*	0	R
9	5	0	
10	6*	0	R
11	7	0	R
12	8	0	
13	9	0	R
14	10	0	R
15	11	0	R
16	12	0	
17	13	0	
18	14	0	
19	15	0	
20	16	0	
21	17	0	
22	18	0	
23	19	0	
24	20	0	
25	22	0	
26	25	0	

Rank	User 13 ActivityID	Score	
1	17	0.539	
2	11	0.522	
3	7	0.452	
4	1	0.426	
5	2	0.426	
6	13	0.426	
7	24	0.426	
8	26	0.426	
9	6*	0.348	R
10	15	0.348	
11	19	0.348	
12	20	0.302	
13	23	0.302	R
14	3	0.213	
15	4	0.213	
16	5	0.213	
17	8*	0.213	R
18	9	0.213	
19	12	0.213	
20	18	0.213	
21	21	0.213	
22	16	0.174	
23	10	0	
24	14	0	
25	22	0	
26	25*	0	R

With this data cataloged, computations of precision, recall and NDCG can be computed. Precision is arguably the most important measure for Refugenious as it describes the fraction of relevant items the system recommends to the user. The ultimate goal of Refugenious is to function as a way for the refugee users to obtain relevant activities without having to manually find them. As stated in Chapter 2, precision is calculated using the formula:

$$P = \frac{tp}{tp + fp}$$

where *tp* is the number of recommended items that are deemed relevant by the test user, while *fp* denotes the amount of recommended items that the test user marked as irrelevant.

Recall describes the fraction of relevant activities that were recommended to the user.

$$R = \frac{tp}{tp + fn}$$

where *fn* denotes the activities the test user found relevant but were not recommended by the system. These results will aid in determining whether k = 5 is the right choice.

Normalized Discounted Cumulative Gain is an indication of the quality of the ranking algorithm.

$$NDCG = \frac{DCG}{iDCG}$$

where iDCG denotes the ideal score that could be obtained with the same activities, and DCG:

$$DCG = \sum_{i=1}^{m} \frac{rel_i}{log_2(i+1)}$$

where rel_i is the relevance score given by the test user to the activity ranked i^{th} by the system, m is the total number of rated items.

The calculations are made for each test user's results and are shown in the table below.

User ID	Precision	Recall	NDCG
1	0.6	0.231	0.792
2	0.8	0.267	0.711
3	0.8	0.286	0.819
4	0.4	0.182	0.621
5	0.4	0.333	0.688
6	1	0.417	0.863
7	0.2	0.333	0.51
8	0	0	0.369
9	0.8	0.267	0.807
10	0.6	0.3	0.831
11	0.4	0.222	0.605
12	0	0	0.464
13	0	0	0.376
Average	**0.462**	**0.218**	**0.651**

Table 7: Precision, recall and NDCG scores for each test user's result

Immediately, three test users' results stand out, user 8, user 12 and user 13, whose actual results vary so greatly to Refugenious' recommendations that they scored a 0 on precision and recall. These three users alone drag the average precision score from 0.6 to 0.462. A look at their category preferences in Table 6 and comparing that to their activity ratings yields some explanation to their low scores. User 8 and 12 have in common that they had very few category preferences and that these did not match their activity ratings. User 8 only checked off *Sports*, but showed interest for activities within the *Social activities* and *Care and aid* categories. User 12 only entered two categories as her preferences, *Food and cooking* and *Care and aid*, while 4/6 of the activities she deemed interesting were activities that were predominantly under the *Social activities* category. User 13 made the opposite choice to check nearly all the categories (11/14) as preferences while only showing actual interest in 4 activities.

Although important to think about when creating a recommender system, to sanction oneself from such extremes is difficult and impractical to attempt. One possibility is that certain category names need further development that may make them easier to understand. For example, it was a recurring event that users did not specify *Care and aid* as a category preference, but seemed interested in activities that had to do with helping with the care and company of elders, families or underprivileged kids. This includes user 8.

As previously stated, high precision is imperative for Refugenious. Although not great, the average score is respectable and Table 7 shows that there are several scores equal to or higher than 0.8. As this is only the first iteration, before the system has gotten a chance to learn the users actual preferences from usage data etc, these are scores that give an optimistic view on Refugenious' recommendation quality.

In addition, 10 of 13 users were recommended at least one of their top choices within the top 3 activity recommendations that were given to the user by Refugenious. Unsurprisingly, the users who did not receive such recommendations were user 8, 12 and 13. As the main goal of Refugenious was to serve as an easy way to get started and engaged in activities, these are promising results.

The recall scores are consistently quite low, which could indicate that k should be increased from 5. However, the precision scores are not high enough to support this, as a higher number of returned recommended activities is likely to lower its score.

The NDCG scores are mostly high. The average score of 0.651 indicates that the ranking algorithm is doing an adequate job at sorting the activities by the user's interest. Not counting users 8, 12 and 13 gives an average NDCG score of 0.725, which is very high, considering this is the first iteration.

6 Evaluation of research process and focus

This chapter discusses the quality and suitability of the methods and approach chosen to obtain the findings of the paper's research. The approach will be evaluated in light of the original problem the thesis was meant to solve.

6.1 Research focus and chosen approach

The problem presented as the main motivation for this thesis is the passivization that affects refugees that arrive in Norway due to the long wait to get applications approved and get integrated into their new society. The approach chosen for solving this problem is deploying a recommender system into a web portal with an overview of activities, that filters out irrelevant items and provides the most suitable activities for the individual refugee user.

This approach was chosen on the premise that recommender systems are suitable for problems related to helping users find relevant information, and that this could be transferred to refugees finding activities in a way that lowers the threshold for participation. However, aspects of the research show signs of the project's scope being too small to reap the real benefits of the extensive computational power of the state of the art tools and technologies of recommender systems.

After reviewing numerous articles that might be of relevance to the project, a majority did not fit the scope of the thesis due to the complexity of the proposed systems and technologies. Research within the area of recommender systems seem to revolve around conquering the massive amount of information about people and habits etc. that have become accessible due to the rise of social media and information on online behavior. Complex techniques and methods, such as an Artificial Immune System [14], taxonomy [15] and predictive analytics and personality traits [16], find ways to handle the workload of processing this data in order to improve recommendations further.

A goal of this paper's project is simplicity, and to forego the need for extensive input of data from the system's users, making it plausible that the scope of the project made it less prone to take full advantage of recommender system technologies. Arguably, the fault tolerance regarding the recommendations is not as low for Refugenious and InterConnect as for other systems, because it is a unique concept that is wanted, and does not need instant commercial success. For systems such as Netflix, imprecise recommendations can cost them their customers to rivaling sites like Amazon or HBO.

However, the perceived focus on complexity in the reviewed articles on state of the art recommender systems could be attributed to the chosen research domain. The vast majority of the articles that were considered were regarding job recommendation, as it seemed a natural area to explore due to the parallels that could be drawn between finding jobs and finding activities for a user. Particularly the activity *volunteer work* concretised this notion. Within the area of job recommendation, extensive research has been done, and social media and professional online networks such as LinkedIn.com have created possibilities for obtaining large amounts of personalized data, such as profiles, user interactions etc. With such an ongoing increase in research possibilities and data sources, it is not surprising that newer research articles focus on harnessing this information through powerful techniques and complex algorithms.

Broadening the research area for this thesis might have provided more suitable techniques and tools for the project scope. However, recommender systems are relevant to so many areas of research, on anything from movies, to restaurants, to social connections, so narrowing down the field of research was imperative in order to stay within the time limitations of the project. No research directly related to the project was found, and job recommendation was then chosen as a focus area.

Complexity of the system aside, the simplicity of having relevant activities clearly displayed and easily accessible, is undoubtedly an advantage InterConnect would benefit from in regards to its goal of lowering the threshold for participation among refugees. Therefore, the result of the research and proposed system of this paper will be implemented and tested as a part of a master thesis this coming year.

6.2 Inclusion of stakeholders

A majority of the investigation into the current situation and needs of refugees in Norway was done prior to defining the problem to be solved by this paper. Official documents regarding this topic were reviewed and quantitative research was done through surveys and interviews at reception centers for refugees and aid organizations. The investigation consistently supported evidence of the negative impact the long waiting period has on refugees' quality of life, especially due to the lack of activities to pass time with. From the time the project was defined and the research initiated, however, the direct contact with refugees was lessened. The technical aspects, as well as those pertaining to the web portals UI, were decided without direct feedback from or in cooperation with the refugees.

On the other hand, the time limitations and amount of work to actualize the web portal as well as decide on a design approach for Refugenious did not make this feasible. Instead, contact with organizations such as INNsatssenteret in Trondheim was maintained, that have the sole purpose to work toward getting refugees into work or education. A complete list of organizations and entities contacted regarding the requirements and needs of organization and refugee users can be found Appendix A. INNsatssenteret reported that the web portal and recommendation service was a solution refugees would likely use, which they would help market to the refugees of their programs. These organizations have a clear notion of the wants and needs of refugees, and their input shaped parts of the design process.

In addition, time to establish and continue communication with the organizations responsible for *posting* activities needed to be prioritized. Contrary to the refugees, these organizations do not have obvious incentives to adopt the portal with its recommendation service instead of continuing with their old solutions. Some used their own web pages to display information, while others sent information directly to the refugee centers and organized participation through their administration.

It was imperative to get these actors enthusiastic about the project, and the time spent surveying them about what functionality they desired in order to use the proposed system resulted in defined and clear requirements, as well as considerable interest in the project. It is because of this communication that InterConnect, along with Refugenious, is being launched within 2017.

7 Conclusion and Future Work

7.1 Conclusion

This thesis investigated the possibilities of using a recommender system on a web portal to aid refugees in finding useful pastime activities and aid with integration into society in Norway. Recent research on relevant recommendation methods, as well as functionality on existing related systems led to the conclusion that accuracy and high quality of recommendations often came at a price of complex computations and need for user supplied data on a scale that compromised the intended simplicity of the web portal. The recommender system made for this purpose therefore centered around the basic principles of content-based recommender systems and resulted in Refugenious, a system where simple information about users' interests is the main basis for recommendations.

The system was then evaluated on a technical and non-technical basis. The information gathered from focus group interviews showed positivity towards such a concept, and it became clear that there was a desire to have access to more activities. According to the refugees themselves, activities that involved interacting and talking with other refugees and Norwegians helped their language and social skills, thereby increasing their chances at finding a job. The web portal concept seemed to engage the participants' enthusiasm for joining in activities. The possibility of personalized recommendations set the activity portal apart from other tools known to the refugees.

The technical evaluation yielded hopeful results regarding the system's recommendation quality, especially seeing as tests only revolve around a first iteration of recommendations, without a chance for the system to take learned user preferences into account. More than 75% of the participants were recommended one of the activities they were most interested in within the top three recommended activities. The NDCG scores also showed that the overall ranking algorithm of Refugenious was working well, despite requiring only a small amount of data from the users to do so.

Going back to the original research question of this thesis, the results of Refugenious' concept and recommendation evaluation indicate that a recommender system can be a powerful tool for aiding in refugee integration. It is a way to lower the threshold for activity participation, simplifying the process of finding and joining activities as well as helping the refugees to find new interests. Although recommendations can be improved by acquiring more information about its users, Refugenious yields great results with only basic personal information and data regarding their activity category preferences.

7.2 Future work

The immediate future goal of this project is to launch the web portal and integrate the already made recommender system Refugenious. Due to the feedback from the refugees in the focus groups, it seems fitting to include work on a smart phone application with the same functionality, as the comfort level with smart phones far exceeded that of computers.

A long term goal of Refugenious is to extend its functionality towards job recommendation as the web portal grows in popularity. The proposed solution may be qualified to handle odd jobs such as painting fences etc. as these jobs, too, can be categorized by domain to explain content. However, jobs that require experience and education need a different classification scheme to be effective.

Another possible venture for Refugenious is to transition into a hybrid system, that combines the proposed content-based system with item-based CF. Collaborative filtering performs well, and the combination of these two methods will let one approach weigh up for the faults in the other, and vice versa. CF will discourage the overspecialization of the content-based component and provide novelty in the recommendations, while the content-based methodology will counteract the cold-start and data sparsity problem.

The hope for the future of this projects is to further its work to help engage and integrate refugees in Norway. The research and results of this thesis shows that simple steps can make a big difference, and that an research within this area is new, exciting and yields interesting results.

8 References

[1] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[2] Robin Burke. The adaptive web. chapter Hybrid Web Recommender Systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.

[3] Robin Burke. Knowledge-based recommender systems.

[4] Iman Avazpour, Teerat Pitakrat, Lars Grunske, and John Grundy. Dimensions and metrics for evaluating recommendation systems. In *Recommendation Systems in Software Engineering*. 2014.

[5] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu Tie-YanLiu, and Wei Chen. A Theoretical Analysis of NDCG Type Ranking Measures. 2013.

[6] Yingya Zhang, Cheng Yang, and Zhixiang Niu. A Research of Job Recommendation System Based on Collaborative Filtering.

[7] Sang Min Choi and Yo Sub Han. A content recommendation system based on category correlations. In *Proceedings - 5th International Multi-Conference on Computing in the Global Information Technology, ICCGI 2010*, 2010.

[8] Mamadou Diaby, Emmanuel Viennet, and Tristan Launay. Toward the Next Generation of Recruitment Tools: An Online Social Network-based Job Recommender System.

[9] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features.

[10] Nikolaos D Almalis, George A Tsihrintzis, Nikolaos Karagiannis, and Aggeliki D Strati. FoDRA -A New Content-Based Job Recommendation Algorithm for Job Seeking and Recruiting.

[11] Yao Lu, Sandy El Helou, and Denis Gillet. A Recommender System for Job Seeking and Recruiting Website.

[12] Marcel Caraciolo, Bruno Melo, and Ricardo Caspirro. Crab: A Recommendation Engine Framework for Python. *PROC. OF THE 10th PYTHON IN SCIENCE CONF*, 2011.

[13] Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich Recommender, and Robin Burke. Book Review. *Intl. Journal of Human–Computer Interaction*, 28:72–73, 2012.

[14] Shaha T Al-Otaibi and Mourad Ykhlef. An Artificial Immune System for Job Recommendation.

[15] Mamadou Diaby and Emmanuel Viennet. Taxonomy-based job recommender systems on Facebook and LinkedIn profiles. In *Proceedings - International Conference on Research Challenges in Information Science*, 2014.

[16] Imran Ali Mirza, Shainila Mulla, Rishit Parekh, Satej Sawant, and Krishna Mohan Singh. Generating Personalized Job Role Recommendations for the IT Sector through Predictive Analytics and Personality Traits. pages 4–6, 2015.

Appendix A

Complete list of contacted entities regarding organizations' and refugees' requirements and needs:

- INNsats center in Trondheim
- Midtbyen Management
- Innovasjon Norge (Innovation Norway)
- Integrerings- og mangfoldsdirektoratet (IMDi), Integration and Diversity Directorate
- Bergen kommune
- Ytre Arna reception center
- Åsane reception center
- Landås reception center
- Røde Kors Bergen (Red Cross Bergen)
- Røde Kors Oslo (Red Cross Oslo)
- Redd Barna Region Vest
- Kirkens Bymisjon Bergen
- GiveAJob Oslo