Distributional Similarity Music Recommendations Versus Spotify: A Comparison Based on User Evaluation

Nevyana Boycheva

ANR: 548280

Master thesis

Communication and Information Sciences

Specialization: Business Communication and Digital Media

School of Humanities

Tilburg University, Tilburg

Supervisor: dr. S. Wubben

Second reader: dr. C. S. Shahid

January, 2015

Table of contents

Introduction p. 3

Recommendation systems p. 3

The disadvantages of traditional recommendation systems p. 4

Distributional similarity p. 5

The current research p. 5

Theoretical framework p. 6

Content-based filtering p. 6

Collaborative filtering p. 8

Knowledge-based p. 11

Context-aware p. 13

Hybrid designs p. 14

A new type of a recommendation system p. 15

Distributional similarity p. 16

Research questions, variables and concepts p. 21

Method p. 25

Dataset p. 25

Training phase p. 25

Experimental procedure p. 26

Results p. 29

Discussion p. 34

Conclusion p. 37

References p. 38

Appendix A p. 44

Appendix B p. 45

The introduction of the MP3 format in the middle of the 90s and streaming services several years later changed the face of the music industry for good (Magaudda, 2011). Albums could now be listened to and downloaded as individual tracks – legally or illegally (Lam & Tan, 2001). This lead to a massive change in the manner of music consumption that people adopted. One no longer has to make the trip to a record store but can just sit behind a computer and delve deep into the vast collection of music that the Internet has to offer.

Recommendation systems

The growing need to easily discover new music among the vast quantity of songs on the Internet led to the implementation of various recommender systems – websites which give recommendations to a user based on a variety of features or users' behaviours. The most popular examples of such systems include streaming services Spotify[1], Last.fm[2], Pandora[3], Soundcloud[4], etc.

It is not difficult to imagine how different the listening experience on these websites must be compared to the past when music was distributed in physical formats. Spotify for example gives its users access to more than 20 million songs (Setty et. al, 2013) and no brick-and-mortar store could ever compete with that. In the first six months of 2014 streaming has seen a 42% growth compared to the previous year with more than 70 billion tracks streamed on-demand (Nielsen, 2014).

With music being as easily accessible as ever the importance placed on recommender systems is constantly growing because they provide a quick way to sift through the vast quantity of songs and find those suited for a particular user alleviating them from the burden of manually searching for items they would like and thus leading to an easier decision making process while avoiding information overload (Herlocker, Konstan, Terveen, & Riedl, 2004) which is a common phenomenon in the digital age.

[1] https://www.spotify.com
[2] http://www.last.fm/
[3] http://www.pandora.com/
[4] https://soundcloud.com

There are various types of recommendation systems architectures: collaborative filtering, content-based filtering, knowledge-based, context-aware or hybrid designs. The former is currently the most widely spread in online applications. The simplest way to describe collaborative filtering is with the notion of 'word of mouth' – in the past before computers were even invented people always relied on their family and friends for recommendations about everyday choices such as which restaurant to go to or is a certain movie worth renting. Collaborative filtering recommender systems generate recommendations for a given user based on similar users' ratings. Content-based systems generate recommendations based on the features of the items comparing them to items the user has already liked in the past. Knowledge-based recommenders use specific domain knowledge to answer a user query, while context-aware systems take into account information about the context in which the item is to be consumed.

The disadvantages of traditional recommendation systems

All recommendation systems have certain flaws in the way they operate. Collaborative filtering suffers from privacy concerns and the cold-start problem, which is related to a lack of ratings for either new users or new items (Desrosiers & Karypis, 2011). Content-based recommender systems, on the other hand, tend to over-specialize and thus lack the ability to surprise the user (Lops, Gemmis & Semeraro, 2011). Knowledge-based recommenders require knowledge engineering which is referred to as the 'knowledge acquisition bottleneck' problem (Felfering, Friedrich, Jannach & Zanker, 2011). Last but not least, the addition of contextual information in context-aware systems could possibly narrow down the list of recommendations too much (Adomavicius, Mobasher, Ricci & Tuzhilin, 2011).

Due to the constraints of the traditionally used systems this thesis aims at testing a new method for generating music recommendations based on the notion of distributional similarity, an approach originating from the field of natural language processing.

Distributional similarity

According to the distributional hypothesis proposed in the 1950s by the American linguist Zellig Harris words that appear in similar contexts are also similar to each other (Sahlgren, 2008). In distributional models word meanings are represented as vectors and in a high-dimensional space semantically similar words are grouped together (Riordan & Jones, 2011). Distributional representations of words in the vector space play a key role in many natural language processing tasks such as automatic speech recognition and machine translation (Mikolov, Sutskever, Chen, Corrado & Dean, 2013) and the aim of the current research is to test the notion of distributional representations in music recommender systems as well. In computational linguistics distributional similarity uncovers thematic associations among words (Riordan & Jones, 2011) which makes it applicable to the music domain where playlists perform a similar function.

The current research

The distributional representations model was evaluated in terms of quality of recommendations compared to one widely used recommender system. The system of choice for the current study was Spotify which is built upon collaborative filtering algorithms (Bernhardsson, 2013; Johnson, 2014)**.** The quality of the recommendations provided by the two systems was evaluated in a study where participants answered questions about the novelty and serendipity of the recommendations which are the focus of the sub-questions of the current research. Novelty refers to whether the recommendations are new to the user and serendipity signifies how surprising and at the same attractive the recommendations are to the user. A separate analysis also compared the diversity of the recommendations from the two systems in terms of mean number of artists. These metrics are often mentioned in the recommender systems literature as being important for user satisfaction.

The general research question of this study is:

How do the recommendations provided through distributional representations differ in terms of quality from those provided by Spotify?

Theoretical framework

Recommender systems apply principles from different subfields of computer science. According to the Information Retrieval perspective seeking recommendations is similar to seeking information in a search engine in which process the users communicate to the system their information needs in the form of queries (Lops et al., 2011). Recommendations differ according to their features and the extent to which these match the query (Lops et al., 2011).

From the perspective of Machine Learning the recommendation system requires knowledge of users' profiles such as ratings based on which new items are classified as relevant to the user or not (Lops et al., 2011).

In order to understand how recommender systems work the next sections will take a closer look at the approaches which are currently used in various online applications: collaborative filtering, content-based, knowledge-based, context-aware and various hybrids. Despite the fact that the focus of the current research is on music recommendations, this theoretical overview will describe the concepts in general as they are applicable in a wide variety of domains besides music.

Content-based filtering

Content-based recommendation systems extract data about items rated by a particular user and create a profile based on which further recommendations are given. In this case ratings are again acquired through implicit or explicit feedback (discussed in the following section). The only widely-used music recommendation system which relies on content-based filtering is Pandora which is not available in Europe.

In content-based filtering the items are represented as a set of features (also called attributes or properties) which usually exist in the form of unstructured data (Lops et al., 2011). These features could be matched to user profiles in two ways – either by using keywords matching or through the Vector Space Model combined with tf-idf weighting (Lops et al., 2011).

Keyword matching often creates problems because of natural languages' properties synonymy (different words with the same meaning) and polysemy (a single word with multiple meanings). That is why the Vector Space Model provides a relatively simple and reliable method of matching items' features to users' profiles.

In Information Retrieval the Vector Space Model is a spatial representation of documents in the form of vectors in an n-dimensional space (Manning, Raghavan & Schütze, 2008) where each dimension signifies a term present in the document (Lops, et al., 2011). Every vector has a term weight and each weight corresponds to the degree of association between the term and the document. The similarity between vectors in the vector space is measured by their cosine similarity, which is estimated by calculating the cosine of the angle between those vectors in the coordinate system. In content-based filtering the features of the items and the user profiles are represented as weighted term vectors and based on their cosine similarity relevant recommendations are provided to the user (Lops, et al., 2011).

Figure 1 illustrates how the similarity between a query q and the documents d_1, d_2 and d_3 is established by representing them as vectors and computing the cosines of the angles between them.

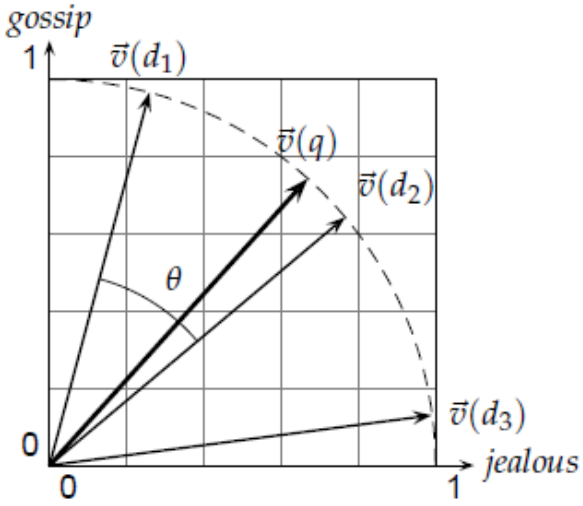

Figure 1: Cosine similarity in information retrieval. Adapted from Manning et al., 2008.

The weights of the terms are acquired through calculating their tf-idf, based on the following principle: frequently occurring terms (tf, term frequency) in a single document but rarely occurring in

the rest of the document collection (idf, inverted document frequency) are deemed as more correctly describing the topic of the document (Lops, et al., 2011). According to tf-idf a weight to a term t in a document d will be assigned using the following formula (Manning et al., 2008):

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \text{ X } \text{idf}_{t}$$

Content-based filtering provides user independence as it relies solely on content features and thus the problem of user privacy often associated with collaborative filtering is solved. For the same reason the problem of recommending new and niche items is also not applicable to content-based filtering. Content-based filtering, however, suffers from the problem of overspecialization which means it does not allow for serendipitous recommendations (Lops et al., 2011). In recommender systems research serendipity is considered as an extension of novelty: all serendipitous recommendations are novel but the other way around is not always true (Lops, et al., 2011).

Other disadvantages of content-based filtering the limited nature of the content analysis which does not allow the system to evaluate the quality of the item (Schafer et al., 2007), and the new user problem: when a user has no profile history and ratings, it is difficult for the system to provide personalized recommendations (Lops et al., 2011).

Collaborative filtering

Collaborative filtering utilizes other users' ratings to make recommendations for the target user (Koren & Bell, 2011). In order for users to be suitable recommendation partners they must have provided similar ratings to certain items in their previous activity (O'Donovan & Smyth, 2005). Collaborative filtering is considered a good choice for applications dealing with content that has subjective characteristics (Schafer, Frankowski, Herlocker & Sen, 2007), such as music, since it does not utilize the content of the item in any way (which is considered to be objective), but is only concerned with people's opinion (which is highly subjective).

Most collaborative filtering systems classify users and items to be recommended using the Nearest neighbour classifying technique (kNN). Generally speaking when a record has to be classified kNN searches for the k closest points from the training data which are referred to as nearest neighbours

(Amatriain, Jaimes, Oliver & Pujol, 2011). The record receives a class label corresponding to the class labels of the nearest neighbours, thus if a record gets assigned to a certain neighbourhood with a given predominant class label, then this record belongs to the same class or neighbourhood (Amatriain et al., 2011). Thus the nearest neighbours of a given user u are the users x, y, z if the ratings for the same items rated by u, x, y and z are similar.

In order to make the recommendation process more efficient collaborative filtering systems (as well as content-based systems) apply a technique called clustering in which items are assigned to groups based on similarity in the form of unsupervised learning where the system by itself discovers natural groups of items (Amatriain et al., 2011). Similarity is measured by the distance between the items, using measures like Pearson correlation and cosine similarity.

There are two manners in which the system could acquire knowledge about the user's opinion of a certain item, either through explicit or implicit feedback (Korren & Bell, 2011). Explicit feedback is the most reliable measure since it consists of a user's direct ratings of items and is believed to most closely represent the user's opinion (Korren & Bell, 2011). However, often explicit feedback data is scarce because rating items requires effort from users (Oard & Kim, 1998). In such cases implicit feedback helps the system establish a user's interests in order to provide further recommendations. Implicit feedback denotes everything a user does on a certain website – time spent viewing a certain page, clicks on links and call-to-action buttons, or in the case of music recommendations whether a certain song was listened to from beginning to end or stopped in the middle. This type of feedback, despite demanding practically no effort from the user, is often not trustworthy (Schafer et al., 2007) as in reality something else could have influenced a user's behaviour which may be unrelated to their opinion of a certain item.

Collaborative filtering approaches are divided into two broad categories – neighbourhood-based (also called memory-based or heuristic-based) and model-based. In the former user's ratings are stored in the memory of the system and used for recommendations by predicting the probable ratings the user would assign to new items (Schafer et al., 2007). Neighbourhood-based approaches have two subtypes – user-based recommendations and item-based recommendations. User-based

recommendations evaluate the interest of a given target user for a certain item by extracting ratings for the same item by other users who provided similar ratings to the target user (thus called neighbours) (Desrosiers & Karypis, 2011). Item-based recommendations use the ratings of a user for a certain item to predict ratings for items similar to that item (Desrosiers & Karypis, 2011).

The other type of collaborative filtering approaches, model-based, learn a model according to which to predict users' interests in certain items based on the ratings which are already given (Desrosiers & Karypis, 2011). Neighbourhood methods focus on computing similarity between items or people using most often metrics like cosine similarity and Pearson correlation. Model-based methods, on the other hand, transform both users and items to a latent factor space and predict ratings based on factors extracted automatically from user feedback thus modelling the similarity between those users or items (Koren & Bell, 2011).

Model-based collaborative filtering is generally considered as being superior to neighbourhood-based approaches in terms of prediction accuracy (Desrosiers & Karypis, 2011). Accuracy, however, is not enough for a recommendation to be satisfying for a user. Serendipity is often mentioned in the recommender systems literature as something upon which users place a lot of importance. Neighbourhood-based collaborative filtering allows for the existence of serendipitous recommendations as it "captures local associations in the data" (Desrosiers & Karypis, 2011).

This big advantage of collaborative filtering stems from the fact that it does not rely on content to give recommendations as it only needs users' feedback. However, this is an issue when the data is sparse (Celma, 2010) as usually not all items on a certain website tend to receive enough ratings. This makes it harder for the system to recommend them (Rashid, et al., 2002) and is especially true for the items in the so-called Long Tail which is comprised of less popular and niche items with fewer ratings (Celma, 2010).

The inability of the system to recommend less popular items, which is applicable in the music domain, leads to less diversified and novel recommendations for the user and the lack of a possibility to broaden the user's taste and thus expand on sales. This is an important problem in e-commerce

websites where often 20% of the products (being books, music etc.) generate around 80% of the profit – a phenomenon related to the Pareto Principle according to which 20% of everything in the world generates 80% of the results (Brynjolfsson, Hu & Smith, 2006).

Privacy issues are also often mentioned as a concern when discussing collaborative filtering (Schafer et al., 2007). This issue is of utmost importance to users especially nowadays in the era of 'big data' and the NSA leaks which left millions of people around the world troubled by the idea that their online activity is being monitored.

Collaborative filtering systems are often referred to as 'black boxes' because they don't provide transparency regarding the recommendations given to users (Desrosiers & Karypis, 2011). Another drawback is their vulnerability to attacks by malicious users who could intentionally skew the ratings for a certain item either in a positive or a negative direction by providing artificial feedback (Schafer et al., 2007) thus diminishing the trustworthiness of the system.

Nevertheless, collaborative filtering is still one of the most popular approaches employed in recommendation systems and is currently used in a number of music applications, for example, Spotify, Last.fm and Soundcloud.

Knowledge based

Knowledge-based recommenders are another type of systems which are considered suitable for recommending items which are usually purchased on a less regular basis, for example, cars, houses or expensive technology. Knowledge-based systems recommend a product to a user based on some requirements that the user provided to the system as input.

Knowledge-based systems resemble content-based in the fact that both make use of items' features which have to be extracted and stored in the memory either manually or automatically using a machine learning algorithm. The main difference is that knowledge-based systems do not need data about users' ratings (Jannach, Zanker, Felfernig & Friedrich, 2011). This way recommendations are independent of users' opinions on certain products but this makes them also less personalized (Burke, 2000).

Knowledge-based recommenders have two subtypes – constraint-based and case-based (Jannach et al., 2011). In both the system provides a solution to certain requirements the user inputs, which could further be changed if no solution is found at first (Jannach et al., 2011).

The difference is that in case-based systems recommendations are given by calculating the similarity between the requirements of the customer and the features of the items, while in constraint-based systems the recommended items must match a set of recommendation rules (Jannach et al., 2011). Constraint-based systems aim at finding a solution to a constraint satisfaction problem, while case-based systems retrieve items from the catalogue based on their similarity metrics.

Knowledge-based recommenders have a strong interactive element, they guide the user based on their input and are therefore referred to as conversational systems (Burke, 2000).

In constraint-based recommenders different techniques to improve usability, user satisfaction and quality of the output are used to help the users when interacting with the system (Jannach et al., 2011). These could be for example providing default values in cases when users lack knowledge of technical details or selecting the next question and thus identifying features that the user may be interested in (Jannach et al., 2011).

In case-based recommender systems the technique which aids users who lack substantial domain knowledge is called critiquing and allows users to make specific changes to the request by selecting filtering criteria which have not been satisfied yet.

Knowledge-based recommenders do not suffer from the cold-start problem which is characteristic of collaborative filtering and content-based filtering (Burke, 2000). On the other hand, these systems are an example of the so-called ‘knowledge acquisition bottleneck’ because they require knowledge-engineering to transform the knowledge of domain experts into representations that the system could work with (Felfernig et al., 2011).

Context-aware

Context-aware recommenders are systems which recommend certain items to a user while considering the contextual situation in which the items would be consumed. 'Context' is considered any piece of information which defines the situation in which the user currently is (Celma, 2010).

Recommender systems which incorporate contextual information aim at identifying contextual factors which further influence the generated recommendations. These factors could be either dynamic or static (Adomavicius et al., 2011).

Researchers distinguish between two different points of view regarding context, the representational and interactional view. According to the former, context is a stable piece of information not dependent on any activity, while the latter posits that context consists of a set of previously known 'observable attributes' which are not static (Adomavicius et al., 2011). According to the interactional view context and activity are two parts of a cycle which influence each other (Adomavicius et al., 2011).

If traditional content-based and collaborative filtering systems are considered to be two-dimensional and take into account only users and items, context-aware systems add context to users and items as a factor for estimating possible ratings. Contextual information could be gathered in various ways, either explicitly by asking direct questions, implicitly by using data from the environment of the users such as location tracking in mobile devices, or the context could be inferred using methods from statistics and data mining (Adomavicius & Tuzhilin, 2011).

Context-aware recommender systems employ three different paradigms – contextual pre-filtering, contextual post-filtering and contextual modelling.

Contextual pre-filtering selects the most relevant user and item values based on contextual information and then generates recommendations based on a limited amount of relevant data (Adomavicius et al., 2011). If the contextual information narrows the options down too much, generalized pre-filtering creates a new generalized data which match the context but are broader (for

example if the contextual data include Saturday, generalized pre-filtering would include all data from the weekend as well) (Adomavicius et al., 2011).

Contextual post-filtering, on the other hand, ignores information about context when generating recommendations at first but incorporates it at a later stage to match the recommendations to the user's contextual requirements (Adomavicius et al., 2011). This is done by either excluding recommendations that are not relevant for a given context or changing the ranking of the recommendations (Adomavicius et al., 2011).

This approach could rely either on heuristic or model-based techniques. The focus of the former is finding characteristics of an item which match a given context and based on these adjust the recommendations (Adomavicius et al., 2011). Model-based techniques built predictive models which estimate the probability that the user will choose a certain item in a certain context and respectively generate recommendations (Adomavicius et al., 2011).

Contextual modelling, on the other hand, incorporates contextual information from the beginning of the recommendation process thus transforming it into a multidimensional process (Adomavicius et al., 2011). Again recommendations are either generated using predictive models or heuristic calculations which add contextual data to the user and item data.

Hybrid

Hybrid recommenders combine different recommendation techniques in order to avoid some of the limitations associated with traditional approaches discussed in the previous section of this chapter and thus provide better quality recommendations. The most common problem which hybridization approaches solve is the cold-start problem (Burke, 2007) encountered both in collaborative filtering (lack of ratings for new users and items) and content-based filtering (lack of ratings for new users).

A hybrid recommender system is any system which combines two or more recommendation techniques. Those could either be different (collaborative filtering and content-based filtering) or of the same class (two different content-based recommenders) (Burke, 2007).

There are various techniques to build a hybrid recommender system among which three main designs could be distinguished – monolithic, parallelized and pipelined (Jannach et al., 2011).

Monolithic hybridization designs have two subtypes, feature combination and feature augmentation (Jannach et al., 2011). In feature combination features of one recommendation source are added to an algorithm usually used for processing data of another recommender source (Burke, 2007). In this approach there is only one source of recommendations but a different source of knowledge, "a feature combination hybrid borrows the recommendation logic from another technique rather than employing a separate component that implements it" (Burke, 2007).

The other type of monolithic hybridization design is quite similar to feature combination and is called feature augmentation (Jannach et al., 2011). This hybrid does not use features' data obtained from the additional recommender but generates them based on the principles of work of the additional recommender (Burke, 2007).

Parallelized hybrid designs are divided into three subtypes: weighted, mixed and switching (Jannach et al., 2011). Weighted hybrids include scores for a given item from both recommender elements which are then linearly combined (Burke, 2007). Mixed hybrids, on the other hand, present recommendations from the different elements simultaneously in a list without combining the scores (Burke, 2007). Switching hybrids select a single recommender element which is expected to be the most suitable for a given situation based on specific criteria (Burke, 2007).

The last type of hybridization design is called pipelined and includes cascade and meta-level hybrids as subtypes (Jannach et al., 2011). Cascade hybrids establish a hierarchical order in which a weaker hybrid is used to refine the scores estimated by a stronger hybrid (Burke, 2007). Meta-level hybrids, on the other hand, adopt a model which one recommender has already learned and use this model as input for another recommender (Burke, 2007).

A new type of recommendation technique

The advantages and disadvantages of the traditional widely-used recommendation systems are outlined in the respective sections. Table 1 in Appendix A provides an overview to which some

assumptions about the performance of a recommender system based on distributional similarity are added. A recommendation system based on distributional similarity of songs mined from playlists could possibly surpass some of the problems usually associated with the traditional approaches and provide high quality recommendations despite its relative simplicity. As it would not rely on user data, there are no privacy issues and the users would be independent from one another.

The distributional similarity method could make a valuable addition to current recommendation approaches if not as a stand-alone technique then as a component of a powerful hybrid design.

Distributional similarity

The idea of distributional similarity stems from the distributional hypothesis proposed in the middle of the 20th century by Zellig Harris according to which 'words which are similar in meaning occur in similar contexts' (Sahlgren, 2008). This notion establishes a correlation between distributional similarity and meaning similarity and allows us to 'use the former to estimate the latter' (Sahlgren, 2008). Distributional models rely on statistical regularities in order to establish co-occurrences of words and thus semantic representations of those words in the n-dimensional space are built (Riordan & Jones, 2011). If we match the distributional hypothesis to the Vector Space Model, words similarly distributed across the vector space will have similar meanings.

According to Harris basic classes could form groups based on distributional behaviour therefore if two words share similar distributional properties (such as co-occurrence with another word) then they are representatives of the same linguistic class (Sahlgren, 2008). It is considered that if language resembles the structure of the environment we occupy, then linguistic distributional representations would match meaning representations (Riordan & Jones, 2011). Distributional similarity could be the result of meaning similarity but could also stem from extralinguistic factors, thus differences in meaning correlate with differences in distribution (Sahlgren, 2008) but this does not establish a causal relationship between the two.

As an example, the word 'student' may often be found in the same context with the words 'university' and 'thesis'. From the distributional hypothesis it would follow that these three words are

semantically related. If we observe the contexts in which the word ‘university’ appears we may also see the word ‘faculty’. Distributional models will group these words together in the n-dimensional space and thus we could infer that ‘student’ and ‘faculty’ are also related even though they rarely occur in the same context but share a relation with the word ‘university’.

Distributional similarity models create word representations in the form of vectors associated with words and placed in a co-occurrence matrix which includes vocabulary size, words and contexts (Turian, Ratinov & Bengio, 2010). Distributed representations are an additional approach to word representation. These compact low-dimensional representations, also called word embeddings, represent latent features of words which give semantic information (Turian, Ratinov & Bengio, 2010) based on which similarity could be inferred.

From a semantic point of view similarity may or may not be due to meaning but it always follows a distributional regularity (Sahlgren, 2008). Thus it is postulated that if words X and Y are more semantically different than word Z, then their distributions would also be different. Distributional models make it possible to divide the whole language into groups based on distributional properties and it has been established that these models are most successful in inferring similarity when they operate with large textual context such as documents (Sahlgren, 2008).

There are many approaches for inferring word similarity from distributional representations. Here, some of the most common ones, Latent Semantic Analysis and Latent Dirichlet Allocation will be briefly discussed and afterwards the newer models continuous bag-of-words and continuous skip-gram on which the current research will be based will be outlined.

LSA and LDA. Latent Semantic Analysis (LSA) is a method used for clustering entities (such as words) by ‘extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text’ (Landauer, Foltz & Laham, 1998). The model generates representations which exhibit word similarity (Landauer & Dumais, 1997). The input which LSA uses is words organized into sentences or paragraphs thus LSA relies on the context of use to infer relations between entities but does that by interpreting the text passage as a bag-of-words

without taking into account word order and syntactic or morphological relations (Landauer et al., 1998).

In LSA, words, sentences, paragraphs or any kind of text passages are plotted as points on a high dimensional semantic space by applying Singular Value Decomposition (SVD) (Landauer et al., 1998). Through SVD a rectangular matrix whose rows and columns represent different concepts could be linearly decomposed into the product of three other matrices (Landauer & Dumais, 1997) thus reducing dimensionality. By doing this the model captures high-order relationships between words and/or passages, such as if X is related to Y and Y is related to Z, then X is related to Z as well (Landauer & Dumais, 1997). This closely approximates human cognition, and correlations of LSA's relations between words and text passages and cognitive functions such as association and semantic similarity have indeed been found (Landauer et al., 1998).

There are two main steps in LSA. First, the text should be represented as a matrix in which rows stand for words and columns stand for contextual entities such as paragraphs or sentences. Each cell shows the frequency of appearance of the word in the corresponding context. The second step is applying SVD and decomposing the matrix into three componential matrices. The first and the second present respectively the vectors of the entities in the rows and the columns of the original matrix. The third is a diagonal matrix with scaling values and if the three of them are multiplied, the result would be the original matrix.

To conclude, LSA works on the principle that "a representation that captures much of how words are used in natural context captures much of what we mean by meaning" (Landauer & Dumais, 1997).

A similar method to LSA is Latent Dirichlet Allocation (LDA) which stresses the exchangeability of topics in a certain document which generate the words (Blei, Ng & Jordan, 2003), thus LDA's function is probabilistic topic modelling. LDA reduces documents to a set of features which represent the Dirichlet parameters of the document. The model has three levels where each item is modelled over a set of topics (Blei, Ng & Jordan, 2003).

Distributional similarity in music recommendations. A framework called Auralist is an example of LDA being used in the domain of music recommendations. Auralist is a hybrid system which combines an item-based collaborative filtering algorithm built upon LDA with two other algorithms with the aim to produce recommendations which go beyond accuracy and emphasize novelty, serendipity and diversity in a recommendation list (Zhang, Seaghdha, Quercia, Jambor, 2012). Those characteristics of music recommendations will be discussed in detail in the following section.

Another work from the domain of music recommendation systems which is directly related to the idea of distributional similarity is a playlist generation algorithm called Latent Markov Embedding (LME) which uses example playlists to learn representations (Chen, Moore, Turnbull & Joachims, 2012; Moore, Chen, Joachims & Turnbull, 2012; Chen, Xu & Joachims, 2013). In LME playlists are represented as Markov chains and are embedded as points in a latent space (Chen, Moore, Turnbull & Joachims, 2012). After the embedding process is completed LME calculates the probabilities of meaningful transitions occurring between the songs (Chen, Moore, Turnbull & Joachims, 2012) and thus generates new playlists from the training data.

Word2vec. Recently researchers have proposed two new models for computing vector representations of words which could learn word vectors from a very large vocabulary dataset and as mentioned above, distributional similarity demands large context in order to correctly identify relations between words. The continuous bag-of-words model uses the context to predict the word and the word order does not affect the vector representation (Mikolov, Chen, Corrado & Dean, 2013). In the continuous skip-gram model, on the other hand, words surrounding a particular word within a certain range in a sentence could be predicted assuming that the more distant from each other two words are, the less related they are (Mikolov, Chen, Corrado, et al, 2013). These models are relatively simple and require short training time (Mikolov, Le & Sutskever, 2013).

According to the notion of distributional similarity it is considered that when two words appear in the same context during the training phase of the algorithm, the system brings the vectors of these words closer in the vector space (Mikolov, Le & Sutskever, 2013). The word representations learned

by the models exhibit semantic similarities (Mikolov, Yih & Zweig, 2013). These similarities could have different degrees and could be inferred by performing simple algebraic operations on the vectors, as in the following example: vector ('Madrid') – vector ('Spain') + vector ('France') = vector ('Paris') (Mikolov, Sutskever, Chen, Corrado & Dean, 2013). This goes to show that the model is capable of organizing concepts based on their distributional similarity without receiving prior instructions about the manner in which these concepts are related (Figure 2).

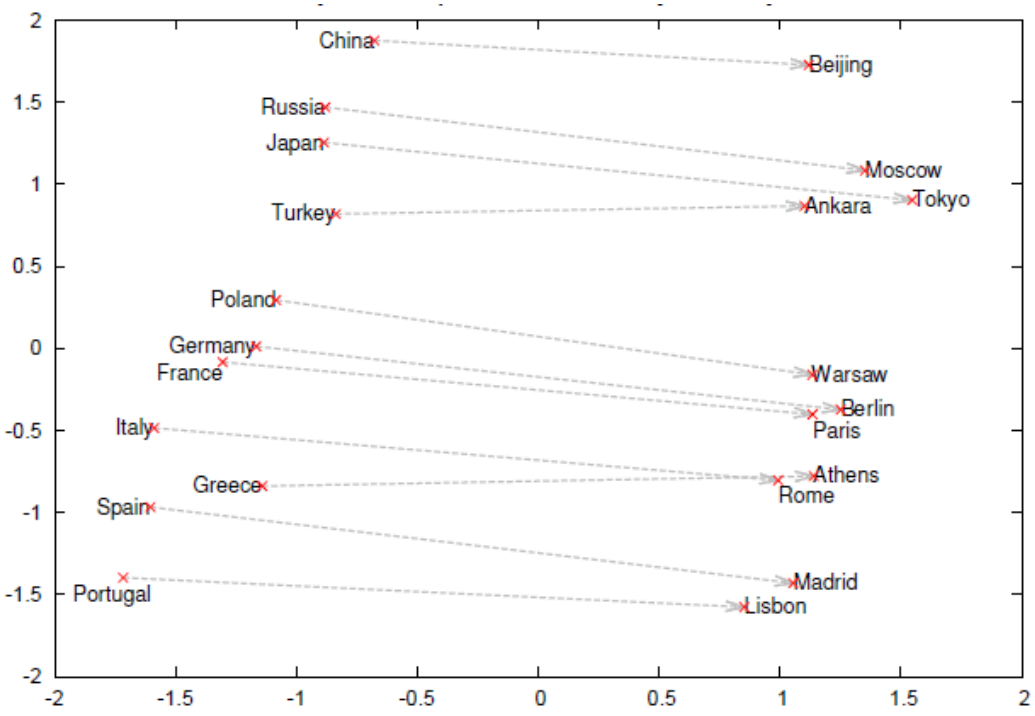

Figure 2: The projections of the vectors of countries and their capital cities. Adapted from Mikolov, Sutskever, Chen, et al. 2013.

Word2vec[5], the tool the current research relies on, is an open-source tool which implements the continuous bag-of-words and continuous skip-gram architectures. It takes text as input and provides an output in the form of word vectors by first constructing a vocabulary from the text input and then learning the vector representations.

[5] https://code.google.com/p/word2vec/

Research questions, variables and concepts

The aim of the present research is to compare the performance of a distributional similarity recommender to a widely-used system such as Spotify. Spotify is a commercial service for streaming music which provides new artists and songs recommendations to users based on their previous activity on the platform, similar to the artists they have already listened to or organized in playlists by genre or mood.

Each song on Spotify has a 'Start Radio' option upon clicking on which the user is presented with a playlist of songs similar to the one they had been listening to. The version used for the current research was the Spotify Web Player[6] which could be used for free after creating an account.

Spotify is based on collaborative filtering techniques such as matrix factorization and building latent factor models to recommend music to its users (Bernhardsson, 2013; Johnson, 2014). It is important to note that the application relies exclusively on implicit feedback obtained through observing user behaviour and does not ask for user ratings of the songs (explicit feedback).

The comparison between the two systems will be carried out based on users' evaluation of the quality of recommendations. Although accuracy is the most obvious choice of a quality metric, numerous researchers (Herlocker et. al, 2004; Ziegler, McNee, Konstan & Lausen, 2005; McNee, Riedl & Konstan, 2006; Ge, Delgado-Battenfeld & Jannach, 2010; Lops, et al., 2011; Zhang, et al. 2012) emphasize the importance of metrics such as novelty, serendipity and diversity of recommendations in improving users' satisfaction and quality evaluations.

Novelty is the ability of the recommender system to provide the user with unknown recommendations which they have not experienced yet (Zhang, et al. 2012). An item which the user is already familiar with could be considered as an accurate recommendation but also a fairly obvious one (Herlocker et. al, 2004), thus lacking in novelty. Obvious recommendations do serve the purpose of establishing a new user's trust into the system (Swearingen & Sinha, 2001). However, as users'

[6] https://play.spotify.com/

usage of the system persists, the recommender's algorithm should be able to include novel items in the recommendation lists or otherwise there exists a risk that the user will become bored with the recommendations and stop using the system as it would not introduce them to anything new.

Serendipity is an extension of the concept of novelty and represents the 'unusualness' of the recommended content (Zhang, et al. 2012). Serendipitous recommendations could possibly be less similar to the items the user has already rated but nevertheless the user may perceive them as interesting (Ge, et al. 2010). An important characteristic of a serendipitous recommendation is that the user should not be able to discover it on their own (Herlocker et. al, 2004) as it is in some way different from the user's usual preferences.

Serendipitous recommendations are 'non-trivial' recommendations which are believed to match users' interests which they have not yet expressed (Ge, et al. 2010) or they are possibly even not aware of. These recommendations are not just unexpected (unusual as mentioned above) and surprising, but they are also attractive, interesting and useful to the user (Ge, et al. 2010, Zhang, et al. 2012).

Consider the following example. Band X is user Y's favourite band. The recommender system recommends user Y songs by band X as well as songs by the solo projects of band X's members. Being a big fan of band X user Y is acquainted with all of its and its band members' material. These recommendations, then, are neither novel, nor serendipitous, since Y could find them by herself. If, on the other hand, the system takes the user out of her comfort zone and takes the risk of recommending a song that is quite different from what band X plays, but the user still likes it, then that is a novel and serendipitous recommendation which could possibly broaden user Y's interests.

Both novelty and serendipity in a recommender system contribute to lowering the level of obviousness thus diversifying the user's experience (Vargas & Castells, 2011) and broadening the user's tastes (Herlocker, et. al, 2004; Ge, et al. 2010). The concept of serendipity is hard to grasp due to the high level of subjectivity involved (Ge, et al. 2010) but nevertheless it is an interesting characteristic of recommendation systems which several authors suggest could be measured by asking

users to evaluate a list of recommendations (Ge, et al. 2010; Parameswaran, Koutrika, Bercovitz & Garcia-Molina, 2010; Zhang, et al. 2012).

The final metric the current research would pay attention to is the diversity of recommendations. Diversity is considered as a measure of the usefulness of a recommendation list is as it broadens the field of choice for the user (Vargas & Castells, 2011). It stands for the variety in a set of recommended items (Zhang, et al. 2012) or 'how different the items are with respect to each other' (Vargas & Castells, 2011). In the above example of user Y and her favourite band X, an assumption could be made that a recommendation list consisting only of songs by band X would not be a truly diverse one.

A connection between novelty and diversity exists, such that when the recommendations from a certain system are novel to the user, this leads to a diversified user experience (Vargas & Castells, 2011).

Therefore this thesis will try to answer the following research questions:

RQ1: How do the recommendations based on distributional similarity differ from those provided by Spotify in terms of novelty?

RQ2: How do the recommendations based on distributional similarity differ from those provided by Spotify in terms of serendipity?

RQ3: How do the recommendations based on distributional similarity differ from those provided by Spotify in terms of diversity?

Based on the preceding theoretical review the present research puts forward the following hypotheses:

H1: A recommender system based on distributional similarity will provide the user with more novel recommendations than Spotify.

H2: A recommender system based on distributional similarity will provide the user with more serendipitous recommendations than Spotify.

H3: A recommender system based on distributional similarity will provide the user with more diverse recommendations than Spotify.

Method

Dataset

The dataset used in the current study is a publicly available dataset under the name 'Playlist Dataset' found on the Internet[7] which consists of 75 262 songs organized in playlists. The data has been collected by Shuo Chen from Cornell University from December 2010 to May 2011 by crawling the website Yes.com. Yes.com used to be a web service providing information about playlists from all radio stations in the US but is no longer functioning. The dataset had been used in several studies mentioned in previous sections (Chen et al., 2012; Moore, et al., 2012; Chen, et al., 2013).

Using an API, the playlists of various US stations have been obtained for a 7-day period which leads to this data set containing playlists of various genres from a variety of music stations (Chen et al., 2012; Moore, et al., 2012; Chen, et al., 2013). The data is divided into a training set and a testing set, where each song is listed with its numeric ID. These IDs range from 0 to the total amount of songs minus 1. A separate file in the dataset provides information about which ID corresponds to which song.

Training phase

The recommendation system on which the current research is focused was created by using the distribution of the Yes.com playlists from the dataset described above, transformed to high-dimensional vectors by word2vec.

A distributional model was trained by feeding the dataset to word2vec which computes vector representations of words or in the present case, of songs from playlists. The word2vec hyper parameters used were the default ones presented in the instructions of the tool (size of word vectors: 200; training iterations: 3; window, or the maximum skip length between words: 5; model: continuous bag-of-words).

[7] http://www.cs.cornell.edu/~shuochen/lme/data_page.html

After the training was completed, recommendation lists for 25 songs were created by using the *Distance* tool in word2vec which computes the similarity between different elements in the data which was previously fed to the system. The songs which were selected for the task were picked from an article in the magazine Rolling Stone, titled '500 Greatest Songs of All Time'[8]. All of the songs are from a period ranging from the 1960s to 1990s and generally consist of what could be deemed as 'classics'. This choice was made in order to make sure that every participant would be able to find a song they were familiar with despite their specific taste in music.

The *Distance* tool computes the cosines of the angles between the vectors after embedding them in a high-dimensional space. The resulting numbers represent the cosine similarity (or distance) between particular vectors and based on this information a conclusion could be made that song x is more similar to song y if their cosine similarity is higher.

For example, according to the vector representations created by word2vec song 3727 has a cosine distance of 0.9939 to song 3810 and 0.9931 to song 3605 (both of them are located in the top of the list, which means they should be the most similar to the seed song), but 0.9768 to song 19886 (the last one on the list of 40 songs which means it should be the least similar). Upon checking the file with the song titles, it can be discovered that the seed song is James Brown's 'I Got You (I Feel Good)' and the first two most similar songs were respectively Rick James' 'Super Freak' and Heart's 'These Dreams', while the last one on the list is England Dan and John Ford Coley's 'Love is the Answer'.

Experimental procedure

Questionnaire. In the experimental procedure the recommendations from word2vec were compared to the recommendations from Spotify for the same seed songs. The design of the study was within-subjects: after training word2vec on the dataset, the seed songs and the corresponding recommendations from both systems were included into an online survey in which the participants were asked to evaluate the recommendations from both systems by answering questions related to the characteristics of novelty and serendipity. For each system the first five recommendations from the

[8] http://www.rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407

recommendation list were used. These recommendations were organized in YouTube playlists which also included the seed song.

In the survey the participants were first presented with a list of seven songs and asked to choose the one they were the most familiar with. After doing so, they were directed straight to two question blocks containing the recommendation lists from the distributional similarity recommender and Spotify for the given seed song and four questions related to the novelty of recommendations, degree of fit to the seed song, the likelihood of finding the recommendations without the help of a recommendation system and how much the recommendations appealed to the participants. They were asked to answer the questions on a 5-point scale ranging from 'Not much' to 'A great deal'. After doing so, the participants were presented with another list of seven songs to choose from and afterwards they had to answer the same questions. The whole procedure was repeated three times thus resulting in more than one observation per subject which was done to ensure the accuracy of the measurement. A model of the survey can be found in Appendix B.

It is important to note that the participants were neither given any details about the two systems, nor were informed which recommendations belonged to which system to avoid any possible bias. The presentation of the distributional similarity lists and Spotify lists was randomly rotated to try to counteract order effects.

An analysis on the diversity of recommendation lists was also conducted by observing the number of various artist names in all of the recommendation lists. An assumption was made that the more various artists the lists included, the more diverse they were. Results are presented in the following section.

The independent variable in this research is the type of a recommendation system (distributional similarity based or Spotify) and the dependent variables are the four questions.

Participants. 46 participants completed the online questionnaire. Four of them were excluded due to missing data. From the remaining 42 participants 25 were female and 17 were male. Their age ranged between 18-56 with an average of 25.69 (SD = 6.1) and a mode of 24. Each of the 42

participants gave their evaluation of the two recommendation sets three times which lead to a total of 126 observations.

Results

The data from the questionnaire were entered into SPSS and analysed using descriptive statistics, Levene's test of homogeneity of variance, a t-test and a correlation analysis. Overall, the recommendation sets from Spotify received higher scores on all four questions.

Question 1

The question '*How familiar would you say you are with the recommended songs*?' relates directly to the concept of novelty discussed earlier. The songs recommended by the distributional similarity recommender system were overall more novel to the participants (M = 2.5, SD = 1.3) compared to the songs recommended by Spotify (M = 3.5, SD = 1.3). The variances were equal for the two recommendation systems, $F(1, 82) = 1.75$, *ns*. A paired samples t-test revealed that this difference was significant, $t(41)=-7.249$, $p < 0.01$, $r = .75$.

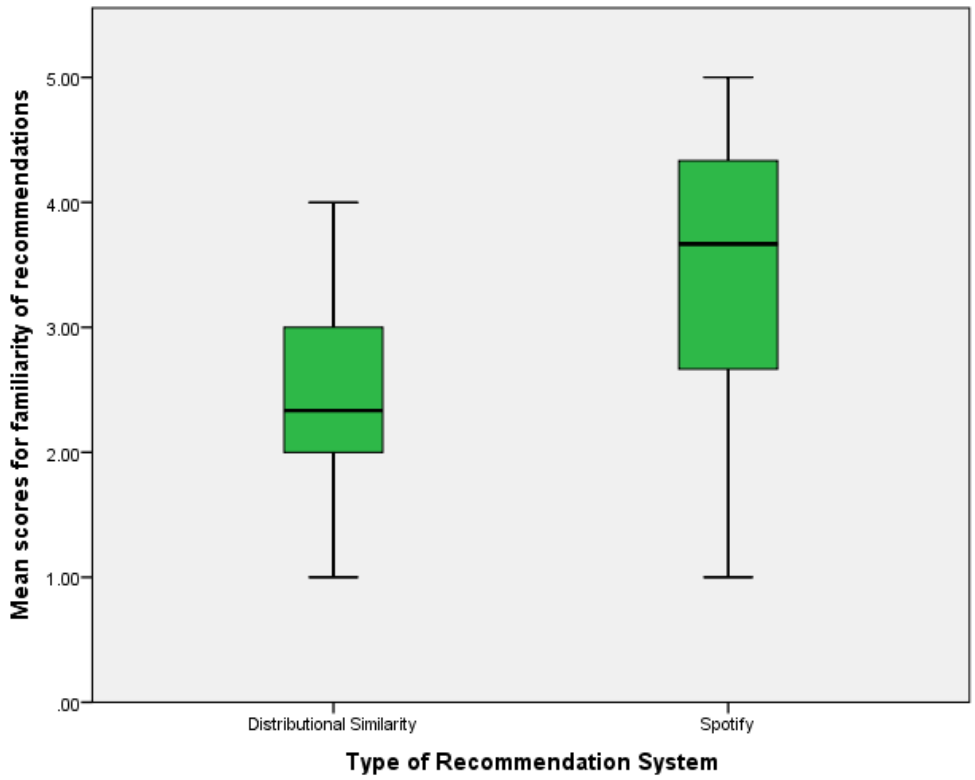

Figure 3: A boxplot showing the mean scores of the two recommendation systems for familiarity of the recommendations.

Question 2

The question '*How well do the recommended songs fit the song chosen by you*?' in a similar fashion shows a higher average score for the Spotify playlists (M = 3.7, SD = 1), compared to the distributional similarity playlists (M = 2.6, SD = 1.1). The variances were equal for the two recommendation systems, $F(1, 82) = .15$, *ns*. This difference was also significant, $t(41)=-7.711$, $p < 0.01$, $r = .77$.

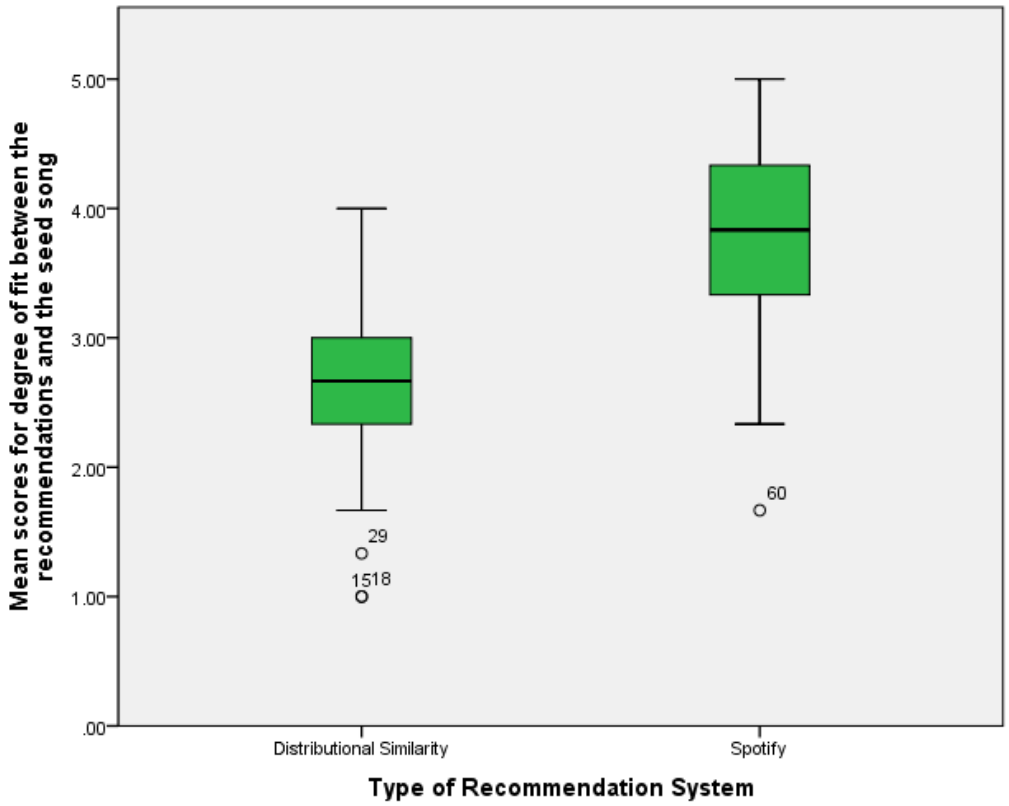

Figure 4: A boxplot showing the mean scores of the two recommendation systems for the degree to which the recommendations fit the seed song.

Question 3

The question '*How much do you like the recommended songs*?' in a similar fashion shows a higher average score for the Spotify recommendations (M = 3.6, SD = 1) compared to the distributional similarity playlists (M = 2.7, SD = 1.2). The variances were equal for the two recommendation systems, $F(1, 82) = 1.25$, *ns*. The difference was significant, $t(41)=-7.439$, $p < 0.01$, $r = .76$.

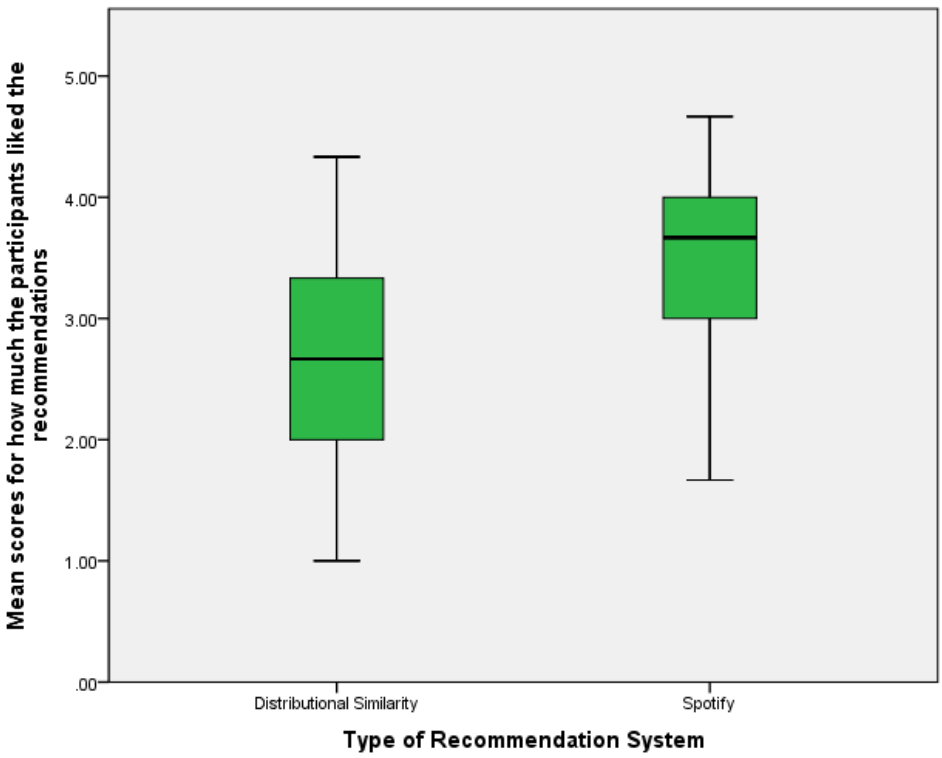

Figure 5: A boxplot showing the mean scores the two recommendation systems received for how much the participants liked the recommendations.

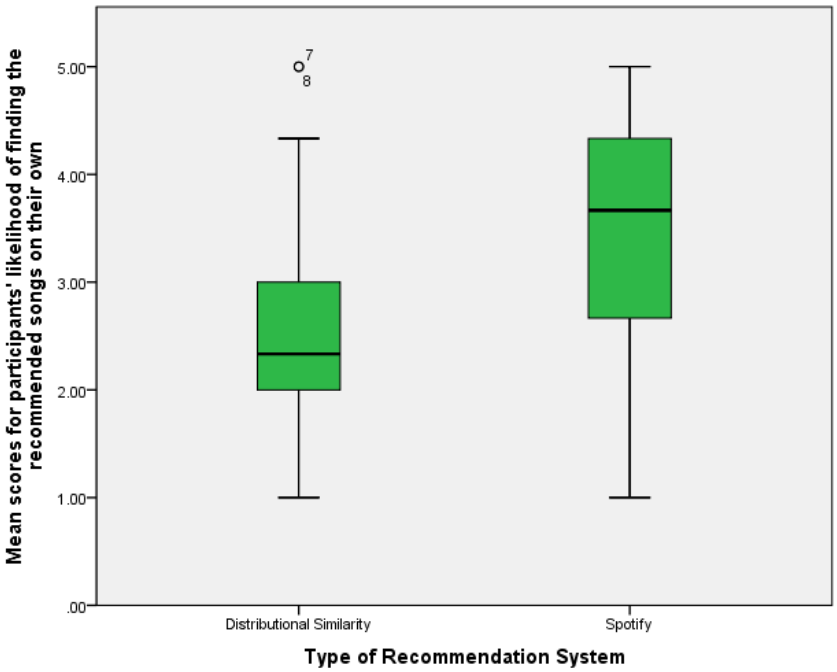

Figure 6: A boxplot showing the mean scores the two recommendation systems received for how likely the users were to find the recommended songs on their own.

Question 4

The scores for the final question '*How likely are you to find the recommended songs on your own*?' show that as participants were more familiar with the Spotify recommendations, they were also more likely to find those songs on their own (M = 3.5, SD = 1.3) compared to the recommendations from the distributional similarity recommender (M = 2.6, SD = 1.2). The variances were equal for the two recommendation systems, $F(1, 82) = .43$, *ns*. The difference was significant; $t(41)=-5.518$, $p < 0.01$, $r = .65$.

Figures 4 and 6 show the presence of several outliers in participants' scores for the degree of fit between the recommended songs and the seed song, and the likelihood of finding the recommended songs on their own. After careful consideration of the situation and the nature of the research a decision was made to leave those values unchanged since evaluating music is a very subjective task and extreme values seem possible depending on participants' tastes.

The results from the performed Pearson correlation, show a positive correlation with a medium effect between how familiar the recommended songs are and how much the participants like them, for the distributional similarity recommender system, $r\ (125) = .45$, $p > .01$. For Spotify the effect was even larger, $r\ (125) = .50$, $p > .01$.

A separate analysis of how diverse the recommendation sets from the two systems are in terms of number of artists included was also conducted. The data from all the compiled playlists (N = 25) show that the Spotify recommendation sets include a much smaller average number of artists (M = 3.5; SD = 1.3) compared to the distributional similarity recommender which provides a more diverse selection of artists (M = 5.84; SD = 0.4).

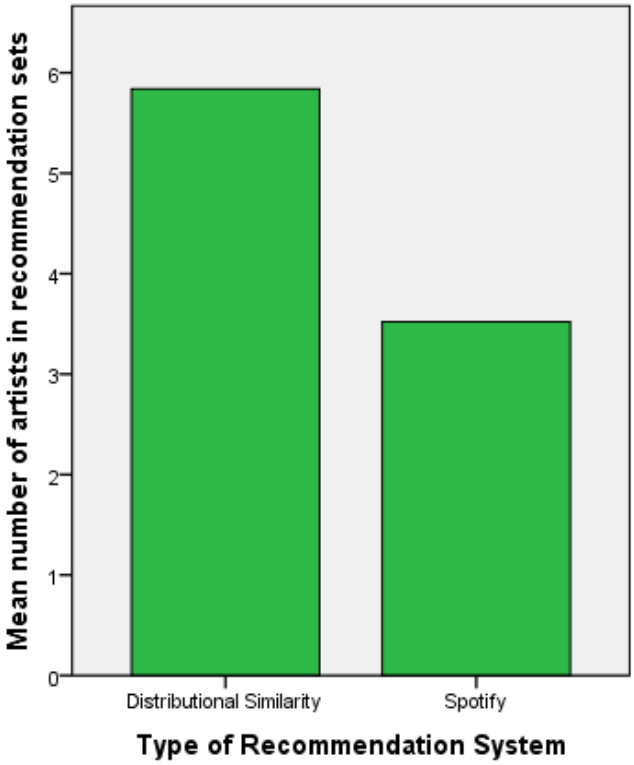

Figure 7: A bar chart showing the mean number of artists in the recommendation sets of the two recommendation systems.

Discussion

Based on the results, of the three hypotheses it could be concluded that Hypothesis 1 (the distributional similarity recommender system would provide more novel recommendations than Spotify) and Hypothesis 3 (the distributional similarity recommendation system would provide more diverse recommendations than Spotify) are true. Hypothesis 2, however, which stated that the distributional similarity recommender system would provide more serendipitous recommendations than Spotify, is false.

The differences in the values for Question 1 and Question 4 show that the distributional similarity recommender system provides users with more novel recommendations which they are less likely to find on their own compared to the Spotify output. This partially relates to the previously introduced concept of serendipity, a serendipitous discovery being described by the Miriam-Webster dictionary as '*finding valuable or pleasant things that are not looked for*'. If it is to be assumed that the idea of something not being looked for is close to the idea of one not being able to find it on their own, this gives a partial advantage for the distributional similarity recommender system in terms of the concept of serendipity.

It is, however, only partial, because for a discovery to be deemed as serendipitous, it should also be 'valuable or pleasant' or in the current questionnaire's wording, it should be liked. The mean values for Question 2 which deals with how much the participants like the recommendation sets do not show that the more novel and more difficult to find on one's own items from the distributional similarity recommender system are more liked than the more familiar and less difficult to find on one's own items from Spotify.

Recommendations from the distributional similarity recommender system were evaluated by the participants in the study as being more novel and more difficult to find on their own than those from Spotify. They were also more diverse in terms of mean number of artists which is generally considered an advantage for a recommender system since this way it is capable of broadening its users' tastes. In this sense the distributional similarity recommendation system successfully manages

to address some of the disadvantages of collaborative filtering systems (Spotify being one such system) discussed in previous sections.

Despite this advantage that the distributional similarity recommender possesses compared to Spotify, the results from the study clearly show that even though the recommendations in the distributional similarity condition were more novel and more diverse, they were less liked and were considered a worse fit to the respective seed song compared to the recommendations in the Spotify condition according to the users.

This suggests that such a system as the one used in the distributional similarity condition is not capable of replacing the sophisticated architecture implemented by Spotify, at least from the point of view of users' enjoyment which was the main focus of this study.

Nevertheless, the results still show that the principle of distributional similarity is a viable option for a recommender system, if not as a stand-alone solution, then as a part of a more sophisticated hybrid system.

Limitations

The current study has several limitations. The first one lies in the failure of the questionnaire to provide the participants with the option to choose the seed songs by themselves and thus create a unique and personalized experience. Instead of being completely free to choose whatever song and genre they preferred, the participants were confined to a set of songs which was chosen for its perceived universality.

However, when asking one to choose the song they are the most familiar with, they will in all cases make a choice but the degree of familiarity could never be certain. If the chosen seed song is not one that the participant truly likes and knows well, this could definitely affect the results further in the questionnaire. The need to compromise the degree of personalization was directed by the constraints of both the dataset which was used and of the time allocated for the completion of this thesis.

Another related limitation is that the questionnaire did not distinguish people in terms of how often they listen to music. According to a 2006 research titled the Phoenix 2 Project, people could be classified in four groups by the degree of interest in music they possess (Jennings, 2007 as cited in Celma, 2010). These groups are called savants, enthusiasts, casuals and indifferents with the degree of interest in music diminishing from the first to the latter (Jennings, 2007 as cited in Celma, 2010).

This distinction between the different types of users is important since the different types would seek various qualities in a recommender system (Celma, 2010). For example a savant would possibly rank a system like the distributional similarity based one, which provides more novel recommendations, higher, than one which provides more familiar recommendations. The explanation is that this type aims at discovering new music all the time, while people who fall in the 'indifferents' category would possibly prefer a more familiar recommendation set.

Conclusion

The results from the conducted study show that despite its relative simplicity the distributional similarity recommender system is capable of overcoming some of the disadvantages of a commercial system like Spotify in terms of the investigated metrics of novelty and diversity but not in terms of serendipity.

When evaluating a recommender system how accurately its algorithms work is important but arguably, even more important is how users perceive the system and the level of enjoyment that it brings to them while using it. Users' evaluation is important because the users are those for whom these systems are being built. Even if a system's algorithm passes all the stringent accuracy tests, if the people it was intended for do not enjoy using it, its worth stays within the confines of theoretical implications.

Enjoyment is an extremely subjective matter especially in the music domain since deriving satisfaction from a piece of music depends on many different factors some of them external to the music itself such as mood in the current moment, social setting, even the time of day. In this sense, even though the distributional similarity recommender takes the lead on some of the measured constructs, namely novelty and diversity, it fails to deliver value on the most important one which is general enjoyment of the recommended songs.

References

Adomavicius, G. & Tuzhilin, A. (2011). Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B. (Eds.), *Recommender systems handbook* (pp. 217–253). New York: Springer, doi:10.1007/978-0-387-85820-3_7

Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A. (2011). Context-aware recommender systems. *AI Magazine*, 67-80. http://dx.doi.org/10.1609/aimag.v32i3.2364

Amatriain, X., Jaimes, A., Oliver, N., Pujol, J. M. (2011). Data mining methods for recommender systems. In Ricci, F., Rokach, L, Shapira, B. & Kantor, P. B. (Eds.), *Recommender systems handbook* (pp. 39–71). New York: Springer, http://dx.doi.org/10.1007/978-0-387-85820-3_2

Bernhardsson, E. (2013). Collaborative filtering at Spotify (PowerPoint slides). Retrieved from http://www.slideshare.net/erikbern/collaborative-filtering-at-spotify-16182818

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993-1022.

Brynjolfsson, E., Hu, Y. J., & Smith, M. D. (2006). From niches to riches: The anatomy of the long tail. *Sloan Management Review*, *47*, 67-71.

Burke, R. (2000). Knowledge-based recommender systems. *Encyclopaedia of Library and Information Systems*, *69* (32), 175-186.

Burke, R. (2007). Hybrid web recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.). *The adaptive web* (pp. 291-324). Berlin: Springer, http://dx.doi.org/10.1007/978-3-540-72079-9_12

Celma, O. (2010). *Music recommendation and discovery: The long tail, long fail, and long play in the digital music space*. Berlin: Springer, http://dx.doi.org/10.1007/978-3-642-13287-2

Chen, S., Xu, J., & Joachims, T. (2013). Multi-space probabilistic sequence modeling. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 865-873, http://dx.doi.org/10.1145/2487575.2487632

Chen, S., Moore, J. L., Turnbull, D., & Joachims, T. (2012). Playlist prediction via metric embedding. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 714-722, http://dx.doi.org/10.1145/2339530.2339643

Desrosiers, C. & Karypis, G. (2011). A comprehensive survey of neighbourhood-based recommendation methods. In Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B. (Eds.), *Recommender systems handbook* (pp. 107–144). New York: Springer, http://dx.doi.org/10.1007/978-0-387-85820-3_4

Felfering, A., Friedrich, G., Jannach, D. & Zanker, M. (2011). Developing constraint-based recommenders. In Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B. (Eds.), *Recommender systems handbook* (pp. 187–215). New York: Springer, http://dx.doi.org/10.1007/978-0-387-85820-3_6

Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. *Proceedings of the Fourth ACM Conference on Recommender Systems,* 257-260, http://dx.doi.org/10.1145/1864708.1864761

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, http://dx.doi.org/10.1145/963770.963772

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2011). *Recommender systems: an introduction*. Cambridge: Cambridge University Press.

Johnson, C. (2014). Algorithmic music recommendations at Spotify (PowerPoint slides). Retrieved from http://www.slideshare.net/MrChrisJohnson/algorithmic-music-recommendations-at-spotify

Koren, Y. & Bell, R. (2011). Advances in collaborative filtering. In Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B. (Eds.), *Recommender systems handbook* (pp. 145–186). New York: Springer, http://dx.doi.org/10.1007/978-0-387-85820-3_5

Lam, C. & Yan, B. (2001) The Internet is changing the music industry. *Communications of the ACM, 44* (8), 62-68, http://dx.doi.org/10.1145/381641.381658

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104* (2), 211, http://dx.doi.org/10.1037/0033-295X.104.2.211

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25* (2-3), 259-284, http://dx.doi.org/10.1080/01638539809545028

Lops, P., de Gemmis, M., Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B. (Eds.), *Recommender systems handbook* (pp. 73–105). New York: Springer, http://dx.doi.org/10.1007/978-0-387-85820-3_3

Magaudda, P. (2011). When materiality 'bites back': Digital music consumption practices in the age of dematerialization. *Journal of Consumer Culture, 11* (1), 15-36, http://dx.doi.org/10.1177/1469540510390499

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press, http://dx.doi.org/10.1017/CBO9780511809071

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. *Paper presented at CHI'06 Conference on Human Factors in Computing Systems, Quebec, Canada, 22-27 April* (pp. 1097-1101). New York: ACM.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:1301.3781.

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. arXiv:1309.4168.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L. & Welling, M. (Eds.) *Advances in Neural Information Processing Systems*. *Paper presented at Neural Information Processing Systems Conference, Nevada, United States, 5-10 December* (pp. 3111-3119). Cambridge, Massachusetts: MIT Press

Mikolov, T., Yih, W. T., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. *Paper presented at HLT-NAACL, Atlanta, United States, 10-12 June* (pp. 746-751).

Moore, J. L., Chen, S., Joachims, T., & Turnbull, D. (2012). Learning to Embed Songs and Tags for Playlist Prediction. *Proceedings of ISMIR 2012,* 349-354

Nielsen. (2014, July 3). Nielsen Entertainment & Billboard's 2014 Mid-Year Industry Report. Retrieved from http://www.nielsen.com/content/dam/corporate/us/en/public%20factsheets/Soundscan/nielsen-music-2014-mid-year-us-release.pdf

Oard, D. W., & Kim, J. (1998, July). Implicit feedback for recommender systems. *Proceedings of the AAAI Workshop on Recommender Systems,* 81-83

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. *Proceedings of the 10th International Conference on Intelligent User Interfaces,* 167-174

Parameswaran, A. G., Koutrika, G., Bercovitz, B., & Garcia-Molina, H. (2010, June). Recsplorer: recommendation algorithms based on precedence mining. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data,* 87-98. http://dx.doi.org/10.1145/1807167.1807179

Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., & Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. *Proceedings of the 7th International Conference on Intelligent User Interfaces,* 127-134

Riordan, B., & Jones, M. N. (2011). Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *Topics in Cognitive Science, 3* (2), 303-345. http://dx.doi.org/10.1111/j.1756-8765.2010.01111.x

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Linguistics, 20* (1), 33-54.

Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.). *The adaptive web* (pp. 291-324). Berlin: Springer. http://dx.doi.org/10.1007/978-3-540-72079-9_9

Setty, V., Kreitz, G., Vitenberg, R., van Steen, M., Urdaneta, G., & Gimåker, S. (2013). The hidden pub/sub of Spotify. *Proceedings of the 7th ACM International Conference on Distributed Event-based Systems*, 231-240. http://dx.doi.org/10.1145/2488222.2488273

Sinha, R., & Swearingen, K. (2001). The role of transparency in recommender systems. *Paper presented at CHI'02 Conference on Human Factors in Computing Systems, Minneapolis, United States, 20-25 April,* 830-831, New York: ACM.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 384 – 394

Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. *Proceedings of the Fifth ACM Conference on Recommender systems,*109-116. http://dx.doi.org/10.1145/2043932.2043955

Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., & Jambor, T. (2012). Auralist: introducing serendipity into music recommendation. *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, 13-22. http://dx.doi.org/10.1145/2124295.2124300

Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. *Proceedings of the 14th international conference on World Wide Web*, 22-32. http://dx.doi.org/10.1145/1060745.1060754

Appendix A

Type of RS	Advantages	Disadvantages
Collaborative filtering	1. No need for content, features. 2. Quality improves over time. 3. Personalized.	1. Dependent on other users' activity. 2. Black boxes. 3. Cold start problem for users and items. 4. Ignores the items in the Long Tails. 5. Privacy concerns.
Content-based filtering	1. User independence. 2. Transparency. 3. No new item problem. 4. Personalized.	1. Limited content analysis. 2. Domain knowledge needed. 3. Over-specialization. 4. New user problem.
Knowledge-based filtering	1. No cold-start problem. 2. Sensitive to preference changes. 3. Independent of other users and ratings of items.	1. Dependent on domain knowledge. 2. Knowledge engineering.
Context-aware	Makes use of contextual data.	Recommendation list may become too narrow.
Hybrid	Tackles the problem of data scarcity.	Some of the types do not perform better than single recommender algorithms.
Distributional similarity	1. Independent of other users. 2. No need for domain knowledge.	1. Lacks transparency. 2. Recommendations depend on the dataset.

Table 1: Advantages and disadvantages of various recommender systems.

Appendix B

The aim of the present survey is to evaluate the quality of music recommendations provided by two recommendation systems. Your contribution would be much appreciated.

Please take some time to listen to the recommended songs in case you're not familiar with them. They are presented as a YouTube playlist so you can easily navigate forward and backward with the corresponding buttons.

The survey will not take more than 20 minutes of your time.

Enjoy!

****

Please choose one of the following songs which you are the most familiar with.

You can listen to the songs on YouTube by clicking on the titles.

Roy Orbison – Oh, Pretty Woman

The Police – Every Little Thing She Does is Magic

Madonna – Holiday

John Lennon – Imagine

Marvin Gaye – Sexual healing

Michael Jackson – Billie Jean

Radiohead – Creep

***

Please listen to the following set of songs that were recommended by one of the systems for the song chosen by you previously and answer the questions below:

Example Playlist 1

Question	Not much	Little	Somewhat	Much	A Great Deal
How familiar would you say you are with the recommended songs?					
How well do the recommended songs fit the song chosen by you?					
How much do you like the recommended songs?					
How likely are you to find the recommended songs on your own?					

***

Now listen to another set of recommendations provided by a different system for the same song you chose in the beginning and answer the same questions below:

Example Playlist 2

Question	Not much	Little	Somewhat	Much	A Great Deal
How familiar would you say you are with the recommended songs?					
How well do the recommended songs fit the song chosen by you?					

How much do you like the recommended songs?					
How likely are you to find the recommended songs on your own?					