University of Cologne

Faculty of Management, Economics and Social Sciences

Department of Information Systems and Information Management

Prof. Dr. Detlef Schoder

Recommender System for Personalized Spoken-Word Radio

submitted by Tim Ockenga

supervised by

Dr. Stefan Hirschmeier

Cologne, 02.08.2021

Table of Contents

List of Figures

List of Tables

Abbreviations

CB	Content-based
CBOW	Continuous Bag of Words
CF	Collaborative filtering
DSR	Design Science Research
IS	Information Systems
NLP	Natural Language Processing
QCA	Qualitative Content Analysis
RBA	Radio Broadcasting Agency
RS	Recommender System
TF-IDF	Term Frequency, Inverse Document Frequency
W2V	Word2Vec

1 Introduction

Digitalization affects radio broadcasting in the same way as it used to influence other media sectors before. Traditional media services are increasingly being replaced by innovative, non-linear media services where consumers can choose when, where and on which device they want to consume media content. While the film, music, and news industries have already made great strides towards a digital transformation, the radio industry is only slowly getting off the ground (Hirschmeier et al., 2019). This is mainly due to the fact that Radio Broadcasting Agencies (RBAs) have in the past been subject to less innovation pressure than other media sectors. Today, with the increasing use of media streaming services, consumer expectations of the services offered have changed. Large media companies have implemented and established innovative technologies such as Recommender Systems (RSs) in their services, which consumers today take for granted. In the second quarter of 2021, Spotify reached 365 million active users with their music streaming service (Rabe, 2021). Since the consumers' time is limited, RBAs compete vigorously with these companies for their listeners' attention. Traditional linear radio cannot meet these new expectations (Hirschmeier et al., 2019), and redistributing the massive amount of original data on new platforms such as mobile devices or smart speakers is not sufficient. According to Davidson et al. (2010), personalization is critical in the success of content delivery services. Once the consumer uses the service, it is crucial to recommend relevant content that keeps her interested and decreases the bounce rate. RSs that deliver suitable suggestions increase user satisfaction and improve user loyalty (Aggarwal, 2016, p. 4). To truly leverage the value of their content, RBAs must therefore provide it through services that offer personalized recommendations. Enormous RS research has already drastically improved music (Spotify) and film (Netflix) streaming services, not least through initiatives such as the Netflix Prize, driven by commercial interests (Koren, 2009). RSs are typically designed to provide recommendations for a specific type of content, e.g., books, music, movies (Ricci et al., 2015, p. 18). The design of a RS strongly depends on both the particular characteristics of the content to be recommended and the domain (Melville & Sindhwani, 2011, p. 829). Therefore, different algorithms perform better or worse on different data sets (Herlocker et al., 2004). As radio content differs from music and film content, the capabilities to deliver relevant content to radio listeners of existing recommender approaches need to be examined separately. However, radio has received little attention in RS research, and many RBAs are struggling to apply this technology to develop and launch personalized services (Hirschmeier et al., 2019). Therefore, we pose the following research question:

> *How to design different recommender touchpoints for different spoken-word radio situations?*

The primary objective of RBAs is to increase both user engagement with their services and retention time. While public service broadcasters try to retain their listeners with their high-quality content in order to fulfill their public-service remit, private broadcasters do so to serve their business model, e.g., advertising or guide users towards a subscription. To achieve this, the listener must first be directed to the platform on which the content is offered. Then the system must enable her to find interesting content with minimum effort and as quickly as possible. After the listener has finished consuming an item, the system must offer additional interesting content that she subsequently wants to listen to. This study focuses on the last two steps by examining how different recommender touchpoints can suggest spoken-word radio content that is relevant to listeners and creates the need to consume more content.

To address the research question of this study, we conduct one cycle of a design science research (DSR) study to develop an initial set of design principles that concern a class of RSs solving the problem at hand. The objectives of our research are threefold. First, we aim to identify design principles for RSs suggesting spoken-word radio content for different listening situations. We derive the design principles based on knowledge from theoretical (literature) and non-theoretical (RBA experts) sources. Second, we develop a fit-for-purpose IT artifact according to those principles. Third, we assess and revise the resulting design requirements and principles by evaluating the artifact through a user study.

Our goal is to make a level one design science contribution, as defined by Gregor and Hevner (2013) and to provide a foundation for future research to develop higher level contributions. Although our design science research study develops design principles for a certain class of RSs within the substantive context of spoken-word radio, we also hope to contribute to the understanding of how a broader class of

RSs might be designed. From a practical perspective, the results of our study can help RBAs develop new competitive services that provide users with personalized recommendations for spoken-word content. This allows them to respond to the changed expectations of their listeners and broadcast their high-quality content themselves in the long term. In this way, they can avoid acting only as producers of their content and open up the possibility of competing with new innovative media services.

We position this study in the field of design science research. Unlike the natural and social sciences, which aim to describe, explain, and predict, DSR aims to solve certain problems and provide prescriptive knowledge that captures how instances of a particular class of artifacts should be developed (Hevner et al., 2004).

Following a design science approach, the remainder of this work is structured as follows. In the second chapter, we introduce the problem domain by highlighting spoken-word radio, recommender systems, and natural language processing. In chapter three, we clarify the overall methodology of this paper. Next, in chapter four, we formulate design requirements as they emerge from current research and practice. Additionally, we derive design principles based on existing spoken-word radio and RS research addressing the design requirements. We then map the design principles to design features and conceptualize recommender touchpoints to be implemented in a prototype. Thereafter, chapter five contains detailed descriptions and explanations of the implementation of the prototype and the underlying data. In chapter six, we evaluate the prototype before presenting the resulting findings in chapter seven. Finally, we discuss the contributions of this work, present its limitations, and offer suggestions for future research.

2 Background

In this chapter, we give a theoretical background on spoken-word radio and explain its current role and development within spoken-word audio. Additionally, we introduce RSs and link them to spoken-word radio. Finally, we briefly introduce natural language processing, which plays an essential role in recommender systems for textual content.

2.1 Spoken-word Radio

In the following, we define spoken-word radio content, explain the differences to other media content and introduce its current role in the field of spoken-word audio.

We define *spoken-word radio* as textual content in audio format which is produced and distributed by radio broadcasters. Typical formats comprise news, interviews, dialogues, discussions, documentaries, running commentaries, book reviews, quiz shows, talks, and more. In addition to diversification in formats, there is also diversification in topics (e.g., science, politics, sports, economics, etc.). Furthermore, the content varies in depth from light, entertaining to serious, in-depth features and in duration from a few minutes to more than an hour. It can be both mono- and multi-thematic and differs significantly in some characteristics from other media content such as music or movies. While topicality plays a minor role in the consumption of music and movies (old songs and movies can be as relevant as new ones), it is often a decisive factor for the relevance of spoken-word radio content. For example, for news content, which is ranked first among the most popular spoken-word audio topics (Edison Research & NPR, 2020), consumers show more interest in current news than in events further back in time (Montes García et al., 2013). Another feature is that spoken-word content, unlike music, often becomes irrelevant immediately after consumption. Consumers tend to listen to the same song every day but do not listen to the same interview more than once. Spoken-word radio is part of spoken-word audio, which also includes podcasts (e.g., meaning podcasts, conversational podcasts etc.), and audiobooks. There has been remarkable growth in spoken-word audio in recent years, according to The Spoken-word Report (2020), from NPR and Edison Research, which examines listener behaviors and preferences related to spoken-word audio in the US. As for listening to audio content in general, it has steadily taken share away from the ever-popular music. However, this growth is entirely due to the growing popularity of podcasts and audiobooks, which are mainly distributed via platforms such as Apple Podcasts, Spotify, Google Podcasts, Stitcher, and Tuneln that offer personalized recommendations. In fact, spoken-word radio, like music, has also lost its share to podcasts and audiobooks. Perks and Turner (2018) found that people commonly use podcasts as a good replacement for radio. This highlights the urgency with which RBAs should respond to changes in the market to retain their share. Regarding the question of why people listen to spoken-word content, a study by Chan-Olmsted and Wang (2020) uncovered the following motives:

- *Information*: the users' need to stay informed about the world around them and learn new things
- *Entertainment*: the users' desire to enjoy and relax
- *Escapism*: the users' need to escape from the presence and to pass the time
- *Social interaction*: the desire to use spoken-word audio as a means for social interaction (e.g., talking with others about specific content)
- *Identification*: the user's need to identify with someone/something
- *Companion*: spoken-word content can act as a companion (e.g., learn about others)
- *Audio platform superiority*: listeners desire on-demand convenience in terms of time and place, as well as content variety and uniqueness

Furthermore, people use spoken-word content as an accompaniment to other activities (Krause, 2020). In terms of the usage situations of spoken-word audio, Mai et al. (2019) conducted a study in which 380 individuals who regularly listen to podcasts were asked in which situations they listen to them. They found that participants listen to podcasts at least once a week in the following situations:

- Doing housework 42%
- Driving in the car 34%
- Without doing anything else 34%
- Traveling in public service 28%
- Doing sports 24%
- Going to sleep 22%

2.2 Recommender Systems

Today, RSs permeate many areas of our daily lives, such as music recommendations (Schedl & Hauger, 2015), movie recommendations (Chen et al., 2015), news and document recommendations (Turcotte et al., 2015; Wintrode et al., 2015). Burke (2002, p. 331) describes RSs as "[...] any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options". While this definition implies personalization Poriya et al. (2014) state that there are also non-personalized RSs, which represent the simplest type of RSs. Non-personalized RSs provide the same recommendations to all users. In the case of spoken-word radio, the recommendations can be chosen manually by the online editorial office, based on current trends such as the latest features on Covid-19 or the N most popular items. Since RBAs already manage to implement this type of recommendation, this study will exclusively examine RSs, which provide personalized suggestions for each user based on her preferences. We define Recommender Systems as software tools that help their users interact with large, complex information spaces by pre-filtering and suggesting a subset of items that are most likely to be relevant for them (Burke et al., 2011; Ricci et al., 2015, p. 1). The term "item" refers to what a RS suggests to its users. According to (Aggarwal, 2016, p. 3), the recommendation problem can be divided into two different versions. First, the prediction version, which is to predict the rating value for a user-item combination. Second, the ranking version, which is to recommend the top-k items for a specific user. Recommendations are calculated based on assumptions the system makes about the user. To generate these assumptions, the system collects information on the user's preferences for a set of visited items and generates a user model (also user profile) which encodes her preferences and needs. These preferences can be viewed as ratings that are implicitly derived from the user's interactions with the system (implicit ratings) or explicitly given by the user (explicit ratings) (Aggarwal, 2016, p. 11). While implicit ratings do not require the additional active participation of the user in the rating process, explicit ratings do. Commonly implemented manifestations of explicit ratings are binary ratings, where users only express that they like or dislike an item, and unary ratings. Unary ratings only allow users to express interest in an item and do not provide mechanisms to reject content.

Based on how RSs filter content, they are typically categorized into three main categories: Collaborative Filtering (CF), Content-based Filtering (CBF), and Hybrid RSs. Due to differences in the underlying concepts of these approaches, the respective systems in these categories have different advantages and disadvantages (Aggarwal, 2016, p. 15). In the following, we will introduce all three types and explain their strengths and weaknesses.

Collaborative Filtering RSs leverage the power of user-submitted ratings to recommend relevant items to users. They suggest items that other users with similar tastes and preferences liked in the past. The underlying reasoning is that for a user who has agreed with certain users in the past, other items that these similar users have liked should be both relevant and of interest to the user (Ricci et al., 2015, p. 2). In the implementation of this approach, the user profile is represented by a simple list containing the ratings implicitly or explicitly given by the user for certain elements. Typically, collaborative filtering uses one of two types of methods, namely memory-based methods and model-based methods:

Memory-based methods (or neighborhood-based methods) predict the ratings of user-item combinations based on their proximity to other users or items (Aggarwal, 2016, p. 9). Depending on whether the algorithm uses proximity to other users or proximity to other items for this prediction, a distinction is made between user-based- and item-based collaborative filtering.

- *User-based collaborative filtering* identifies users similar to a target user A by comparing A's ratings of items with other users' ratings of the same items. The system then predicts ratings for the unseen items of A by computing weighted averages of the ratings of the identified peer group. In other words, these algorithms make predictions about user A's opinion of an item by combining the opinions of other like-minded users.
- *Item-based collaborative filtering* identifies a set S of items that are most similar to a target item B. To predict whether user A will like item B, the system uses the ratings given by A to the set S.

In terms of choosing one method over the other, user-based approaches tend to perform better in terms of originality of recommendations, which can lead to higher user satisfaction (Ekstrand et al., 2014). Item-based methods, on the other hand, provide higher recommendation accuracy and are more efficient when the number of users exceeds the number of items (Ricci et al., 2015, p. 12). They are therefore to be preferred in the corresponding scenarios.

The great advantages of memory-based methods are the simplicity of both the implementation and the explanation of the recommendations provided (Aggarwal, 2016, p. 9). Nevertheless, they suffer from at least two issues. First, a sparsity problem, i.e., they do not work very well with sparse rating matrices. Second, a scalability problem, i.e., they do not scale well to commercial recommender systems (Mobasher, 2006; Xue et al., 2005).

Model-based methods have been proposed to alleviate sparsity and scalability problems (Gong et al., 2009). These methods generate recommendations by developing a model from user ratings. To build a prediction model, they use machine learning algorithms such as Bayesian network, latent factor models, clustering, and rule-based approaches (Huang & Ouyang, 2007; Sarwar et al., 2000; B. Schafer et al., 2001; Zhong & Li, 2010). Unlike memory-based methods, model-based methods cluster different users into a small number of classes based on their rating patterns. Disadvantages of model-based over memory-based approaches are the computational power required to build and update the model and that they are limited in terms of the number of distinct users covered.

In general, CF has two major strengths, which are the simplicity of implementation and the explainability of the provided recommendations. Nevertheless, CF methods suffer a severe problem, the so-called *cold-start-problem* (Adomavicius & Tuzhilin, 2005). Since CF RSs suggest items based on ratings, they can either make no recommendations or only low-quality recommendations in the absence of ratings. In situations where no ratings are initially available, it is therefore not possible to make reliable recommendations. There are three distinct situations in which the cold-start problem occurs. First, when there is a new community (Lam et al., 2008). This refers to the challenges of obtaining sufficient ratings to make reliable recommendations while launching a RS. Second, when there is a new item (Park & Tuzhilin, 2008). Items that are newly added to the system have a low probability of being recommended as they have not yet been rated by any user. This leads to fewer ratings of these items by the users since they do not see them. The result is a negative feedback loop in which some items never enter the rating process and are never recommended. One solution to this problem is to provide users with alternative ways to discover new content. The third problem is one of the biggest challenges of RS already in operation and occurs when there is a new user. (Rashid et al., 2008). RSs cannot make personalized suggestions to new users who have not rated any items yet. If the user has rated only a few items, the reliability of CF-based recommendations is still not satisfactory, as these systems improve with the number of ratings provided by the user. This may cause the user to stop using the system due to dissatisfaction because her expectations are not met.

Content-based Filtering RSs suggest items that are similar to the ones the user liked in the past. The system determines the similarity of the items based on their attributes. The underlying rationale is that the same user will rate items with similar attributes similarly (M. Pazzani, 1998). In the case of spoken-word radio, attributes could be the topic, duration, or format of a feature, et cetera. For example, if a user has liked several radio features that deal with science and are shorter than 10 minutes, the system can learn to suggest other short features from that topic. To obtain the required attributes of the content to be recommended, they must be extracted from the items (J. Pazzani & Billsus, 2007). Generally, there are two procedures for this. In the first one, the system developers manually determine a set of attributes considering the content domain. The second covers specific cases, e.g., when textual information is to be recommended. Since computers cannot interpret or understand raw texts, the information must first be converted into a format that machines can process. In such cases, developers implement classical *Natural Language Processing* (NLP) techniques for information retrieval such as *Term Frequency, Inverse Document Frequency* (TF-IDF) that automatically define the attributes.

Content-based methods offer some advantages with respect to the recommendation of new items when there is a lack of ratings for these items (Aggarwal, 2016, p. 15). This means that CB methods can help mitigate the problems that occur with the cold-start-problem of a new, unrated item. Since these methods do not depend on sufficient ratings, they are able to suggest new unrated articles as soon as they enter the system. Nevertheless, CB methods entail two challenging problems. First, automatic extraction of the required attributes is often difficult and provides unreliable results for satisfactory recommendations. Second, CB methods tend to suffer a problem called *overspecialization*. This term refers to a well-known phenomenon in which the system does not recommend any diverse items but only items with a narrow focus (Ning et al., 2015, p. 38). As a result, this can lead the user to be captured in a so-called *filter bubble* in which the algorithms effectively isolate her from a diversity of viewpoints or content (Pariser, 2011). This poses a problem since diversity influences the satisfaction of the users with the RS (Hurley & Zhang, 2011).

Hybrid approach RSs combine multiple filtering approaches, e.g., a combination of CF and CB filtering (Choi et al., 2012). The aim is to take advantage of each of the combined techniques. Hybrid approaches can help to alleviate the cold-start problem (Kim et al., 2010; Weng et al., 2008).

2.3 Natural Language Processing

Natural Language Processing (NLP) is a subfield of artificial intelligence and linguistics that deals with getting computers to process the statements or words written in human languages (Nadkarni et al., 2011). The aim is to achieve human-like language processing for a range of tasks or applications (Liddy, 2001). This means that, except for some AI researchers, NLP is not usually considered a stand-alone goal. It is rather the means to achieve certain tasks. The task in this study is to extract the textual attributes from the radio features and convert the information into a format that a RS (a computer) can process.

The representation of words and documents is an essential part of most NLP tasks. Words are typically represented by so-called *word embeddings*. (Almeida & Xexéo, 2019). This term refers to an approach that aims to create a real-valued vector representation of words to capture their meaning. Typically, vectors encode the meaning of words in such a way that words closer together in the vector space are expected to be similar in meaning. For instance, the vector of the word "guitar" would be close to the vector of the word "drums" but far from the vector of the word "astronaut". Moreover, word embeddings can capture further semantic regularities (Mikolov et al., 2010), e.g., male-female or singular-plural. These relations correspond to arithmetic operations other than just calculating the distance between two words, such as adding or subtracting word vectors.

There are several ways of generating word embeddings. Mikolov et al. (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013) introduced *word2vec*, a procedure that is based on a two-layer recurrent neural network consisting of two different models. First, *continuous bag-of-words* (CBOW) and second *skip-gram*. While the CBOW model predicts a word from a given set of words surrounding it, the Skip Gram model works in exactly the opposite way. It predicts a set of words that surround a particular word. The input to a word2vec model is a corpus of text, and the output is a vector representation of each word. Figure 2-1 illustrates the architecture of the two models.

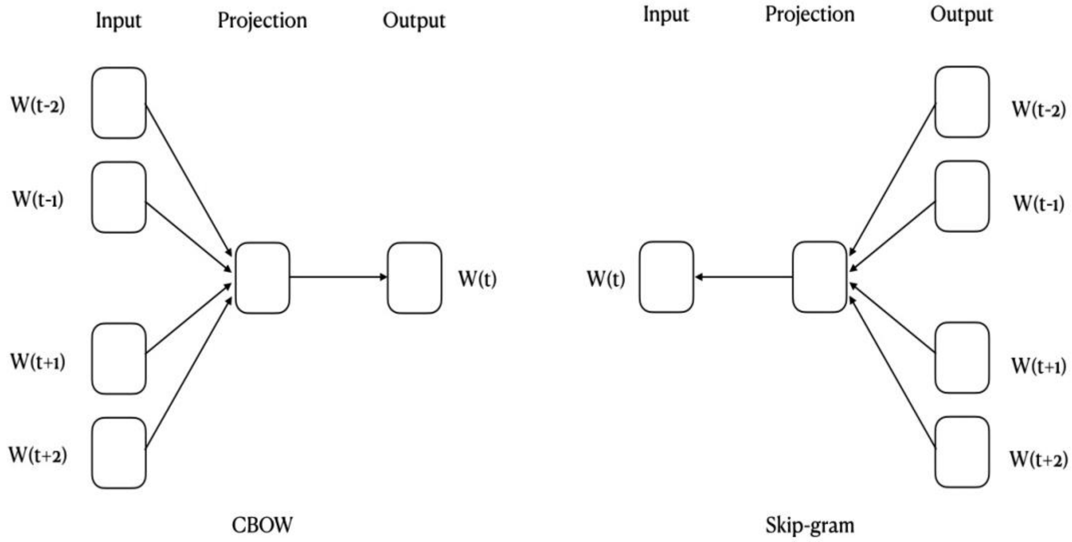

Figure 2-1: W2V CBOW and Skip-Gram architecture (Mikolov, Chen, et al., 2013)

There are also other approaches for the generation of word embeddings that do not rely on neural networks and embedding layers, such as the GloVe model (Pennington et al., 2014). Both types of models rely on the distributional hypothesis by Harris (1954) that is the assumption that words with similar contexts have the same meaning. Based on this assumption, Almeida and Xexéo (2019) define word embeddings as "[...] dense, distributed, fixed-length word vectors, built using word co-occurrence statistics as per the distributional hypothesis".

Since many information retrieval tasks from textual content require more than simply processing the meaning of single words, there are also *document embeddings*. Parallel to the goal of word representations, document embeddings are intended to capture the essential meaning of a document in a format that a computer can process. To overcome the weaknesses of Harris' (1954) bag-of-words approach to generate document embeddings, Le and Mikolov (2014) proposed *paragraph vectors*, also called *doc2vec*. Paragraph vectors are an extension of word2vec that extends the learning of word embeddings to sequences of words. It is an unsupervised framework that generates continuous, distributed vector representations for variable-length texts. The text lengths it can adequately represent range from sentences to large documents.

Another approach to generating document embeddings that stands out for its simplicity is word vector averaging. It generates document embeddings by simply computing the average of the vectors of the words that make up the document. Therefore, its implementation is simple. Word vector averaging works well for short texts that cover one topic but poorly for large texts dealing with multiple topics (Lau & Baldwin, 2016). A major drawback of this approach is that, like standard bag-of-words models, it does not take word order into account.

3 Methodology

The objective of this study is to examine how to design different recommender touchpoints for different spoken-word radio situations. To address this question, our study follows the design science research paradigm. Design Science is a well-established research approach in Information Systems (IS) and has gained renewed importance in recent years. It is a paradigm that focuses on solving problems by developing a new, innovative artifact and applying it in a natural environment (Baskerville, 2008). Many studies have proposed approaches that provide guidance to DSR practitioners on how to conduct DSR projects (Hevner, 2007; Iivari, 2015; Peffers et al., 2007; Sein et al., 2011). All these approaches build upon iterative cycles (essentially: design, build-artifact, evaluation) with continuous reflection and incremental refinement. A new cycle begins under consideration of the learnings from the evaluation of the artifact in the previous cycle. The output of design science research is a fit-for-purpose IT artifact that solves a critical organizational problem in a satisfactory manner (Hevner et al., 2004). It is important to effectively describe it in order to enable its implementation, deployment, and use in an appropriate domain. The artifact contains conceptual knowledge about how to solve the practical problem it is intended to address, also referred to as *design principles* (DPs) (Markus et al., 2002; Pries-Heje & Baskerville, 2008). DPs are derived from the artifact. They capture information about how to develop other instances of artifacts belonging to the same class (Baskerville et al., 2009). Baskerville & Pries-Heje (2010) describe design principles as statements that form the basis for actions. More specifically, they guide or constrain the actions (Hevner & Chatterjee, 2010).

Our research design follows the approach proposed by Vaishnavi and Kuechler (2007). Figure 3-1 depicts the high-level research process of this study. As a first step, we reviewed literature and discussed with experts from our collaboration partner to create awareness of the practical problem at hand. Based on this, we formulated design requirements capturing criteria that the artifact should meet. In the suggestions stage, we used knowledge from our literature review to derive design principles that address the formulated requirements. Next, we conceptualized design features according to the design principles. We then implemented these features in the development phase. Finally, we evaluated the developed artifact by presenting it to users and exploratively investigating its capabilities to recommend relevant content in different situations. In this study, we conduct one design cycle. The results provide a level one design science contribution as defined by Gregor and Hevner (2013). To provide a higher-level contribution, further research can build upon our work and conduct additional design cycles.

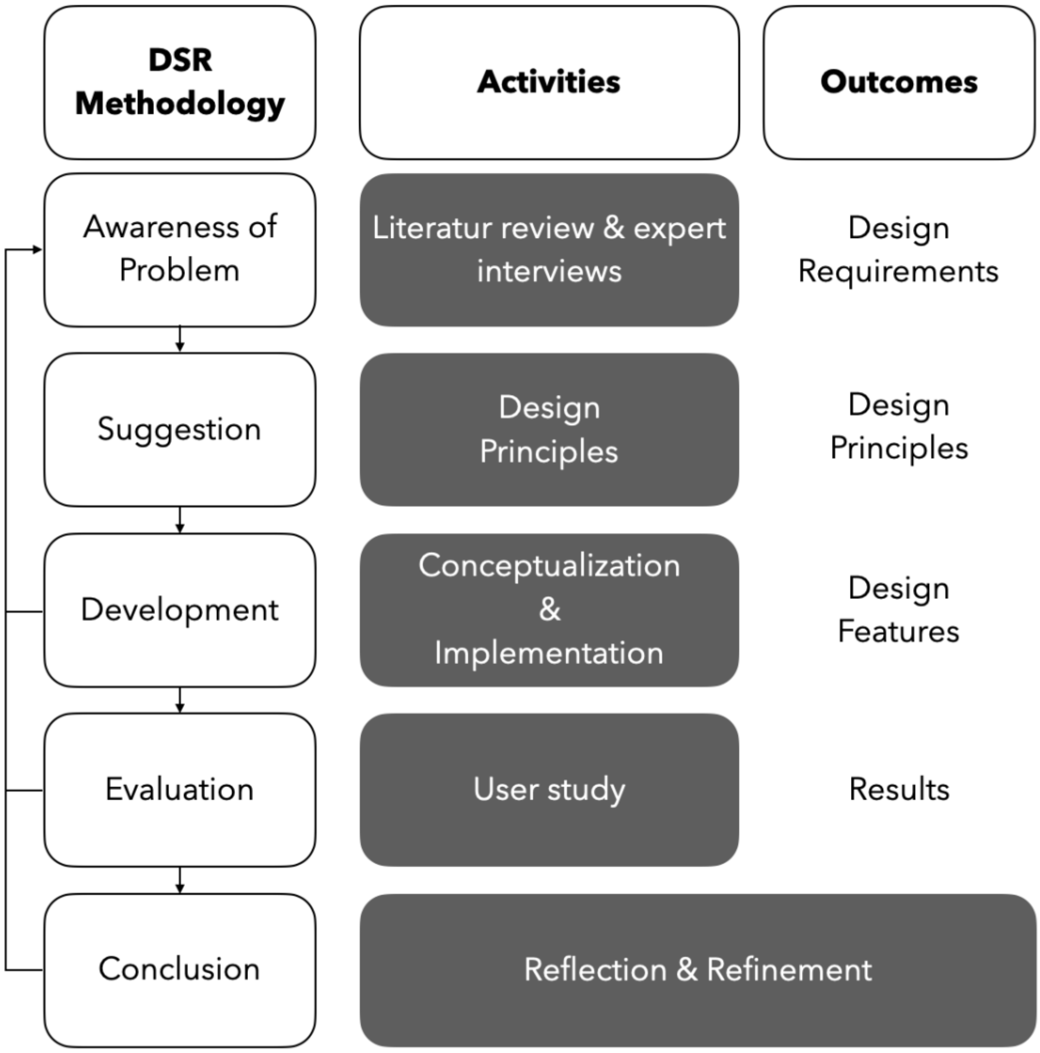

Figure 3-1: Design science research approach of this study

4 Design Requirements, Principles & Features

In this chapter, we formulate design requirements, derive design principles from them, and explain the underlying conceptualization of the implemented prototype (the instantiation).

4.1 Formulation of Design Requirements

In this section, we discuss the requirements for an artifact consisting of different recommender touchpoints that provide the user with relevant spoken-word radio content suggestions for different situations. We derive these requirements mainly from the purpose and scope and refine them through literature research and discussions with a major German public radio broadcasting agency.

We start with the identification of stakeholders and derive the design requirements based on their positions for spoken-word radio. According to Hirschmeier and Melsbach (2019), there are the following three stakeholder groups that have an interest in the design of non-linear radio (consisting of publicly funded spoken-word radio).

- **Radio Broadcasting Agencies** who produce and disseminate the content
- **Listeners** who consume the content
- **The Public** having different requirements for the distribution of the content

RBAs aim to provide relevant content to their listeners in order to increase their engagement with the radio service. Listeners themselves enjoy easy access to relevant content, as this reduces their cognitive efforts in searching for it. Therefore, a meta-requirement that recommender touchpoints should fulfill is the provision of relevant content that matches the listeners' interests. One way to accomplish this is to implement recommender algorithms that predict items most likely to be relevant to the listener. Regarding the choice of the RS type, the advantages and drawbacks must be weighed, taking into account the application domain. Since topicality plays a major role in spoken-word radio (see Section 2.1), it is crucial that the system is capable of suggesting the latest items that have been recently added to the media library.

DR1: Implement recommender algorithms that are capable of suggesting new items.

The meaning of what constitutes interesting content for the listener varies depending on the current situation and motives for listening to spoken-word content. Since it is challenging to predict a listener's current consumption motive, other media companies, such as Spotify or Netflix, offer their content through a series of recommender touchpoints that address different situational needs. In this way, providers let listeners reveal their current motives themselves and can deliver relevant content for different needs by offering recommender touchpoints for a range of motives. To the best of our knowledge, there is no research on the motives of people listening to spoken-word radio content. However, podcasts are very similar to spoken-word radio content and are regarded as a good replacement for it (Perks & Turner, 2018). Therefore, we use Chan-Olmsted and Wang's (2020) findings on the consumption motives of podcast listeners (see Section 2.1) to derive the following design requirements.

As in other areas of human life, the social aspect is important in spoken-word radio content. People listen to it in order to share moments with their friends and family and to socialize by talking with others about it.

DR2: Provide content that other like-minded listeners have listened to.

One main motive is that people have a need to stay informed about things happening in the world around them and to learn new things.

DR3: Provide content that informs about things currently happening in the world, e.g., news.

DR4: Provide content that deals with topics in an educational manner.

Entertainment is another typical motive for people to listen to spoken-word content. Listeners enjoy it, and it helps them to relax.

DR5: Provide content that is entertaining and easy to digest.

Another social motive is that people listen to spoken-word content as a companion to escape loneliness, usually by learning about other people's lives.

DR6: Provide content that brings the listener closer to the lives of others.

Additionally, spoken-word content helps people identify with themselves, the community, or certain values.

DR7: Provide content that helps listeners to fulfill or reinforce personal and communal identity or value.

People consume spoken-word content to pass the time and escape from the presence.

DR8: Provide content that enables the listener to escape from the presence.

Since listeners are not always aware of what their current motives are for consuming media content, a non-linear service should also provide the ability to browse for interesting content regardless of the motive.

DR9: Provide relevant content in general, regardless of meeting specific motives or other specifications.

Recommending related items is one of the most widely implemented recommendation strategies to suggest relevant content to the user (Agarwal et al., 2013; J. B. Schafer et al., 1999). For instance, YouTube shows related videos next to the one currently playing, keeping users engaged with their content for hours. A recent study revealed that users prefer related item recommendations over both per-genre recommendations and overall recommendations (Yao & Harper, 2018).

DR10: Provide content that is related to the one the user is currently consuming.

In the case of publicly funded radio, a third stakeholder group is the public, which has an interest in ensuring that the requirements of the public-service remit are met. Discussions with our collaboration partner revealed that the following two requirements are relevant for a non-linear content consumption service.

DR11: Provide content that includes information, education, culture, and entertainment.

DR12: Provide the content in a balanced manner.

These design requirements represent an initial set that enables us to start the first design cycle. After completing this cycle, we adjust and refine them based on the results of the evaluation phase. This will lay the foundation for further circles in which the design requirements and principles can be sufficiently elaborated.

4.2 Derivation of Design Principles

In order to meet the design requirements presented in the previous section, we derived design principles explaining how to implement an appropriate solution.

DP1: Use content-based filtering algorithms to recommend content for which topicality is a key determinant of relevance.

Section 2.2 reveals that collaborative filtering algorithms suffer from the cold-start problem, especially with respect to new items. When using such algorithms, new items have a low probability of being recommended because they have not yet been rated by any user. In contrast, content-based filtering algorithms do not rely on user ratings and can recommend new items immediately after they are added to the media library. DP1, therefore, addresses DR1. If the available data includes teaser texts, these should be used to calculate document embeddings, as they adequately summarize the main content of the radio features. Otherwise, one can use attributes such as the broadcast, the duration, or the topic of the radio feature.

DP2: Implement a recommender touchpoint that uses collaborative filtering algorithms to suggest content that other like-minded listeners have listened to.

Since the strength of collaborative filtering RSs is to identify users who have similar interests to a particular user and suggest items those users have consumed, they should be used to address the social interaction motive (DR2).

DP3: Implement recommender touchpoints for all consumption motives by filtering for broadcasts that address one or multiple compatible motives.

Each broadcast usually follows a certain format (e.g., news, interviews, dialogues), deals with certain topics (e.g., politics, sports, culture), is characterized by a certain depth (e.g., entertaining, or serious), and has a fixed duration (e.g., 5 minutes or 1 hour). Depending on the characteristics of a broadcast, it appeals better or worse to a certain consumption motive. For example, a broadcast that deals with in-depth scientific news is more likely to appeal to the need for information and education than to the need for entertainment. Therefore, broadcasts can be assigned to specific consumption motives. Based on this assignment, one can filter the media library for broadcasts addressing a specific motive and sort the resulting features by relevance to the user. Consumption motives include information (news and education), entertainment, escapism, companion, and identification, as presented in 2.2 and explained in chapter 3. In addition, we expand these motives with cultural content to address the public interest (DR11). Thus, DP3 addresses all design requirements from DR3 to DR8 and DR11. Furthermore, it meets DR12 by ensuring balanced content delivery through the implementation of different touchpoints that provide diverse content.

DP4: Implement a recommender touchpoint that suggests content similar to what the user has liked in the past based on content-based filtering.

This touchpoint allows users to browse content that is likely to be relevant regardless of current motivations, thus addressing DR9. Since the content should not be limited by any constraints, topicality is relevant for certain items (see Section 2.1). Therefore, the use of content-based filtering algorithms should be preferred.

DP5: Implement a recommender touchpoint that suggests content related to the item the user is consuming based on content-based filtering that leverages descriptive texts.

Yao and Harper (2018) conducted a large study to evaluate different recommender algorithms for the computation of related items from a user-centric perspective. They found content-based filtering to outperform all other methods. Additionally, their results reveal that using texts that describe the content (e.g., content summaries) produces better results than using tags (e.g., genre, topic, duration). DP5 addresses DR10.

4.3 Conceptualization of Design Features

In this section, we map the derived design principles from the previous section to design features. *Design features* refer to specific functionalities of the artifact that serve to fulfill the design principles (e.g., the recommender algorithm for related item computation). Figure 4-1 depicts the design features of the artifact we used to develop our prototype.

To satisfy the design principles derived in the previous section, three main components must be implemented. First, a content-based filtering algorithm to address DP1, DP4, and DP5. Second, a collaborative filtering algorithm to address DP2. Third, a filtering component that filters the content to be recommended according to specific consumption motives, targeting DP3. These three components allow us to implement different recommender touchpoints such that all requirements are satisfied. In the following, we conceptualize each component and explain how to use them to implement the required touchpoints.

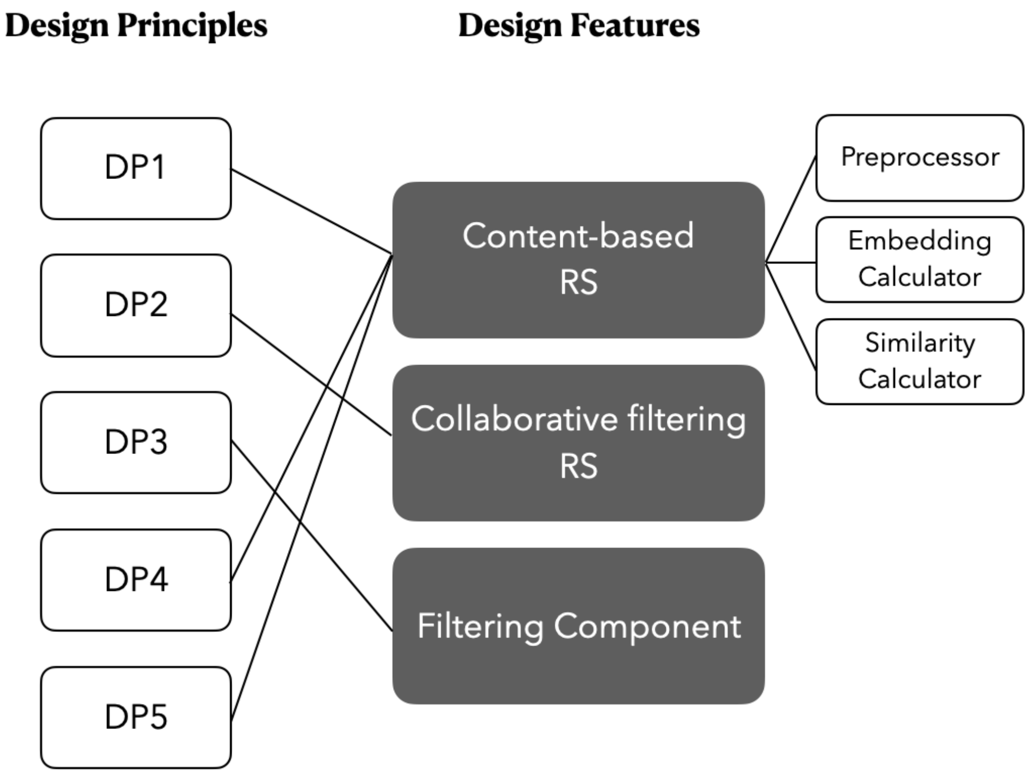

Figure 4-1: Mapping design principles to design features

DF1: Content-based Recommender System

To address DP1, DP4, and DP5, one needs to implement a content-based Filtering RS that predicts similarity between items and returns related items to a target one. CB RSs calculate similarity based on document embeddings representing the items' content. To generate the document embeddings, texts should be preferred over tags to satisfy DP5. Two types of texts are feasible for spoken-word content. Full speech-to-text transcriptions of the features' audio tracks, or short texts that briefly summarize the features’ content, such as teaser texts. To convert these texts into document embeddings, two steps must be performed. First, the texts are preprocessed by applying NLP techniques. Second, the resulting preprocessed texts are used to compute real-valued vector representations. Thus, DF1 requires the following two sub-components:

DF1.1 Preprocessor

The preprocessor component automatically converts the input texts into a structured format that the system needs for further text mining processes. The aim is to optimize preprocessing such that the subsequent processes can be implemented in the best possible way. Therefore, preprocessing should be optimized to compute optimal similarity scores for predicting relevant content.

DF1.2 Embedding Calculator

The embedding calculator component takes the output of the preprocessor as input to generate document embeddings for each text/item. As described in 2.2, there are different approaches to compute document embeddings from texts. As with the preprocessor component, the choice of method depends on how well it contributes to the goal of generating optimal similarity scores for predicting relevant content.

Finally, to use document embeddings to calculate the similarity between elements, another subcomponent is required.

DF1.3 Similarity Calculator

This component implements the functionality to efficiently compute similarities between a target item and a set of comparison items. It outputs a list containing these comparison items sorted by similarity in descending order.

DF2: Collaborative Filtering RS

To satisfy DP2, one must implement a collaborative filtering RS that suggests unknown items that like-minded users have rated positively. The system identifies similar users based on a user-item matrix and suggests items based on what other people with similar consumption patterns appreciated. It is also conceivable to implement an item-based CF RS, which calculates the similarity between items by checking how many users that bought item A also bought item B (Linden et al., 2001).

DF3: Filtering component

The filter component provides the functionality to filter the content to be recommended in such a way that it appeals to specific consumption motives, thus addressing DP3. This is implemented by filtering the content for broadcasts assigned to the motive to be addressed. The assignment is done manually by the developers. In order to simplify the decision of assignment, this process can be done using description texts of the broadcasts, if they are available. It should be noted that some broadcasts might not clearly address any motive while others address multiple motives. Furthermore, it can be useful to cover several motives with one touchpoint, e.g., one touchpoint that addresses the two motives entertainment, and escapism.

Based on these three components, we implement nine different recommender touchpoints in our prototype such that all requirements are met. Below, we introduce these touchpoints and explain the role of the three design features in generating the respective content.

My interests

This touchpoint delivers content that matches the listener's interests in general. The suggested content is not constrained by any limitations, such as specifications on topics or consumption motives. That is, any content can be recommended as long as it is relevant to the listener. DF1 is used to generate the respective recommendations. For each item that the user has liked in the past, the CB RS returns the most similar items that are unknown to the user and concatenates them into a list. This list is then randomly shuffled and displayed to the user.

Related items

The CB RS (DF1) is also used to suggest related items, i.e., items that are similar to the one the listener is currently listening to. The system takes the current item as input and returns items related to it.

What others listened to

This touchpoint suggests unseen content that other like-minded users have rated positively. Recommendations are provided by the CF RS (DF2) based on explicit and implicit user ratings collected during the operation of the system.

Six touchpoints for addressing consumption motives

Using the filtering component (DF3) and the CB RS (DF1), we implement six recommender touchpoints addressing all identified consumption motives. The two motives entertainment, and escapism are addressed via the same touchpoint since one can assume that entertaining content also helps consumers to escape from the presence. Consequently, we implement the following six touchpoints:

- **Information** - content that informs about recent events
- **Education** - content that educates the listener
- **Culture** - content related to culture, e.g., presentations of cultural events or biographies of artists
- **Insights** - content that provides insight into the lives of others
- **Entertainment & Escapism** – light, entertaining content that helps to escape from the presence
- **Society & Values** - content that provides insights into the lives of others

Generating recommendations for one of these touchpoints requires two steps. First, the filtering component uses the previously defined assignments of broadcasts and motives and filters only for broadcasts that match the respective motive. Second, the CB RS compares the items resulting from the filtering process with items that the user has liked in the past and predicts their relevance to the user. The system then suggests the most relevant items.

5 Implementation

In this chapter, we describe the technical implementation of the design features. First, we explain the structure of the database, and how we obtain data. Additionally, we provide descriptions of this data. Second, we present the implementation of the design features, which are divided in three components. Third, we explain how these components are used to provide content suggestions for nine different recommender touchpoints. Finally, we give an overview about the basic functionalities and the interface of the prototype.

5.1 Description of the Data

In this section, we describe in detail the data used in our prototype, how it is retrieved and stored. We implemented a database consisting of three tables. First, an “items” table containing data related to the content to be recommended. Second, a “users” table storing data related to the system's users. Third, a “ratings” table capturing how users rate specific items. While the last two tables are filled during operation, the items table must be set up in advance. This process involves retrieving raw data and cleansing it so that the resulting dataset contains only instances with all the necessary information for further processing.

Items table

Since this study was conducted in collaboration with a major German public RBA, we were given access to their media library via an API to retrieve data from all the radio features they publish online. We downloaded the data using a python script and saved it as a .csv file. Each row of the resulting dataset represents a single item (radio feature) composed of various attributes. Next, we cleaned the raw data by removing all rows that were missing the information needed for further processing. Table 1 provides an overview of the required attributed. The resulting dataset comprises more than 24K items in the German language with a variety of topics and radio formats (e.g., interview, news, dialogue).

Table 1: Description of items table

Attribute	**Data type**	**Description**	**Example**
audio_id	Integer	A unique number that serves as the primary key.	921398
title	String	Represents the title of the radio feature.	'Where does green hydrogen come from - coast or desert?'
authors	String	Represents the name(s) of the author(s).	'Ockenga, Tim'
teaser	String	A brief description (approx. 50 words) of the feature's content.	'Green, regeneratively produced hydrogen - that is the vision in which billions are to be invested. But how? Production in Germany, for example, with large wind farms on the coast? Or imported from sunny regions like the Sahara? In any case, huge quantities are needed.'
broadcast	String	The name of the broadcast the feature belongs to.	'Science in focus'
duration	Integer	The total duration of the audio in seconds.	1789.0
date	Datetime	Timestamp of the publication date. (01/2020 - 08/2021)	2021-08-09 00:00:00
audio_path	String	URL of the audio resource.	'https://mp3path.mp3'
image_small	String	URL of the small image resource.	'https://smallImagePath.jpg'
image_large	String	URL of the large image resource.	'https://largeImagePath.jpg'

Users table

The “users” table stores data related to the users of the system, as presented in Table 2. Each time an unknown client connects to the system for the first time, a new entry is made in this table.

Table 2: Description of the users table

Attribute	Data type	Description	Example
user_id	Integer	A unique number that serves as the primary key.	38024273

Ratings table

The “ratings” table stores all data related to user interactions indicating how users rate certain items. This data is produced during operation. Table 3 provides an overview of the data collected.

Table 3: Description of items table

Attribute	Data type	Description	Example
rating_id	Integer	A unique number that serves as the primary key.	15
date	Datetime	Represents the title of the radio feature.	2021-05-26 11:32:40.939267
user_id	Integer	A unique number that serves as a foreign key and refers to the users table.	38024273
audio_id	Integer	A unique number that serves as a foreign key and refers to the items table.	921398
rating	String	A word that indicates the rating.	'like

5.2 Implementation of Design Features

In this section, we describe the technical implementation of the design features and recommender touchpoints in detail in order to enable its implementation, deployment, and use in an appropriate domain.

1) Content-based Recommender System (DF1)

In the following, we describe our implementation of the CB RS that computes the similarity between items and explain the underlying decisions in detail. This includes preprocessing text data, computing document embeddings, and predicting items related to a target one.

As described in 2.2, CB RSs generate recommendations based on attributes of the content to be suggested. We decided to use the teaser texts of the radio features as a basis for this CB RS. Since online editors write these texts, they are of high quality and adequately summarize the content of the radio features. Thus, for our purposes, they are superior to full audio transcriptions, which capture semi-structured conversations with a lot of irrelevant information partially.

Preprocessor (DF1.1)

In order to convert the teaser texts into a structured format that the system needs for further text mining processes (e.g., the generation of document embeddings), it preprocesses them. The preprocessing of data often strongly impacts a model's performance (Camacho-Collados & Pilevar, 2018). In turn, the optimal preprocessing steps depend on both the domain and the underlying algorithms of the model. This means that two different models with the same prediction task may require completely different preprocessing steps. Thus, the search for optimal preprocessing is partly a trial-and-error process. Therefore, we tested different combinations of preprocessing steps and approaches for the document embedding generation with the aim of adequately predicting the similarity of two items. We ended up with the following preprocessing steps:

1. **Lemmatizing**: All words are transformed into their canonical form, their word root, or lemma. For example, the word "playing" would be transformed into "play" or the word "guitars" to "guitar". This step helps

the algorithms to identify the same words; otherwise, the same word in different forms might be interpreted as different words.

2. **Stop-word Removal**: English, as well as German stop-words are removed. Stop-words are words such as "this, is, at, the etc.". Those words do not add significant meaning to the sentences and only add noise to the models.
3. **Lowercasing**: All words are cast to lower-case. Without this procedure, words that are the same but capitalized (e.g., because they are at the beginning of the sentence) and non-capitalized in another place would not be recognized as the same word.
4. **Remove** all non-alphabetic characters: The documents contain many symbols such as numbers, punctuations, and line breaks. Those added noise and were therefore removed.
5. **Remove all non-German accents**: All non-German accents are removed (e.g., "à" is transformed to "a" and "ê" is transformed to "e").
6. **Tokenization**: Each text document is split into single terms using the state-of-the-art library "NLTK" (Bird et al., 2009).
7. **n-grams**: Regarding word types, we use bigrams, ignoring all words and bigrams with a total collected count lower than 30.

In addition to these basic preprocessing steps, we implemented another step that is specific to the given prediction task. We filtered the text only for nouns and pronouns, assuming that the content of the items is significantly represented by them and that the other word types do not add value to the prediction task. We found that in our data, this greatly improves the performance in terms of identifying the most similar documents to a target one.

Embedding calculation (DF1.2)

Regarding the approach to generate the document embeddings from the preprocessed teaser texts, we tested four different models, as shown in Table 5. The first three are word embedding models that require a further step to obtain document embeddings. The system uses them to calculate the document embeddings by applying word vector averaging over their word embeddings. That is, it takes the component-wise mean of its component word embeddings. In other words, the document embeddings are the average of the word embeddings of all nouns and pronouns in the corresponding teaser. The doc2vec model does not require any further steps but provides document embeddings directly.

Since the evaluation of the models is based on similarity scores between documents, we first describe the subsequent steps before explaining the evaluation and selection of the models. Once the system terminates calculating the document embeddings for all items in the dataset, it uses them to predict the ten most similar items for a target one. For this process, we chose a batch-oriented precomputation approach which extends the items table of our database with a list of the ids of the ten most similar items for each item (Table 4).

Table 4: Description of most_similar_ids attribute

Attribute	**Data type**	**Description**	**Example**
most_similar_ids	List of Integers	A list of Integers containing the ids of the ten most similar items.	[921398, 846513, 678491, 451637, 988672, 845769, 536667, 899742, 7648859, 687712]

This brings some advantages over on-demand calculation of the most similar items. It enables both access to large amounts of data with sufficient CPU resources during recommendation generation and delivery of the pre-generated suggestions with extremely low latency. On the other hand, a major drawback of this approach is the delay between the generation and deployment of a given recommendation dataset. To mitigate this downside, we set the time interval for recommendation generation updates to one day.

Similarity computation (DF1.3)

In the precomputation phase, the system computes the pairwise similarity of each document with all other documents using the cosine similarity defined as follows:

$$similarity = \cos(\theta) = \frac{A * B}{||A|| \, ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

A and B are the document embeddings of two teaser texts, each representing the content of a radio feature. The value range of the cosine similarity is from -1 to 1. A value of 1 indicates a perfect relationship between two documents which means that the content is completely the same. This case occurs only when one and the same document is compared with itself. A value of -1 indicates a perfectly negative relationship between two documents. This means that the two documents under consideration deal with completely different content. In conclusion, one can interpret the cosine similarity of two documents as follows. The closer the cosine similarity to 1, the more similar the documents are. We implemented the calculation of the cosine similarity using the scipy library (SciPy community, 2021).

Based on the prediction of the most similar items and the corresponding similarity scores, we evaluated the four different models as shown in Table 5 using two evaluation metrics. The first metric measures how well a model predicts the exact same document as being most similar to itself. To calculate this metric, we computed the total number of times the model predicted the same item as being most similar to itself for 1000 items and divided this number by 1000. The second metric proved more meaningful but required human input. Since we use unsupervised approaches for the predicting task at hand, objective evaluation and selection of models are challenging (Wallach et al., 2009). In contrast to supervised settings, it is not possible to simply calculate standard metrics such as accuracy, precision, or recall. As suggested in other NLP studies, we also measured the performance of the models using a direct human evaluation (Chang et al., 2009). For this, three subjects rated the five most similar items predicted by the model for 30 given items based on the teaser texts. In doing so, the subjects marked those items for which they felt that they were similar to the target item. On this basis, we calculated the relative proportion of predicted items that the subjects found to fit the target ones. The results clearly show the superiority of the word vector averaging approaches over the doc2vec model in the task at hand. While all models perform well in predicting the same item to be the most similar to itself, the human evaluation shows differences between them in predicting similar items to a target one. Subjects perceived only 26.7 percent of the items predicted by the doc2vec as similar to the corresponding given item. In contrast, the word vector averaging approach based on the pre-trained word2vec model achieved the best score of 93.3 percent of the predicted items being considered to match the content of the reference item. Therefore, we choose to implement the word vector averaging approach based on the pre-trained word2vec model as the method for generating the document embeddings.

Table 5: Evaluation of word embedding models

Model	**Training dataset**	**total words in vocabulary**	**Predicted input as most similar**	**Human evaluation**
Pretrained word2vec	German Wikipedia	3.591.515	1.0	0.933
word2vec	Our dataset	4.374	0.995	0.846
pretrained FastText	German Wikipedia	2.000.000	0.988	0.873
doc2vec	Our dataset	12.213	0.965	0.267

2) Collaborative Filtering Recommender System

As explained in section 2.2, model-based collaborative filtering RSs provide recommendations by developing a model from user ratings. Our prototype collects both explicit and implicit user ratings and stores them in a database table called "ratings" (see Section 5.1). Since the rating values are stored as strings such that humans can understand them easily, they must first be converted into a machine-processable format. To do this, the system assigns real-valued numbers to all unique ratings and replaces them respectively. Table 6 shows the applied mapping from strings to real-valued numbers. It then uses the attributes "user_id", "audio_id" and "rating" to build a user-item matrix. Leveraging this matrix, the system trains a model based on SVD as a learning technique. We implemented this using the surprise library by Hug (2020). The model takes a user id as input and returns a dictionary of audio ids and similarity scores. In addition, two parameters can be set to determine how many recommendations should be returned, as well as a minimum value for the similarity.

Table 6: Mapping of rating strings to real-values number ratings

Rating (string)	**Rating (real-values number)**
,dislike '	1
,skipped '	2
,listened_through '	3
,like '	4

3) Filtering component (DF3)

To implement the filtering component, we manually assigned broadcasts to the consumption motives information, education, identification, entertainment and escapism, companion, and culture. For this, we went through descriptive texts of 73 different broadcasts, which were provided by our collaboration partner, and decided whether and which motives were clearly addressed by the respective broadcast. Although we ended up assigning a set of broadcasts to each motive, there is an imbalance in the number of broadcasts assigned to each motive. While 15 broadcasts were assigned to the information motive, only three were assigned to the companion motive. Furthermore, we did not assign 19 of the broadcasts since we found them not to clearly address any of the motives. We then hardcoded the resulting assignment in the component's code and implemented a function that takes a motive, filters the content of the media library for the appropriate broadcasts, and returns a list of the respective audio ids. Additionally, the maximum time delta between the current date and the publication date of the content can be set.

5.3 Recommender Touchpoints

This section explains the implementation of the recommender touchpoints and introduces their representation as well as the basic functionalities of the prototype.

Related items

This touchpoint leverages the precomputed most similar items of the CB RS (DF1) to provide recommendations to the user. It suggests the most similar items to the one the user is currently listening to. Since the system can simply retrieve the audio ids of these items from the database, no further calculation steps are necessary here. Unlike the other touchpoints, recommendations are not filtered for unseen items. Therefore, it may display items that the user already knows.

My interests

In contrast, to the previous touchpoint, this one requires additional processing steps in order to generate recommendations. The suggestions are based on items the user has liked in the past. For each liked item, the system requests the ten most similar items from the database and appends them to a list. Then it removes all items from the list the user already listened to or disliked in the past. In a final step, the system randomly shuffles this list of recommendations before providing it to the user.

What others listened to

The recommendations of this touchpoint are directly provided by the CF RS (DF2). Unlike the previous two touchpoints, the suggestions are not calculated in advance but during operation. As model-based approaches require notably computational power to build and update the model (see Section 2.2), this might be an issue in large-scale commercial applications. However, this implementation fulfills the purposes of our study.

Touchpoints for addressing consumption motives

Using the filtering component (DF3) and the CB RS (DF2), we implemented six touchpoints for addressing the consumption motives according to the conceptualization in Section 4.2. First, the filtering component filters for content that addresses the corresponding motive and was published in the specified time frame. For the touchpoint addressing the information motive, we set the maximum time delta to

14 days such that it only suggests informative content that was published within the last two weeks. The content of all other touchpoints was not time-constrained. Second, the CB RS predicts the relevance of the items resulting from the filtering process by comparing them with the items the user has liked in the past. For this, the system loops through all filtered items and calculates the similarity between the filtered item and all items the user has liked in the past. It then stores the maximum similarity score for that item, which indicates how relevant the item is to the user. After finishing this process, the touchpoint suggests the items that have received the highest similarity scores.

In addition to the described recommender touchpoints, we implemented another touchpoint called "Latest Features" that provides the latest content to enable users to initially discover and like the content. This was necessary since all recommender touchpoints rely on user likes and do not provide suggestions for users who have not rated items so far.

5.4 User Interface & Basic Functionalities

To implement the frontend of our prototype, we used Angular, which is a popular framework for the development of client-side JavaScript single-page web, mobile web, native mobile, and native desktop applications. Angular code is written in TypeScript, which is a superset of JavaScript. Since Angular is platform-independent, our prototype can be used with different devices. This is an advantage in terms of conducting user studies as the prototype can be used on any device of the participants. Angular's component-based architecture brings another advantage. It simplifies the implementation of future extensions and changes. This is of great importance in DSR since incremental refinements are continuously implemented. In the following, we describe the user interface of the recommender touchpoints and introduce the basic functionalities of our prototype.

Figure 5-1 shows the recommender touchpoint overview on the left and the recommender playlist on the right. When using our prototype, users first land on the overview. This view lists the names of all implemented recommender touchpoints (except for the "related items" touchpoint), including brief descriptions of the content that each touchpoint provides. Clicking on a touchpoint takes the user to the recommender playlist, which lists the suggestions of the selected touchpoint. Again, the user can select (load and play) the item of her choice by simply clicking on it.

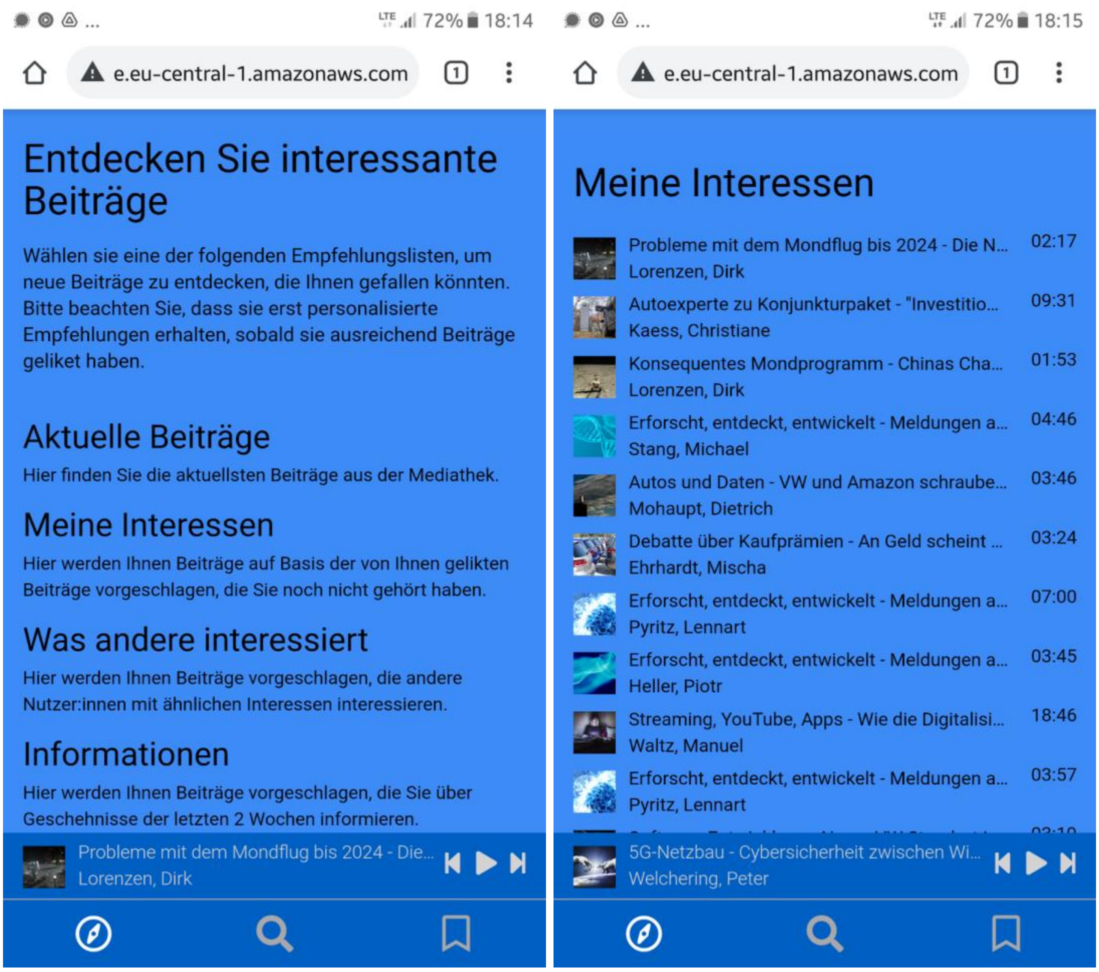

Figure 5-1: Touchpoint overview (left), recommender playlist (right)

In addition to these features, our prototype implements further functionalities of standard applications for the consumption of audio content accessible through the audio player view in the expanded state, shown in Figure 5-2 on the upper left. If this view is not expanded, as shown in Figure 5-1, the user can expand it by clicking on the image, title, or author of the current item. In the expanded state of the audio player view, the user cannot only play/pause, skip forward and backward the current item but also like or dislike it and jump to a specific timestamp. Additionally, the user can access the "related items" recommender touchpoint by clicking on the "Similar" tab, located on the bottom right under the like button. The view then displays a list of ten items that are related to the one currently playing. Like before, the user can select and play an item by clicking on it.

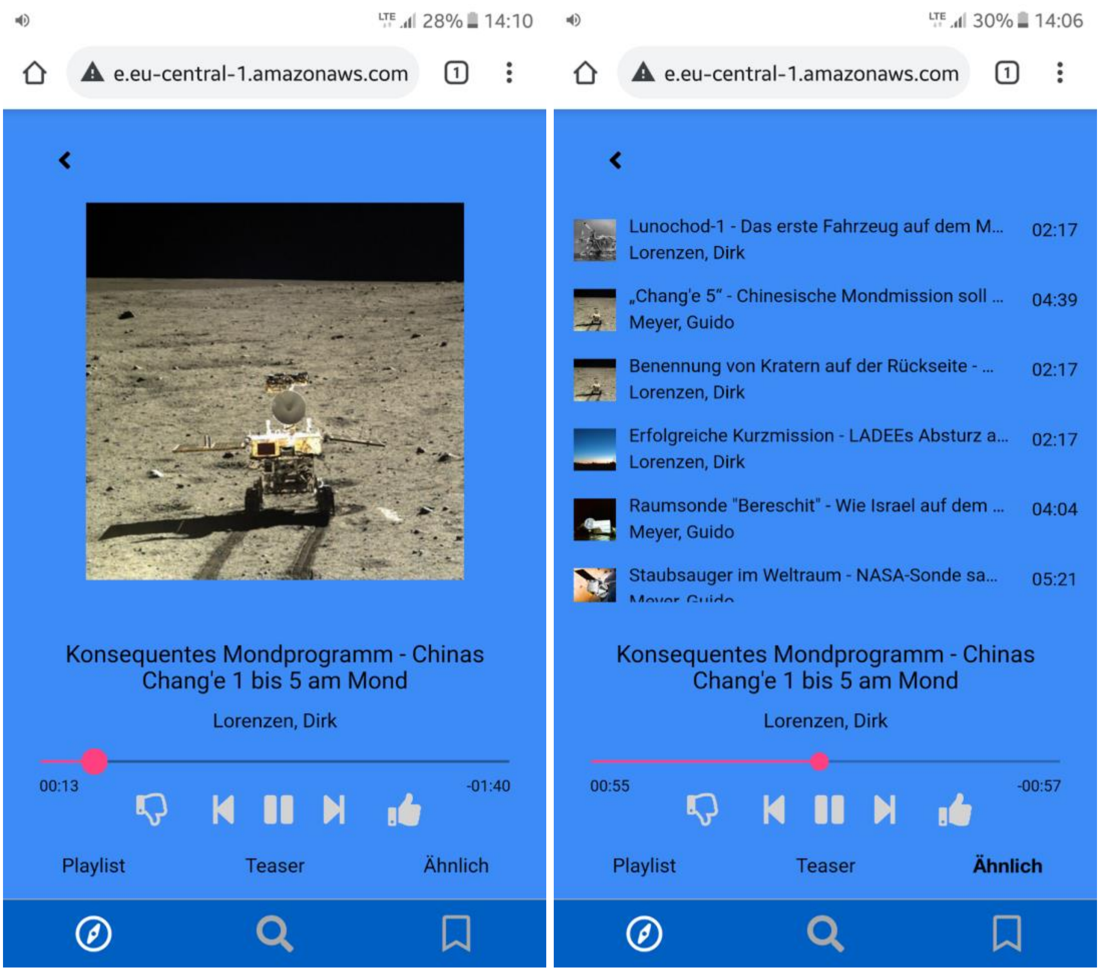

Figure 5-2: Audio play in the expanded state (left), related items touchpoint (right)

The "Playlist" tab on the bottom left under the dislike button allows the user to display the current playlist, as depicted in Figure 5-3. The item currently playing is highlighted by a lighter color, and unlike all other items, cannot be selected. When the user selects an item of a recommender touchpoint, it is added to the current position in the playlist. Finally, the user can display the teaser text of the current item by clicking the "Teaser" tab that is located under the play/pause button (see Figure 5-3). As explained before, this text briefly summarizes the content of the respective radio feature.

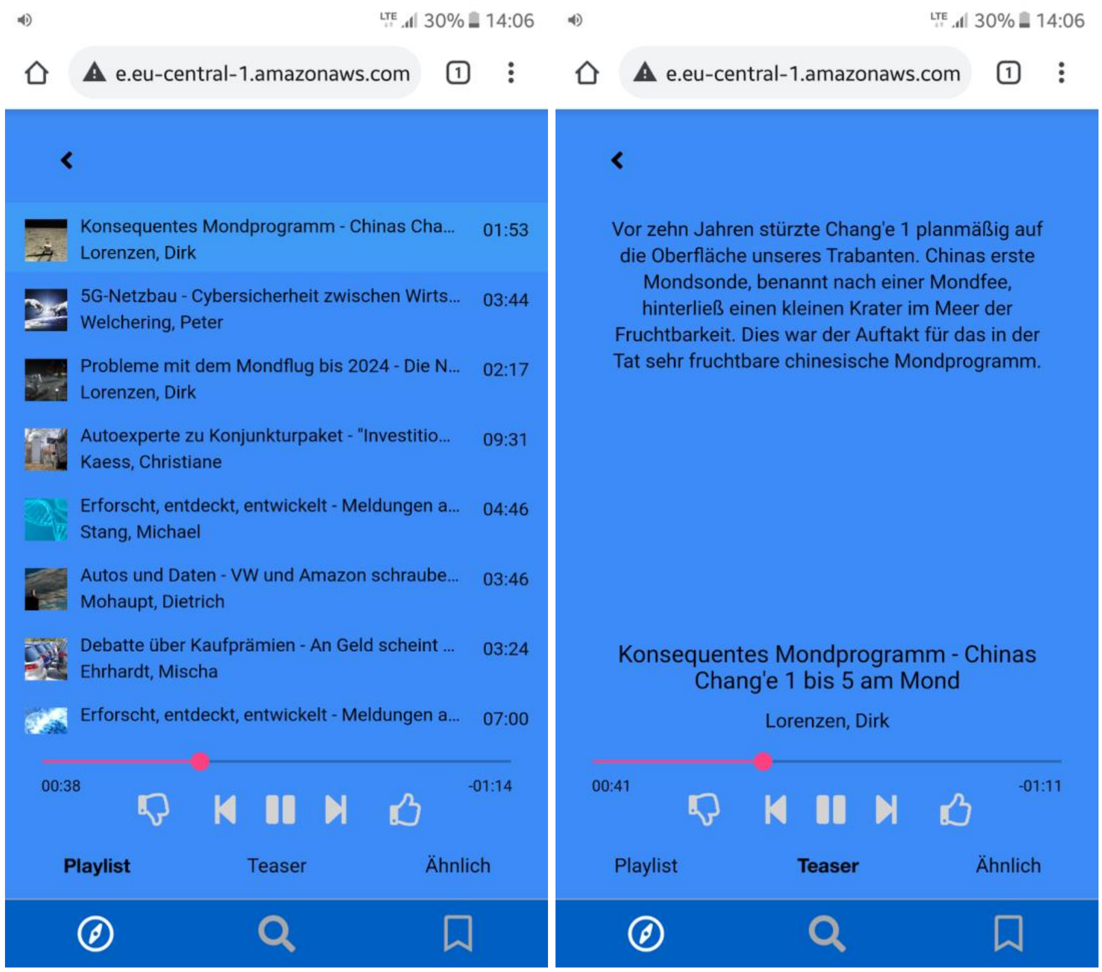

Figure 5-3: Current playlist tab (left), teaser tab (right)

6 Evaluation

In conducting rigorous design science research, evaluation of all outputs is a key activity. By evaluating the artifact, researchers prove that the developed solution solves the problem at hand. Without evaluation, DSR results are unproven claims that the designed artifacts, when implemented and used in practice, will serve their purpose. In addition, (formative) evaluation allows researchers to reflect on and incrementally refine their artifact after each design cycle (Venable, 2006). While any evaluation depends heavily on the artifact, and the purpose of the assessment, literature suggests attributes that can be evaluated. According to Hevner et al. (2004), functionality, completeness, consistency, accuracy, performance, reliability, usability, and other attributes may be subject to evaluation.

We evaluate the functionality and completeness of our artifact in terms of the system's ability to enable the users to find engaging content for all situations in which they listen to spoken-word radio content. Venable et al. (2012) introduced a framework for evaluating DSR projects. Following their terminology, our artifact is a socio-technical product artifact. This means that it is a technology that is used by people to achieve a task and relies on humans interacting with it to provide value. In addition, our evaluation is ex ante and formative in nature, as our artifact is an initial prototype that we will progressively refine in subsequent design cycles.

Evaluation Method

To evaluate our artifact, we exploratively investigate user perceptions of the implemented design features and their ability to suggest relevant spoken-word radio content for different situations. The goal is to collect rich data on the extent to which the implemented recommender touchpoints can suggest engaging spoken-word radio content from the user's perspective. To gather this data, we carry out a user study using our prototype. The study consists of two subsequent phases. In the first phase, participants are asked to explore and use the prototype for one week. Above all, they should consume, like, and dislike content. This ensures both that the recommender algorithm is fed data to provide appropriate suggestions and that participants gain an understanding of the recommender touchpoints. In the second phase, we interview the participants applying the think-aloud method, which is a qualitative research approach that encourages participants to verbally describe what they see, do, and think while performing specific tasks (Bruun & Stage, 2015). We ask the participants to put themselves in the following situations and use the prototype to identify content they would like to listen to in each situation.

- In the morning getting ready for the day
- Doing housework
- In the car
- Focused without doing anything else
- Using public transport
- Doing sports
- Going to sleep

Thus, our method can be seen as mixture of naturalistic and artificial evaluation as participants first use the prototype in their everyday lives (natural environment) and then are interviewed completing tasks in a simulated setting (artificial environment).

There are several variants of think-aloud methods of which we have considered the following three: Ericsson and Simon (1993) introduced the *Traditional Think Aloud* method, which allows no further intervention in the test beyond exploratory phrases. *Speech Communication Think Aloud* follows the speech communication theories by Boren and Ramey (2000). This method allows the interviewer to keep participants engaged in the conversation by picking up on the last word the participant said or using phrases such as "um-huh". The *Coaching Think Aloud* approach goes a step further and allows for more verbal feedback than the previous two methods. By asking direct questions, e.g., on the perceived usefulness of a particular recommender touchpoint for identifying relevant content, the interviewer can guide participants in a certain direction if they have difficulty (Hertzum et al., 2009).

Of these three variants, we chose the *Coaching Think Aloud* method. The resulting semi-structured nature of the interviews allows for tailored exploration of each participant's experience while ensuring that the purpose of the evaluation is not missed. As part of the "coaching" we work out the differences of each touchpoint with the participants before confronting them with the tasks. We also ask them how they perceive the overall quality of the recommendations, and if the implemented touchpoints cover all their needs. In this way, we ensure that we collect data that allows us to draw conclusions about the validity of the design requirements and the general perception of our prototype. For later analysis, we record and transcribe the interviews.

In July 2021, we conducted one-to-one interviews with a sample of 10 German residents (5 female, 5 male, 0 diverse). The participants were aged 17-58 ($M = 32.22$, $Mdn = 27$, $SD = 15.51$). All participants were German and lived in the greater Cologne (Germany) area. The participation in the study was voluntary.

7 Results

In this chapter, we present the results from analyzing the transcribed interviews in three parts. First, we present participants' perceptions as they put themselves in different radio situations and search for relevant content. Second, we summarize how they perceived the overall quality of the suggestions and the different recommender touchpoints. Third, we reflect on the results and draw conclusions about how to refine the design principles and features in preparation for conducting additional DSR cycles.

7.1 Situations

Below, we summarize participants' perceptions and thoughts as they use the prototype to identify content they would like to listen to in a specific situation. The respective situation is shown in boldface.

In the morning

Eight out of ten participants stated they want to consume informative content such as news while preparing for the day after getting up. Their motive is to find out about the latest events. In contrast, two individuals try to avoid any negative content in the morning and prefer light and entertaining content.

Doing housework

All ten participants articulated that they listen to longer features during housework. In addition, they are willing to invest time in finding a suitable feature. While five prefer light content, the other half would search for in-depth content.

In the car

In this activity, participants indicated different preferences. Three of them stressed the importance of not having to interact with the system. They would select the "My Interests" touchpoint to listen to a variety of content that matches their interest. Three other participants prefer short features that deal with light content. The remaining four prefer long, detailed articles that are educational and informative.

Focused listening

All participants indicated that they would listen to content in a focused manner only if they happened to find a very interesting, in-depth feature that required their full attention. None of them would actively look for such a feature.

Using public transportation

Six out of ten participants expressed that they like to consume short informative content when using public transportation. The remaining four would rather listen to long, in-depth features.

Doing Sports

All but one participant stated that they do not consume spoken content while exercising. The person who does so enjoys listening to in-depth features for educational purposes.

Going to sleep

Without exception, all participants stated that they would only listen to light content before bed. Additionally, they prefer longer features that they can fall asleep to.

7.2 Perceptions on Recommender Touchpoints

In the following, we present the participants' perceptions on the different recommender touchpoints.

Overall, participants expressed general satisfaction with the quality of the recommendations. They were particularly impressed by how well the CB RS identifies similar features. One participant using the "Related items" touchpoint stated: *"I like that this touchpoint suggests features that perfectly match the one I'm listening to at the moment."* However, some of the participants complained that the CF RS does not suggest content that matches their interests. In addition, it was criticized that the touchpoints do not offer the possibility to filter by topicality.

My interests

Most of the participants reported they would use this touchpoint a lot. One of them noted: *"I prefer this touchpoint over the other ones since it suggest features that match my interests in general. It saves me from actively searching and I can get inspired when I'm not sure what I want to listen to."* Another participant, on the other hand, would not use this touchpoint but rather *"actively narrow down what type of content is suggested*".

Related items

Our data indicates that this touchpoint greatly contributes to engage the listener. All participants explained they would use it to find more content on a specific topic they are interested in. One person said: *“I would use that touchpoint heavily. I think it might tempt me to consume more than I had planned, just like YouTube.”*

What others like

As mentioned before, some participants noted that the recommendations of this touchpoint do not always match their interests. The social aspect of spoken-word radio content seems to be at least partially important. Most participants explained they would enjoy talking with others about content but not actively search for it.

Information

We found evidence that this touchpoint provides content that is highly relevant to most people. All participants stated they would use this daily to keep up with the latest events that are relevant to them.

Education

This touchpoint was also popular with participants. All ten individuals used it at least once when searching for relevant content during the interview. One participant stated: *"I mainly consume spoken-word content to learn new things or to enhance my existing knowledge."*

Culture

Although not all participants explicitly expressed the need for this touchpoint, most of them mentioned that they would use it time by time.

Insights

Half of the respondents indicated that this touchpoint was relevant to them. The others indicated that they were less interested in learning about others' personal points of view or their lives.

Entertainment

Even though many participants indicated a preference for light, entertaining content in certain situations, they did not interact much with this touchpoint. When asked about this, one stated: *“Although it was clarified what type of content each touchpoint provided, I sometimes found it difficult to find the right one for my needs. If I used the system more, this would probably be easier for me.”*

Society & Values

Three of ten participants stated that this touchpoint was relevant to them. The remaining seven showed less interest but were not averse.

7.3 Reflection & Future Refinements

In this section, we reflect on the evaluation results and draw conclusions about how to refine the design requirements, and principles developed in this work.

The results of the prototype evaluation suggest that our design requirements are accurate and complete. Further, we have evidence that the approach of offering recommender touchpoints that address different consumption motives rather than specific situations proves out. Our data shows that different people prefer different types of content in the same situation. All participants did well with selecting one of the implemented touchpoints for the situation at hand and did not report missing anything. Yet, all participants specified their need for content for a particular situation by distinguishing between short and long, and easy and hard-to-digest content. While the implemented touchpoints allowed them to choose between more light content (culture, entertainment) and more hard-to-digest content (information, education), they had difficulty finding features with a specific duration. Additionally, some missed a possibility to filter the suggested content by topicality. Therefore, we add the following design principle:

DP6: Provide a feature that allows users to filter recommended content by duration and publication date.

All in all, participants were satisfied with the implemented recommender touchpoints. The touchpoints "My Interests", "Related Articles", "Information", and "Education" received particularly positive feedback. On the other hand, less attention was paid to the other touchpoints, and some participants were unsure whether the items suggested by the CF algorithm represent good recommendations. We intend to investigate this in more detail with a larger sample of participants in the next design cycle. Due to the relatively small number of participants in our study, the results should be treated with caution.

8 Discussion, Limitations & Future Research

In this chapter, we discuss the contributions of our work, present its limitations, and offer suggestions for future research.

This work addresses the question of how to design different recommender touchpoints that provide relevant content for different situations in which people consume spoken-word radio content. For this, we conducted one design cycle of a DSR approach aiming to provide a level one design science contribution as defined by Gregor and Hevner (2013).

Following the design requirements developed above, our design provides interested parties with guidance on developing RSs that suggest relevant and engaging spoken-word radio content to its users. The resulting systems aim to increase user engagement with content and enable RBAs to compete with popular media services. Although our design is currently bound to the domain of spoken-word radio, we assume that one could apply the proposed class of systems to spoken-word content in general, e.g., podcasts. We derived design requirements and principles based on spoken-word content and RS literature combined with the knowledge of experts from a major German public RBA. This allowed us to formulate DRs and derive DPs that we propose as useful measures to increase user engagement with spoken-word radio content. Our work differs from previous studies in that it proposes not only a personalized spoken radio stream based on one algorithm, but a design for a system that includes multiple recommender touchpoints that provide personalized content for a variety of consumption motives.

Additionally, it contributes to recent RS literature as it expands this research field to long-tail applications like spoken-word radio.

However, readers should consider the limitations of our research to appropriately interpret the implications of the results. Since we only conducted one design cycle, it does not exceed a level one design science contribution as defined by Gregor and Hevner (2013). We plan to build up on this study as a basis to conduct further design cycles and contribute higher level contributions within future research. This might allow us to develop a more sophisticated set of design principles and technological rules to guide practitioners in developing competitive, state-of-the-art RSs for spoken-word content.

Further limitations lie in the evaluation of our design. Due to the relatively small number of participants in our study, we cannot rule out the possibility that there are other requirements and needs for recommender touchpoints that were not mentioned in our evaluation. Moreover, we conducted an artificial evaluation, as participants were interviewed in an artificial setting and had to put themselves in certain situations. Since such a setting is unreal, evaluation results may not properly reflect reality and thus may not be fully transferable to the real application.

Future research could evaluate the artifact with a larger sample of test listeners and intensively observe their interactions with the system during actual use. Such naturalistic assessments provide more critical validity (Venable et al., 2012). Since we assume that our design can also be applied to related domains such as podcasts, studies in these domains could also yield interesting results. In summary, based on our proposed design, there may be further opportunities for contributions to practice and science.

9 Acknowledgements

We thank the Department of Information Systems and Information Management of Professor Dr. Detlef Schoder at the University of Cologne for making this work possible. Special thanks go to Dr. Stefan Hirschmeier for intensively supervising this work. In addition, we would like to thank the participants in the study conducted and the German public RBA with whom we collaborated.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

Agarwal, D., Chen, B.-C., Elango, P., & Ramakrishnan, R. (2013). Content Recommendation on Web Portals. *Commun. ACM*, *56*(6), 92–101. https://doi.org/10.1145/2461256.2461277

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook* (1st ed. 2016). Springer International Publishing : Imprint: Springer. https://doi.org/10.1007/978-3-319-29659-3

Almeida, F., & Xexéo, G. (2019). Word Embeddings: A Survey. *ArXiv:1901.09069*. http://arxiv.org/abs/1901.09069

Baskerville, R. (2008). What design science is not. *EJIS*, *17*, 441–443. https://doi.org/10.1057/ejis.2008.45

Baskerville, R., & Pries-Heje, J. (2010). Explanatory Design Theory. *Business & Information Systems Engineering*, *2*, 271–282. https://doi.org/10.1007/s12599-010-0118-4

Baskerville, R., Richard, Pries-Heje, J., Jan, Venable, J., & John. (2009). *Soft design science methodology*. https://doi.org/10.1145/1555619.1555631

Bird, Loper, E., Klein, E., & Steven. (2009). *Natural Language Toolkit—NLTK 3.6.2 documentation*. O'Reilly Media Inc. https://www.nltk.org/

Boren, T., & Ramey, J. (2000). Thinking aloud: Reconciling theory and practice. *IEEE Transactions on Professional Communication*, *43*(3), 261–278. https://doi.org/10.1109/47.867942

Bruun, A., & Stage, J. (2015). New approaches to usability evaluation in software development: Barefoot and crowdsourcing. *Journal of Systems and Software*, *105*, 40–53. https://doi.org/10.1016/j.jss.2015.03.043

Burke et al. - Recommender Systems An Overview.pdf. (n.d.).

Burke, R., Felfernig, A., & Göker, M. (2011). Recommender Systems: An Overview. *Ai Magazine*, *32*, 13–18. https://doi.org/10.1609/aimag.v32i3.2361

Camacho-Collados, J., & Pilevar, M. T. (2018). From Word To Sense Embeddings: A Survey on Vector Representations of Meaning. *Journal of Artificial Intelligence Research*, *63*, 743–788. https://doi.org/10.1613/jair.1.11259

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., & Blei, D. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. *Neural Information Processing Systems*, *32*, 288–296.

Chan-Olmsted, S., & Wang, R. (2020). Understanding podcast users: Consumption motives and behaviors. *New Media & Society*, 146144482096377. https://doi.org/10.1177/1461444820963776

Chen, Z., Teng, C.-H., & Chang, P.-C. (2015). Applying artificial immune systems to collaborative filtering for movie recommendation. *Advanced Engineering Informatics*, *29*. https://doi.org/10.1016/j.aei.2015.04.005

Choi, K., Yoo, D., Kim, G., & Suh, Y. (2012). A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, *11*, 309–317. https://doi.org/10.1016/j.elerap.2012.02.004

Davidson, J., Livingston, B., Sampath, D., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., & Lambert, M. (2010). The YouTube video recommendation system. *Proceedings of the Fourth ACM Conference on Recommender Systems - RecSys ’10*, 293. https://doi.org/10.1145/1864708.1864770

Edison Research & NPR. (2020). *The Spoken Word Audio Report*.

Ekstrand, M., Harper, F., Willemsen, M., & Konstan, J. (2014). User perception of differences in recommender algorithms. *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*. https://doi.org/10.1145/2645710.2645737

Ericsson, K., & Simon, H. (1993). *Protocol Analysis: Verbal Reports as Data*. https://doi.org/10.7551/mitpress/5657.001.0001

Gong, S., Ye, H., & Tan, H. (2009). Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System. *Proceedings of the 2009 Pacific-Asia Conference on Circuits, Communications and System, PACCS 2009*, 690–693. https://doi.org/10.1109/PACCS.2009.66

Gregor, S., & Hevner, A. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*, 337–356. https://doi.org/10.25300/MISQ/2013/37.2.01

Harris, Z. (1954). Distributional Structure. *Word*, *10*, 146–162. https://doi.org/10.1007/978-94-009-8467-7_1

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772

Hertzum, M., Holmegaard, K., & c, H. (2009). Scrutinizing usability evaluation: Does thinking aloud affect behaviour and mental workload? *Behaviour & Information Technology*, *28*. https://doi.org/10.1080/01449290701773842

Hevner, A. (2007). The Three Cycle View of Design Science. *Scandinavian J. Inf. Systems*, *19*.

Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In *Management Information Systems Quarterly—MISQ* (Vol. 28, pp. 9–22). https://doi.org/10.1007/978-1-4419-5653-8_2

Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, & Sudha. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, *28*, 75–105.

Hirschmeier, S., & Melsbach, J. (2019). Designing Radio in a Personalized World. *Proceedings of the Americas Conference on Information Systems (AMCIS)*. Americas Conference on Information Systems (AMCIS). https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1116&context=amcis2019

Huang, Q., & Ouyang, W. (2007). Fuzzy collaborative filtering with multiple agents. *Journal of Shanghai University (English Edition)*, *11*, 290–295. https://doi.org/10.1007/s11741-007-0321-2

Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. https://doi.org/10.21105/joss.02174

Hurley, N., & Zhang, M. (2011). Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation. *ACM Trans. Internet Techn.*, *10*, 14. https://doi.org/10.1145/1944339.1944341

Iivari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems*, *24*, 107–115. https://doi.org/10.1057/ejis.2013.35

Kim, H.-N., Ji, A.-T., Ha, I., & Jo, G. S. (2010). Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, *9*, 73–83. https://doi.org/10.1016/j.elerap.2009.08.004

Krause, A. (2020). Krause, A. E. (2020). The role and impact of radio listening practices in older adults' everyday lives. *Frontiers in Psychology*, *11*, 603446. https://doi.org/10.3389/fpsyg.2020.603446

Lam, X., Vu, T., Le, T., & Duc, D. (2008). Addressing cold-start problem in recommendation systems. *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, ICUIMC-2008*, 208–211. https://doi.org/10.1145/1352793.1352837

Lau, J. H., & Baldwin, T. (2016). An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. *ArXiv:1607.05368 [Cs]*. http://arxiv.org/abs/1607.05368

Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st International Conference on Machine Learning* (Vol. 32, Issue 2, pp. 1188–1196). PMLR. http://proceedings.mlr.press/v32/le14.html

Liddy, E. D. (2001). Natural Language Processing. In *Encyclopedia of Library and Information Science* (2nd ed.). NY. Marcel Decker, Inc.

Mai, V. L., Meinzer, N., & Schröter, C. (2019). *Radio- und Audionutzung 2019*. 15.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A Design Theory for Systems That Support Emergent Knowledge Processes. *MIS Q.*, *26*(3), 179–212.

Mayring, P. (2014). *Qualitative content analysis—Theoretical foundation, basic procedures and software solution*.

Melville, P., & Sindhwani, V. (2011). *Recommender Systems* (pp. 829–838). https://doi.org/10.1007/978-0-387-30164-8_705

Mikolov, T., Chen, K., Corrado, G. s, & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR, 2013*.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010, 2*, 1045–1048.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. s, & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, *26*.

Mobasher, B., Burke, R., & Sandvig, J. (2006). *Model-Based Collaborative Filtering as a Defense against Profile Injection Attacks. 2*.

Montes García, A., Alvarez-Rodríguez, J., Labra Gayo, J., & Martínez-Merino, M. (2013). Towards a journalist-based news recommendation system: The Wesomender approach. *Expert Systems with Applications*, *40*, 6735–6741. https://doi.org/10.1016/j.eswa.2013.06.032

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544–551. https://doi.org/10.1136/amiajnl-2011-000464

Ning, X., Desrosiers, C., & Karypis, G. (2015). A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 37–76). Springer US. https://doi.org/10.1007/978-1-4899-7637-6_2

Pariser, E. (2011). *The Filter Bubble: What The Internet Is Hiding From You*. Penguin Books Limited. https://books.google.de/books?id=-FWO0puw3nYC

Park, yoon-joo, & Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. *RecSys'08: Proceedings of the 2008 ACM Conference on Recommender Systems*, 11–18. https://doi.org/10.1145/1454008.1454012

Pazzani, J., & Billsus, D. (2007). Content-based recommender systems. *The Adaptive Web: Methods and Strategies of Web Personalization*, *LNCS*, *4321*, 291–324.

Pazzani, M. (1998). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, *13*. https://doi.org/10.1023/A:1006544522159

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*, 45–77.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *EMNLP*, *14*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Perks, L., & Turner, J. (2018). Podcasts and Productivity: A Qualitative Uses and Gratifications Study. *Mass Communication and Society*, *22*. https://doi.org/10.1080/15205436.2018.1490434

Poriya, A., Bhagat, T., Patel, N., & Sharma, R. (2014). Non-Personalized Recommender Systems and User-based Collaborative Recommender Systems. *International Journal of Applied Information Systems*, *6*(9), 22–27.

Pries-Heje, J., & Baskerville, R. (2008). The Design Theory Nexus. *M I S Quarterly*, *32*(4), 731–755.

Rabe, L. (2021, July 28). *Spotify—Monatlich aktive Nutzer weltweit 2021*. Statista. https://de.statista.com/statistik/daten/studie/812290/umfrage/monatlich-aktive-nutzer-von-spotify-weltweit/

Rashid, A., Karypis, G., & Riedl, J. (2008). Learning preferences of new users in recommender systems: An information theoretic approach. *SIGKDD Explorations*, *10*, 90–100. https://doi.org/10.1145/1540276.1540302

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender Systems: Introduction and Challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 1–34). Springer US. https://doi.org/10.1007/978-1-4899-7637-6_1

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. *Proceedings of ACM E-Commerce*, *1*. https://doi.org/10.1145/352871.352887

Schafer, B., Konstan, J., & Riedl, J. (2001). Electronic commerce recommender applications. *Data Mining and Knowledge Discovery - DATAMINE*, *5*.

Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender Systems in E-Commerce. *Proceedings of the 1st ACM Conference on Electronic Commerce*, 158–166. https://doi.org/10.1145/336992.337035

Schedl, M., & Hauger, D. (2015). Tailoring Music Recommendations to Users by Considering Diversity, Mainstreaminess, and Novelty. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 947–950. https://doi.org/10.1145/2766462.2767763

SciPy community. (2021). *scipy.spatial.distance.cdist—SciPy v1.7.0 Manual*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html

Sein, M., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Quarterly*, *35*, 37–56. https://doi.org/10.2307/23043488

Turcotte, J., York, C., Irving, J., Scholl, R., & Pingree, R. (2015). News Recommendations from Social Media Opinion Leaders: Effects on Media Trust and Information Seeking. *Journal of Computer-Mediated Communication*, *20*. https://doi.org/10.1111/jcc4.12127

Vaishnavi, V., & Kuechler, W. (2007). Design Science Research Methods and Patterns: Innovating Information and Communication Technology. In *Design Science Research Methods and Patterns: Innovating Information and Communication Technology* (p. 226). https://doi.org/10.1201/9781420059335

Venable, J. (2006). A framework for design science research activities. *Proceedings of the 2006 Information Resource Management Association Conference*, 184–187.

Venable, J., Pries-Heje, J., & Baskerville, R. (2012). A Comprehensive Framework for Evaluation in Design Science Research. *LNCS*, *7286*, 423–438. https://doi.org/10.1007/978-3-642-29863-9_31

Wallach, H., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. *Proceedings of the 26th International Conference On Machine Learning, ICML 2009*, *382*, 139. https://doi.org/10.1145/1553374.1553515

Weng, L.-T., Xu, Y., Li, Y., & Nayak, R. (2008). *Exploiting Item Taxonomy for Solving Cold-Start Problem in Recommendation Making*. *2*, 113–120. https://doi.org/10.1109/ICTAI.2008.97

Wintrode, J., Sell, G., Jansen, A., Fox, M., Garcia-Romero, D., & McCree, A. (2015). *Content-based recommender systems for spoken documents*. *2015*, 5201–5205. https://doi.org/10.1109/ICASSP.2015.7178963

Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., & Chen, Z. (2005). *Scalable collaborative filtering using cluster-based smoothing*. 114–121. https://doi.org/10.1145/1076034.1076056

Yao, Y., & Harper, F. M. (2018). Judging Similarity: A User-Centric Study of Related Item Recommendations. *Proceedings of the 12th ACM Conference on Recommender Systems*, 288–296. https://doi.org/10.1145/3240323.3240351

Zhong, J., & Li, X. (2010). Unified collaborative filtering model based on combination of latent features. *Expert Systems with Applications*, *37*, 5666–5672. https://doi.org/10.1016/j.eswa.2010.02.044