FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 48 (2023) No. 2
ISSN 0867-6356
sciendo DOI: 10.2478/fcds-2023-0009 e-ISSN 2300-3405

Content-Based Recommender Systems Taxonomy

Harris Papadakis[1], Antonis Papagrigoriou[1], Eleftherios Kosmas[1]
Costas Panagiotakis[2], Smaragda Markaki[2], Paraskevi Fragopoulou[1] *

Abstract. In the era of internet access, recommender systems try to alleviate the difficulty consumers face while trying to find items (e.g. services, products, or information) that better match their needs. To do so, a recommender system selects and proposes (possibly unknown) items that may be of interest to some candidate consumer, by predicting her/his preference for this item. Given the diversity of needs between consumers and the enormous variety of items to be recommended, a large set of approaches have been proposed by the research community. This paper provides a review of the approaches proposed in the entire research area of content-based recommender systems, and not only in one part of it. To facilitate understanding, we provide a categorization of each approach based on the tools and techniques employed, which results to the main contribution of this paper, a content-based recommender systems taxonomy. This way, the reader acquires a quick and complete understanding of this research area. Finally, we provide a comparison of content-based recommender systems according to their ability to efficiently handle well-known drawbacks.

Keywords: content-based systems, recommendation systems, survey, taxonomy

1. Introduction

Given the global nature of the Internet access currently available, producers and service providers have the unique opportunity to advertise to a large number of customers all over the globe. At the same time, the number of choices available to the customers has vastly increased. Even though this allows for a multitude of possibilities and a wider variety of selection, some important implications arise. On one hand, it became increasingly difficult for producers and providers to increase the efficiency of

*[1]Department of Electrical and Computer Engineering, Hellenic Mediterranean University, 71004 Heraklion, Greece, [2]Department of Management Science and Technology, Hellenic Mediterranean University, 72100 Agios Nikolaos, Greece {adanar,apapa,ekkosmas,cpanag,smarkaki}@hmu.gr, fragopou@ics.forth.gr

their advertisements. Given the diversity of the customers' needs, it became harder to select and target only the customers, whose preferences match particular products. On the other hand, it is not easy for consumers to identify the appropriate products or services that better match their interests, given the diversity and vast availability of options and their time limitations. Both of these facts result in situations where interesting products/services may pass completely unnoticeable to them.

Recommender systems (RS) try to amend this situation by analyzing consumer preferences and trying to predict the preference of a user for a single item (e.g. product or service) [19, 48, 50, 81]. RS have a wide range of applications. RS are well known for their use in applications with a vast range of products. The main use case of the Recommender System in such an environment is to search and select a targeted product or service information for a given customer. Another application can be found in designing marketing strategies, where RS are used to predict the popularity of products. Additionally to traditional consumer items, RS are also used to provide users with recommendations for other entities, such as web pages, relevant or interesting information, etc.

The main purpose of the RS is to predict the degree of preference of a user for an item. This, in turn, enables the recommendation of items with a high probability of acceptance by the user. In essence, information is filtered and provided to the user in a more dynamic and automated fashion, compared to database queries.

Recommender Systems can be divided into two main categories, Collaborating-Filtering and Content-based Recommender Systems. In particular, Content-based RS rely on analyzing items (content) or descriptions of items, to build item representations, that can be used to recommend new items to users by matching them to corresponding user profiles [44]. This is in contrast to Model-based systems which rely mainly on ratings and are to a large degree agnostic of the characteristics of the recommended items. The recommendation process consists in matching up the attributes of the user with the attributes of an item. The result is an appraisal that represents the user's level of interest for that item [52], which results in a recommendation for the user. The main purpose of this survey paper is to present recent techniques used in the context of Content-based RS.

Particularly, in Section 2 a classification of the Content-based RS techniques is given and a detailed description of them in Sections 3 to 6. In Section 7 we provide a comparison of the presented approaches according to whether or not each of them addresses well-known drawbacks, which arise when designing and implementing a Content-based recommender system. Finally, we present our conclusions in Section 8.

2. Content-based recommender systems classification

The nature of the recommendation problem allows for a diverse set of approaches and thus, several systems have been proposed to achieve the aforementioned goal. In this work, we present some of the most recent and/or more impactful approaches. Our goal is to provide a review of the systems proposed in the whole research area of Content-based RS and not only on one part of it so that the reader acquires a quick

and complete understanding of this research area. For this reason, we focus mainly on presenting the key ideas/techniques used to solve the recommendation problem. More specifically, for each approach, we present an overview of its most representative systems, by selecting some of them and we avoid presenting most of their implementation details. Additionally, we assist the readers that may be interested in finding more details on some of these approaches (or systems), by providing a large number of references.

In order to facilitate a better understanding of the various approaches to the recommendation problem, we provide a categorization of each approach based on the tools and techniques employed, which results in the main contribution of this paper that is a Content-based RS taxonomy. We remark that several hybrid approaches may fall into more than one category. Nevertheless, each of the described approaches has been assigned to the Content-based recommendation system category, the best-fitted one, according to our understanding.

Content-based RS analyze items (content) or descriptions of items, to build item representations that can be used to recommend new items to users [44]. The recommendation process consists in matching up the attributes of the user profile against the attributes of a content item. The result is an appraisal that represents the user's level of interest for that item [52], which results in a recommendation for the user.

In the following, we briefly describe the subcategories of each category of Content-based systems, used in our taxonomy:

1. **Bi-polar sentiment analysis.** In bi-polar sentiment analysis, the basic sentiment classes of positive and negative are used. In some polarity sentiment analysis methods, there is also a neutral class, apart from the positive and negative ones, while in others this class is ignored. The neutral class can be thought of as a measure of the objectivity of the analyzed item, as it means that no positive or negative sentiment can be inferred from it. In most RS the bi-polar sentiment analysis is used in conjunction with another recommendation technique to obtain more accurate recommendation results.

2. **Multidimensional sentiment analysis.** In terms of functionality in RS, the bi-polar sentiment analysis and multidimensional sentiment analysis are similar. The main difference is that multidimensional sentiment analysis, aims to extract sentiment (opinion polarity) in a higher dimensionality and not just on a binary scale (positive vs. negative). Multidimensional sentiment analysis extracts sentiment in vectors of various polarized states, such as "like/dislike", "happy/sad", "accept/reject", etc.

3. **Term Frequency/Inverse Document Frequency (TF-IDF).** The most common use of TF-IDF is to produce training data for document-based recommendation systems, i.e. Content-based RS for which the source of information is mainly text documents. TF-IDF combines the TF method with IDF. In TF-IDF each document is represented as a vector with n dimensions, where each dimension corresponds to the weight of a term, according to the frequency of that term in the document (TF). However, in order to measure the term

Figure 1. Hierarchical presentation of Content-based recommender system categories.

specificity, IDF is used. As the number of documents in the collection that contain a term becomes larger, the weight of this term becomes smaller. Usually, a regression-based model is used to produce recommendations from TF-IDF training data.

4. **Abstract semantic graph (term graphs).** The abstract semantic graph in a recommendation system is a weighted graph, which indicates and incorporates the "strength" of the connection between two vertices as a weight added to the corresponding edge that connects the two vertices. We can find three common graph types: item-to-item, user-to-user, and item-to-user. By combin-

ing results from walks in these graphs a recommendation system can provide recommendations.

5. **Matrix factorization/LDA.** Latent Dirichlet Allocation (LDA) is a generative statistical model technique for discovering and exploiting the hidden thematic structure in large archives of text, where each topic probability distribution is assumed to have a sparse Dirichlet (after Peter Gustav Lejeune Dirichlet) prior probability distribution [42]. In recommendation systems, LDA is used to find the latent relations between keywords of item descriptions and item tags created by users, so that items can be recommended based on their tags. LDA can be used in conjunction with other recommendation algorithms like (multi-dimensional) sentiment analysis.

6. **Matrix factorization/LSA.** Latent Semantic Analysis (LSA) is a technique of analyzing relationships between a set of documents and the terms they contain, by producing a set of concepts related to these documents and terms. In recommendation systems, this is used to recommend a new document. LSA uses dimension reduction to establish term correlations, thus, allowing users to compare documents and queries for similar topics. LSA can also be used in conjunction with other recommendation algorithms like sentiment analysis.

7. **Conversational Recommender Systems.** Conversational Recommender Systems use natural language-based interaction with the user in order to fine tune and provide more desirable recommendations. The main difference to more traditional RS is the user's more active participation during the recommendation process, a fact that is both an advantage and a drawback, since it can lead to higher recommendations quality but assumes the user's time and willingness to participate.

8. **(Social) tagging systems.** Social tagging systems are Web 2.0 applications concerned with the publication and tagging of web resources by ordinary internet users. This gives users the flexibility to explore tags, resources, or even other users' profiles, without being bounded by any rigid predefined conceptual hierarchy [75]. There are many algorithms for recommendation systems that use social tagging to produce recommendations. From simple ones, like constant tagging, where the most frequent tags of social tagging are recommended, to more complex ones, like automated tag clustering and hierarchical tag clustering.

9. **Knowledge graphs.** Knowledge graphs are large networks of entities and their semantic relationships. In knowledge graphs, nodes represent topics/data and edges represent relations between them. For this reason, knowledge graphs can be used to represent heterogeneous information that is incorporated from several information source types. In recommendation systems, a calculation method, like the spreading activation-based technique, is used over the nodes of the knowledge graph to obtain a rating. Then, the rating itself is used to obtain the recommendation.

3. Sentiment Analysis and Opinion Mining Content-based Systems

Although sentiment analysis and opinion mining have been techniques well-known from database theory, they came into the center of attention with the emergence and wide popularity of social networking and micro-blogging sites [18, 71]. These techniques involve natural language processing, and text analysis and sometimes incorporate advanced voice, image, or video processing techniques. There are several, well-known, open-source tools based on natural language processing for sentiment analysis. Sentiment analysis was further facilitated by the use of emoticons and other standard symbols/expressions [82] that accompanied the evolution of social networks, which allow the explicit manifestation of consumer sentiment (permit consumers to explicitly and concretely manifest/declare their sentiment).

Sentiment analysis has been successfully integrated into hybrid RS, to enhance the prediction power of traditional approaches, like Collaborative and Content-based Filtering, or hybrid approaches, which are based on raw data provided either by consumers or by producers. Collaborative Filtering is based on the predictive power of a rating matrix [67], that consolidates explicit consumer provided ratings, while Content-based Filtering [4] extracts conclusions from metadata of products. Metadata is usually provided by product companies and, thus, purely Content-based approaches are not considered consumer driven recommendations. In many online stores and services, users are given the possibility to provide explicit feedback for items. These explicit reviews are rich in user sentiment and allow to more easily extract features/aspects of products. While metadata is company produced, product features/aspects extracted from the analysis of consumer reviews constitute user defined metadata, or refined versions of item defined metadata. Sentiment analysis can be applied to user reviews and can be seen as users' rating scores on the corresponding features/aspects extracted from these reviews.

In a more general context, opinion mining tools can be integrated into RS that process sets of more broad search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregate opinions/extract sentiment about each of them (poor, mixed, good or in a more refined scale). Popular approaches of opinion-based RS utilize various techniques including text mining, information retrieval and sentiment analysis (multimodal sentiment analysis), or even the cross-integration of consumer site with social networking and micro-blogging sites. In these RS, the opinions/sentiments are transformed into a numerical evaluation and are fed into the recommendation system to implement the recommendation, using some other recommendation technique, like Collaborative Filtering [88] or like topic models [87].

3.1. Bi-polar Sentiment Analysis

In Bi-polar sentiment analysis, the basic sentiment classes of positive and negative are used. In some polarity sentiment analysis methods, there is also a neutral class, apart from the positive and negative one, while in others this class is ignored. In most

RS the Bi-polar sentiment analysis is used in conjunction with another recommender system technique, in order to obtain a better recommendation result. For example, [88] used the sentiment polarity analysis on a dataset of opinions, obtained from the reviews extracted from a recommended dataset, which was obtained using Collaborate Filtering. Then, follows the extraction of the features from the reviews and the use of polarity analysis on the features, to obtain the polarity scores. These scores are used as input to further refine the recommendation result dataset for the user and provide more accurate recommendations. This is shown in Figure 2. Some of the known techniques that incorporate bi-polar sentiment analysis from consumer reviews intended for use in RS are analyzed below.

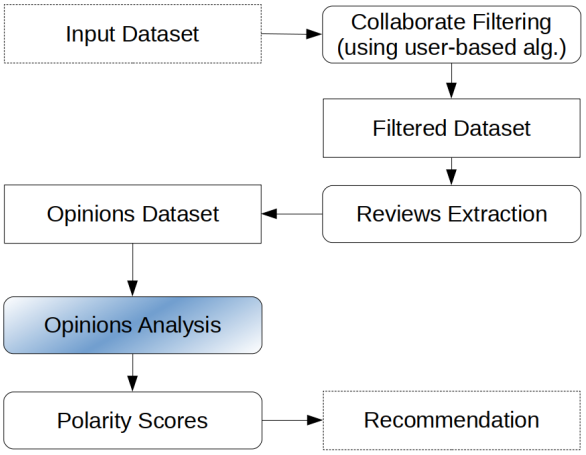

Figure 2. An example of a process for recommendation system using opinions analysis to refine results.

Preliminary work on the semantic classification of product reviews has been conducted by [22, 28]. The authors train a classifier in order to identify appropriate features and scoring methods for determining whether reviews are positive or negative. In order to extract sentiment from reviews (thus classifying them as either positive or negative), the authors initially use structured reviews from two specific major websites (namely Amazon and C—net) to train and test a classifier. Structured reviews contain binary or quantitative ratings, and small and relatively precise text, while products contain useful metadata. Such datasets are ideal to train and test a classifier that extracts product features and scoring, and assigns positive or negative sentiment to reviews. The classifier is subsequently applied to less structured texts mined from broad web searches. Starting with a raw piece of text, a preprocessing phase is applied which is based on word extraction, metadata, statistical and linguistic substitutions, and language based modifications (n-grams, substrings), in order to achieve an effective feature extraction. After an appropriate set of features are selected, these features are assigned scores based on term frequencies, in order to place test documents in the set of positive or negative reviews. To assign scores

to features, a number of methods were tested like machine learning, SVM, and a Naïve Bayes with Laplace classifier, but none of these provided perfect results across all tests. More consistent performance for all tests with less computation overhead was obtained when various calculated frequencies and techniques from information retrieval were used. Once each feature has a score, the sum of the scores of the words in an unknown document is calculated and used to determine a class, thus being characterized as positive or negative.

[54] introduce OPINE, an unsupervised information extraction system that mines reviews in order to build a model of important product features, their evaluation by reviewers, and their relative quality across products. OPINE uses the novel technique of relaxation labeling, known from computer vision, to find the semantic orientation of words in context. It leads to strong performance on the tasks of finding opinion phrases and their polarity. Initially, the problem of review mining is decomposed into the following main subtasks:

i) identify product features,

ii) identify opinions regarding product features,

iii) determine the polarity of opinions,

iv) rank opinions based on their strength.

OPINE embodies a solution to each of these subtasks. Novel components include the use of relaxation labeling to find the semantic orientation of words in the context of given product features and sentences. OPINE reports its precision and recall on the tasks of opinion phrase extraction and opinion phrase polarity determination in the context of known product features and sentences.

[41] focuses on the standard sentiment analysis model for product reviews which consists of three steps, data preparation, review analysis and sentiment classification, describing representative techniques for each of them. The main purpose of the review analysis step is to perform the necessary text processing in order to extract interesting information such as opinions on products and features from the processed reviews. Regarding review analysis, two major approaches from the literature are examined, namely the Sentiment Orientation (SO) and the machine learning approach. The SO approach initially determines the sentiment orientation of the individual opinions extracted during the review analysis step, and subsequently determines the sentiment orientation (either positive or negative) of an entire review. Some methods are based on the identification of semantic similarity between words or phrases to predict SO by comparing (semantically) extracted text against a small set of predefined seed words with known SO, while automatically expanding the predefined set of words with synonyms and antonyms. These methods are based on the assumption that semantically similarity implies sentimental similarity. Following the prediction of SO, reviews are classified by most methods as positive or negative, while other methods try to infer review ratings (e.g. on an 1 to 5 scale). The machine learning approach is similar to topic classification, with the topics being sentiment classes such as positive and negative. Machine learning approaches work by breaking down reviews into

words or phrases (bag-of-words model), and then classifying them. Standard topic classification techniques such as Naïve Bayes, Support Vector Machines (SVM) and Maximum Entropy have been applied without giving impressive results [49]. The use of n-ary classifiers, such as SVM classification, for multi-class classification has also been attempted. The paper emphasizes the need for a paradigm shift from binary classification (positive vs. negative) to multi-point rating inference (e.g. in an 1 to 5 rating scale), opening up an interesting direction toward ordered multi-category classification. Furthermore, it identifies the potential integration of sentiment analysis to enhance Collaborative Filtering approaches.

3.2. Multidimensional Sentiment Analysis

In terms of functionality in RS, the Bi-polar sentiment analysis and Multidimensional sentiment analysis are similar. The main difference is that multidimensional sentiment analysis aims to extract sentiment (opinion polarity) in a higher dimensionality (ie: larger number of emotions) and not only like/dislike. Multidimensional sentiment analysis extracts sentiment in vectors of various polarized states like "like/dislike", "happy/sad", "accept/reject", etc. An opinion is a quintuple, *(e, a, so, h, t)*, where e is the name of an entity, a is an aspect of e, so is the orientation of the opinion about aspect a of entity e, h is the opinion holder (the person or organization who holds the opinion), and t is the time when the opinion is expressed by h [5]. In Multiclass sentiment analysis the opinion orientation so of a single emotion is expressed with different strength/intensity levels, but in Bi-polar sentiment analysis the opinion orientation so can have one of two (sometimes three) possible values: positive, negative (or neutral). A variety of such models, with a different group of sentiments each, have been proposed in the literature. Some of these are reviewed below.

[27] studies a Bayesian modeling approach to multiclass sentiment classification and multidimensional sentiment distribution prediction. It proposes effective mechanisms to incorporate supervised information such as labeled feature constraints and document-level sentiment distribution derived from the training data into model learning expanding on previous work of mapping sentiments or emotions into multiple dimensions. Classifying text into multiple emotion categories can be cast as a multiclass single-label classification problem. The results demonstrate that using the latent representation of the training documents derived from the aforementioned approach as features to build a maximum entropy classifier outperforms other approaches to multiclass sentiment classification.

[82] present a multidimensional sentiment analysis of tweets method. It is based on a combination of a sentiment analysis and classification of the verbal part of tweets and how this is altered (enhanced or softened compared to the original sentiment) by emoticons included in certain tweets. This was identified as an important problem, as an increasing percentage of tweets possess emoticons, which is the only non-verbal form of expression allowed on twitter. The multidimensional analysis of the verbal parts of tweets was based on the ten scale sentiment classification of Nakamura for Japanese tweets, which was altered to define five bipolar scales as follows: Sorrow-Joy,

Dislike-Liking, Shame-Relief, Fear-Anger, Surprise-Excitement. Emoticon's effect on tweets was classified as "Emphasis", "Assuagement", "Conversion", or "Addition". A sentiment lexicon for words and an emoticon lexicon were compiled from movie review data collected from Yahoo, enriched with results from user-centric experiments. The system created a sentiment lexicon based on calculating the co-occurrence frequency between sentiment words that are defined in advance and numerous words in documents. The lexicon consists of sentiment dimensions, words categorized by each dimension, and a sentiment value for each word. Compiling an emoticon lexicon was a nontrivial task as there is a huge variety of emoticons and users can create new ones freely. The proposed method determines the sentiment of a tweet based on the sentiment of the sentence in the tweet and the role of the emoticon following the sentence. The relation of sentiment between sentences and sentences with emoticons was formalized using regression analysis.

In a somewhat different direction, [76] present a new type of recommendation explanations which is called Tagsplanation. It is a feature-based approach that uses tags (features of the recommended items) as intermediate entities. For example, a movie recommendation system could use movie features like genre, director, cast, etc. to justify recommendations. A keyword-style approach is adopted for tags. Two aspects of tag-based explanations are identified: the relationship of the tag to the recommended item, which is called tag relevance, and the relationship of the user to the tag, which is called tag preference. Tag preference is highly relevant to the user's sentiment for a given tag. A user's preference for a tag is computed in one of two ways. It is either directly assessed, by asking the user's direct opinion, or it may be indirectly inferred based on the user's behavior. To estimate a user's preference for a tag, a weighted average of the user's ratings for movies with that tag, on a 5-star scale, is computed. The 5-star scale provides a finer grade scale, yet the information is easy to interpret since it follows the standard movie rating scale. The experimental evaluation of the proposed method is performed on a Movie-Lens dataset which was released in 2006.

4. Vector Space Model Content-based Systems

In Content-based RS, the recommendation is based on the correlation between the content (attributes) of the items and the user's preferences. There are several ways in which the attributes of the items can be represented in order to be used as a basis for a Content-based recommendation system. A way of representing each item is as an m-dimensional vector, where each dimension of the vector corresponds to a distinct attribute in correlation with the user preferences; m is the total number of attributes used in the collection of items. The representation of a set of items as vectors in a common vector space is known as the vector space model [13] and it is shown in Figure 3. An example of the usage of vector space modeling in recommender systems is a recommendation engine for movies, where the preferences of a user are correlated with the attributes of the movies like movie genre, actors, directors, etc. (for a total of n attributes per movie) in an n-dimensional vector, as shown in Figure 4. Another

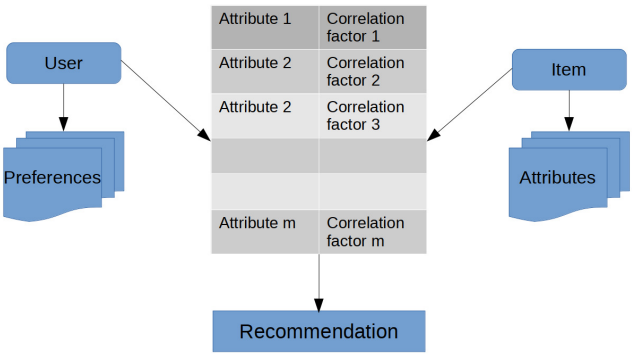

Figure 3. Schematic representation of a Vector Space Model method.

Figure 4. An example of Vector Space Model for a movie recommendation system.

example can be found in document filtering where a document is represented as an m-dimensional vector, where each dimension corresponds to a distinct term in the document and m is the total number of terms used in the collection of documents.

4.1. TF & TF-IDF

One way to represent an item in vector space modeling is to assign weights to the attributes of each item as values for the item vector. This approach is more common in document recommendation systems, where documents are the items and terms are their attributes. The weight of an attribute can be defined in many different ways. For example, in document-based recommendation systems, the simplest approach is to set the weight of each term in a document to the number of its occurrences in the document. This weighting scheme is referred to as Term Frequency (TF) [13]. For a term t in some document d, the TF is denoted as $TF_{d,t}$. TF is equal to 0, when the document does not contain the term; otherwise, it has a different value. In the latter case, TF is measured using the frequency of the term in the document versus

the total number of occurrences of all terms[1] in the document. Specifically, TF can be computed using the following equation [58]:

$$TF_{d,t} = \begin{cases} 1 + log(1 + log(freq_{d,t})), & \text{if } freq_{d,t} \neq 0. \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $freq_{d,t}$ is the occurrence frequency of term t in the document d (ie: number of occurrences of term t divided by number of occurrences of all terms in d).

The drawback of term frequency, e.g. in document based recommendation, is that it does not account for how important each term is for a document. So, terms, which are commonly used in the entire corpus of documents have statistically larger term frequencies than terms, which are not commonly used. However, since they are common in many documents, they are often incorrectly used to discriminate between documents. To solve this problem, some measure of term specificity is required. One such solution is the Inverse Document Frequency (IDF). The IDF of a term t is denoted by IDF_t. The computation of IDF is based on the number of documents in the collection that contain a certain term. If a term occurs in many documents, then it is not a good discriminator and should be given less weight than other terms that occur in fewer documents [59]. There are several formulas for the calculation of IDF. The basic one has been presented by [35] and it is stated as follows:

$$IDF_t = log\left(\frac{N}{n}\right) \quad (2)$$

where N is the total number of documents in the collection, and n is the number of documents that contain t. Another formula for IDF is presented by [58] and it is the following:

$$IDF_t = \frac{log(1 + |N|)}{|n|} \quad (3)$$

In both cases, each term is weighted during a similarity computation stage and then its weight is used during the recommendation process.

Combining TF and IDF results in the Term Frequency/Inverse Document Frequency (TF-IDF) method, which is used to create a vector space model for a recommendation algorithm. The TF-IDF measure emerged from extensive empirical studies of combining various weighting factors, particularly by the Cornell group [61]. In the TF-IDF method, the TF weight of a term affects as a factor the IDF of the same term.

$$TF/IDF_{t,d} = TF_{t,d} \cdot IDF_t \quad (4)$$

Another approach more useful to Naive Bayes machine learning models[2], was presented by [34]. In this work, the log function, used in the calculation of IDF, is

[1] we remark that an article is a word but not a term

[2] More as a way to classify documents into a fixed number of predefined categories (known as categorization task) rather than ranking them.

replaced with a square-root function. Therefore, the resulting formulation of IDF is significantly different from the traditional IDF. However, this approach works under several assumptions, such as that all the collections in its categorization task, should have approximately the same number of documents.

An example of a recommendation system, that can make use of the above algorithm, is a movie recommendation system that is based on the category tags of the movies. In this case, as terms we have the movies and as term types, we use the category tags of the movies. By doing so, as frequency of the term we can use the ratings of the movies by the users, and as weight of the terms we can use the impact of each category tag of the movie. Usually the TF-IDF method is used to compute the training data for a recommendation system and subsequently, machine-learning algorithms (such as SVM, Rocchio, and Naive Bayes classifiers) usually applied to the training data in order to make the actual predictions [32].

4.2. Abstract Semantic Graph (Term Graph)

Generally, in computer science, an Abstract Semantic Graph (ASG) or Term Graph (TG) is a form of an abstract syntax in which an expression of a formal or programming language is represented by a graph whose vertices are the expression's sub terms [23]. Graph-based algorithms are commonly used in citation analysis, social networks, and the analysis of the link-structure of the World Wide Web, like Google's PageRank [10] and Kleinberg's HITS algorithms [38].

Graph-based algorithms can also be used in recommendation systems, where the abstract semantic graph model is essentially used to determine the importance of a vertex in a graph for the recommendation, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based recommendation model is that of "voting". When one vertex is linked to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of that vertex. Also, the importance of the vertex casting the vote determines how important the votes it casts are. This information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the importance of the vertices casting these votes.

In abstract semantic graph models, weighted graphs can be used to indicate and incorporate into the model the "strength" of the connection between two vertices as a weight added to the corresponding edge that connects the two vertices. Recent popular approaches rely on Vector Space Models (VSM) [60], which use frequency measures in a text corpus in order to leverage semantic information [74]. Other approaches, e.g. TextRank [55] and HashGraph [36] use more complex text extracting algorithms for vertex terms.

In modern Content-based or hybrid recommendation systems the following three common graph types are found:

i) *item-to-item,* where the graph defines the correlation between items. In this case a variety of personalized PageRank methods can be used to perform item

recommendations [24];

ii) *user-to-user,* where the vertices are defined with the notion of horting. Horting is an asymmetric relationship between users. It is defined on the basis that users have rated similar attributes. Using a transformation function, that provides predictions for a user using the hort of other users, a prediction graph is created from which recommendations for the user can be obtained.

iii) *item-to-user,* where the graph is based on the importance of each item for each user. This graph is used to define neighborhoods. Then, random walks are used as in the personalized PageRank method. Starting from a user node in the graph, similar users can be identified. Similarly, starting from an item node, similar items can be identified. Random walks are used to provide recommendations [20].

A recommendation system for a digital bookstore based on abstract semantic graphs is presented in [30]. In this scenario, books are the items and customers are the users in an extended graph that incorporates:

i) book-to-book (item-to-item) correlations in a graph, based on specific book attributes,

ii) customer-to-customer (user-to-user) correlations in another graph, based on user demographic information, and

iii) book-to-customer (item-to-user) correlations, based on the purchase history of each user.

By combining results from random walks on these graphs, the system recommends books to customers.

5. Semantic Content-based Systems

In several existing systems, where most of the information is presented in textual form, the recommendation systems have to deal with huge amounts of unstructured text. In those cases, efficient text mining techniques are required in order to understand these documents and extract important information from them. Traditional term-based or lexical-based analysis cannot capture the underlying semantics. In order to overcome this limitation, semantic-based analysis approaches have been employed [16]. Semantic text analysis approaches are used to provide a conceptual understanding of the documents. Several approaches were proposed based on tagging or ontologies some of which are presented below.

In tagging systems, objects are encoded with keywords that individuals find interesting, so they can be easily retrieved, at a later stage, with the known keywords. Tags are used in RS since 2006 [15]. A recommender system that incorporates tags into its recommendation model is usually referred to as a tag-aware recommender

system. Generally speaking, tags are a way of grouping content by category in order to make it easy to retrieve it by topic [73].

An ontology is a formal explicit description of concepts in a domain of discourse. The properties of each concept describe various features and attributes of the concept and are known in ontology as properties. The restrictions on slots are called facets or role restrictions. The main role of ontologies in the semantic analysis is to map terms to semantic concepts. Ontology concepts are linked together to provide useful semantic relations that can be exploited. The use of ontologies has helped enhance semantic-based analysis, where hierarchies of concepts are built in order to capture conceptual relations between terms [84].

5.1. Conversational Recommender Systems

Conversational Recommender Systems use natural language-based interaction with the user in order to fine tune and provide more desirable recommendations. The main difference to more traditional RS is the user's more active participation during the recommendation process, a fact that is both an advantage and a drawback, since it can lead to higher recommendations quality but assumes the user's time and willingness to participate. Therefore, it makes more sense to be employed as an additional functionality to a more traditional recommender system, especially given the fact that user participation in Conversational Recommender Systems goes a long way in order to alleviate the cold-start problem traditional recommender systems often face.

Such systems require the existence of appropriate techniques for natural language processing, as well as determining the correct questions the user should be asked at each point of the interaction. The user's natural language responses are analyzed and translated to (semantic) structured information in order to generate the desired recommendations. During the multi-turn interaction with the user, several methods can be used in order to refine the provided recommendations. Such methods include choice-based methods, where the user is presented with a pair or list of items and is requested to choose the one he/she prefers as well as Bayesian based methods where the utility function is updated in a Bayesian fashion between turns of the interaction. Neural networks is the most common approach for user's intention and sentiment extraction from the natural language responses.

Conversational Recommender Systems allow users to specify arbitrarily complex preferences in a more natural and efficient manner. An early prototype of a Conversational Recommender Systems is the adaptive place advisor [11]. The natural language input in the adaptive place advisor is primarily limited to providing concrete attribute values in response to questions posed by the system, and as a result the system draws little benefit from it. Another Conversational Recommender Systems is ExpertClerk [68], which receives written natural language as input. The system engages the user in a dialog by asking domain questions until the search space is narrowed down to a sufficiently short list of viable products. Then, the system presents the user with a set of three maximally different options. Further navigation is possible by critiquing

any of the displayed options. A textual natural language recommender system was further discussed by [79], which focuses on optimizing the dialog system, with equally promising results.

5.2. (Social) Tagging Systems

Social tagging systems are Web 2.0 applications concerned with the publication and tagging of web resources by ordinary internet users. These systems transformed users from passive consumers to active producers of content [85]. Social tagging systems are now widespread, with millions of people using them daily to organize and retrieve online content. With the increase in tagging activity, a complex network is created by many annotations. Thomas Vander Wal described the result of personal free tagging of information and objects (anything with a URL) for one's own retrieval as "folksonomy". Folksonomy is created from the act of tagging by the person consuming the information. The value of this external tagging is derived from people using their own vocabulary and adding explicit meaning, which may come from inferred understanding of the information/object. People are not so much categorizing, as providing a means to connect items (placing hooks) providing the meaning in their own understanding [75]. Therefore, a 3-way relationship exists between users, items, and tags.

Formally, a folksonomy is defined as a relational structure $F := (U, R, T, Y)$ in which:

- U, R, and T are disjoint non-empty finite sets whose elements are called users, resources (items), and tags, respectively, and
- Y is the set of observed ternary relations between them, i. e., $Y \subseteq URT$, whose elements are called tag assignments [29].

Folksonomy data can be represented in various ways and different representations stimulate different types of models. The set of tag assignments Y can be represented:

- as a three dimensional binary tensor where 1 indicates observed tag assignments and 0 missing values [70], as shown in Figure 5, or
- as a three dimensional tensor. For each tag a matrix is given that contains the observations of tag assignments of a user for a specific resource. The observed tag assignments are interpreted as positive feedback, whereas the non-observed tag assignments are marked as negative and all other entries are marked as missing values [57], as shown in Figure 6.
- An equivalent, but maybe more intuitive, representation of a folksonomy is an undirected tripartite hypergraph $G_F := (V, E)$, where $V := U \cup R \cup T$ is the set of nodes, and $E := u, r, t|(u, r, t) \in Y$ is the set of hyperedges [43].

Folksonomy gives users the freedom to explore tags, resources or even other user's profiles, unboundly, starting from a rigid predefined conceptual hierarchy. However,

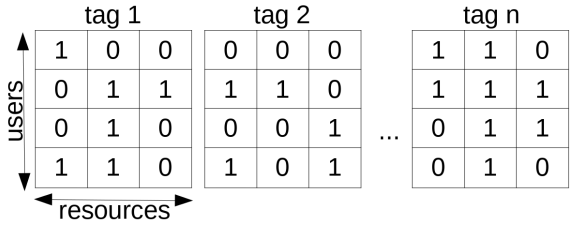

Figure 5. Tensor representation where positive feedback is interpreted as 1 and the rest as 0.

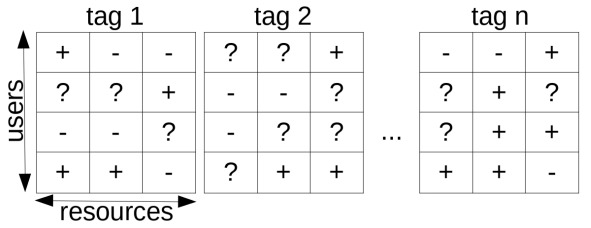

Figure 6. Tensor representation where observed tag assignments are considered positive feedback while non-observed tag assignments are marked as negative feedback. All other entries are missing values.

this freedom comes at a cost, an uncontrolled vocabulary which can result in tag redundancy and ambiguity hindering navigation.

There are many algorithms for recommendation systems using folksonomies. Recommending the most frequent tags of the folksonomy is the most simplistic approach. This method is referred to as constant tagging [43], since it always recommends the same set of tags regardless of the target entities. Alternatively, one can score a tag by counting the frequency of its co-occurrence with a given resource (item) or user. Other, more complex algorithms rely on data mining techniques, such as automated tag clustering [6] or hierarchical tag clustering [66].

5.3. Knowledge Graphs

Knowledge graphs are currently used to explain search results, explore knowledge spaces, semantically enrich textual documents, or feed knowledge-intensive applications such as RS [46]. In RS, knowledge graphs are large networks of entities and their semantic relationships. The nodes of these graphs represent topics/data and the edges represent relations between them. The meaning of the data is encoded alongside the data in the graph in the form of ontologies. An ontology is typically based on logical formalisms, which support some form of inference, allowing implicit information to be derived from explicitly asserted data. Some of the information inferred can be otherwise hard to discover [17]. Thus, the knowledge graph provides a single place

to find the data and understand what it's all about [12]. In a sentence, a knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge as output. A knowledge graph system is shown in Figure 7.

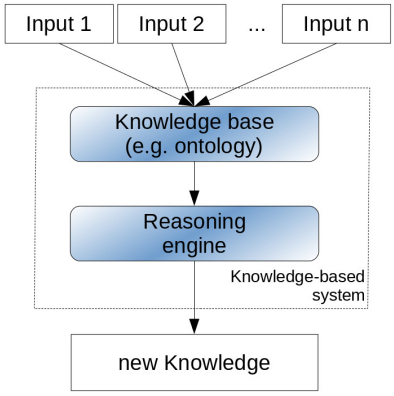

Figure 7. A knowledge graph system.

According to [53], the minimum set of characteristics to describe knowledge graphs are:

i) mainly describe real world entities and their interrelations, organized in the form of a graph,

ii) define possible classes and relations of entities in a schema,

iii) allow for potentially interrelating arbitrary entities with each other, and

iv) cover various topical domains.

In recommendation systems, knowledge graphs are used to represent heterogeneous information, incorporating several information source types. To produce a recommendation, a calculation method over the nodes of the graph is used. An example of a calculation method is the spreading activation based technique [25]. The produced rating is used as the recommendation result. In general, existing knowledge graph recommendation methods can be classified into two categories:

i) embedding-based methods, which initially pre-process the knowledge graph with the use of Knowledge Graph Embedding algorithms (KGE) [77] and then incorporate the learned entity embeddings into a recommendation framework;

ii) path-based methods, which explore various patterns of connections among items in knowledge graphs in order to provide additional guidance for recommendations [83]. Path-based methods make use of knowledge graphs in a more natural and intuitive way, but they rely heavily on manually designed meta-paths. Meta-paths are defined in the scope of information network schemas, and describe how two entity types could be connected via different path types [69]. This method is hard to optimize in practice.

Knowledge graphs are also used in Hybrid Recommendation Systems with several applications e.g. on describing musical and sound items [46]. They offer maximum advantage when the dataset is sparse and gradually becomes redundant, as more training data becomes available. Therefore, they are more useful in cold-start settings [86].

6. Topic Modeling Content-based Systems

Topic modeling explores the idea that the concept of a set of terms can be represented as a weighted distribution over a set of topics. Each topic is a linear combination of terms, where each term is assigned a weight reflecting its relevance to that topic. The primary objective of probabilistic topic models in RS is to capture latent topical information from a large collection of discrete data. Then subsequently we can use this information to assign meaningful probabilities to unobserved information, based on the hidden layer of these topics [47]. Formally, a topic is a probability distribution over terms in a vocabulary. Informally, a topic represents an underlying semantic theme; for example, a document consisting of a large number of words may be modeled as a document originating from a number of topics, smaller than the number of words[8].

The core idea of topic-based modeling is that terms that often occur together are likely to represent the same topic. These models are very popular due to their relative simplicity and efficiency; also, their results are often easy to interpret. More recently, several techniques have been proposed that leverage available metadata to build topic models, which are then used for recommendation [8].

A problem of RS that implement direct (i.e., without any pre-processing of the documents) topic modeling arises when the majority of terms appear in a single document. In such a case, the extreme sparsity and volatility of co-occurrence patterns within the data limit the applicability of these recommendation methods [2].

6.1. Matrix Factorization/LDA

Latent Dirichlet Allocation (LDA) [9] is known to be a powerful generative statistical model technique for discovering and exploiting the hidden thematic structure in large archives of text. In LDA, each topic probability distribution is assumed to have a sparse Dirichlet prior (which is a prior probability distribution, consisting of a family of continuous multivariate probability distributions, parameterized by a vector of positive reals) [42]. The principle behind LDA is that documents contain multiple topics and each topic can be viewed as a probability distribution over a fixed vocabulary. More specifically, the goal of LDA is to decompose a conditional term into two different distributions, one based on term probability distribution and one based on document probability distribution. By doing this, each semantic topic z can be represented as a multinominal distribution of terms, and each document d can be represented as a multinominal distribution of semantic topics [9].

The richer structure in the latent topic space allows the interpretation of docu-

ments in low-dimensional representations [56]. Since in Matrix factorization models the user-item ratings matrix is a product of two lower-rank matrices, the user and the item ones, LDA can also be considered as a matrix factorization approach. There, the document over term probability distribution can be split into two different distributions: the topic over term distribution, and the document over topic distribution [7], as follows:

$$document \times term = (document \times topic)(topic \times term) \tag{5}$$

In recommendation systems, LDA is used to find the latent relations between keywords in item description and item tags created by users, so that the items can be recommended based on their tags [80]. LDA can be used in recommendation in conjunction with other algorithms like sentiment analysis.

6.2. Matrix Factorization/LSA

Latent Semantic Analysis (LSA) in recommendation systems is a technique of analyzing relationships between a set of documents and the terms they contain, by producing a set of concepts related to the documents and terms. Based on these the recommendation system recommends a new document. LSA uses dimension reduction to establish term correlations, allowing the user to compare documents and queries for similar topics [51].

LSA provides a simple and efficient procedure for extracting a topic representation from the associations between terms of a term-document co-occurrence matrix. This representation makes LSA one of the most prominent methods for extracting a spatial representation for words from a multidocument corpus of text, but it also makes it difficult for LSA to deal with the polysemous terms, because each occurrence of a word is treated as having the same meaning, due to the fact that each word is represented as a single point in space.

More specifically, LSA works as follows. It takes a word-document co-occurrence matrix as input and then provides a representation of the essences of the document as a distribution of topics. This can be viewed as a form of "dimensionality reduction", since LSA attempts to find a lower dimensional representation of the structure expressed in a collection of documents [26]. A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called Singular Value Decomposition (SVD) is used to reduce the number of rows while preserving the similarity structure among columns [21]. Using generative techniques the recommended documents can be obtained. LSA can also be used in recommendation, in conjunction with other algorithms like sentiment analysis.

7. Evaluating Content-based RS

In this Section, we provide an overview of the strengths and weaknesses of the Content-based RS. We also present a comparison of the various categories in our taxonomy based on the strengths and weaknesses of the discussed methods.

7.1. Overview of strengths and weaknesses of the Content-based RS

In this subsection, we provide an overview of the strengths and weaknesses of the Content-based RS based on the studied bibliography.

One of the greatest advantage of Content-based RS is that they rely on the content of each item. This provides to these RSs a number of advantages.

- **User Independence:** Content-based RS exploit solely information provided by the active user to build his/her own profile and not information from other users. For this reason Content-based RS make recommendations that match the unique taste of every user. This characteristic of Content-based RS is a great advantage, especially in the case of vast sets of items [89].
- **Transparency:** In Content-based RS, we can easily see how the recommender system works, by been provided by explicitly listing content features or descriptions that caused an item to occur in the list of recommendations. Those features are indicators to consult in order to decide whether to trust a recommendation[52].
- **No vulnerability to Cold Start:** The cold start problem occurs when a new user or item just enters the system. There are three kinds of cold start problems: The new user problem, the new item problem and the new system problem. In Content-based CF methods, in the case of a new item, recommendations can be provided, as these do not depend on any previous rating information of other users [64]. In other terms, Content-based RS does not suffer from the Cold Start problem

But this reason also produces some drawbacks/weaknesses.

- **Sparsity:** Vulnerability to data sparsity is one of the major problems encountered by RS, where data sparsity, when encountered, has a great influence on the quality of the provided recommendations. The main reason behind data sparsity is that most users only rate a small subset of the items, thus the available ratings are usually sparse [1].
- **Over Specialization/Filter-bubble:** The over specialization problem prevents users from discovering new items and other available options. This means that users are restricted to recommendations that resemble those already available to them or directly related to their profiles[31, 1, 33].

- **Synonymy:** The problem of synonymy arises when an item is represented with two or more names or entries with similar meanings. In these cases, the Content Based recommender system cannot identify whether the terms represent different items or the same item, therefore, the recommendation does not consider the latent association between them [33].

7.2. Comparison of of the Content-based RS categories

In this subsection, we provide an overview/comparison of the strengths and weaknesses of the various categories in our Taxonomy. The comparison is based on the studied bibliography, with regards to the most important open issues in the field today. We would like to point out that this comparison is based on conclusions presented in the studied literature. We conducted no quantitative or qualitative comparison tests ourselves, neither is this comparison based on any formal comparison method.

Table 1. An overview of Content-based RS categories sensitivity, with respect to four different known Content Based Recommender System drawbacks, where "+" indicates good performance, "v" indicates average performance, "-" indicates poor performance, and "?" indicates unknown performance.

RS Category	**Cold Start**	**Sparsity**	**Filter Bubble**	**Synonymy**
Bi-polar sentiment analysis	-	-	-	?
Multidimensional sentiment analysis	-	-	-	?
TF-IDF	+	-	-	?
Abstract semantic graph	+	-	-	?
LDA	+	-	-	?
LSA	+	-	-	?
Conversational Recommender Systems	+	-	-	?
(Social) tagging systems	+	v	-	?
Knowledge graphs	+	+	-	+

From the contents of Table 1, we draw the conclusion that the cold start problem has been tackled to an important degree as far as Content-based systems are concerned. Sparsity also remains an important issue, despite recent efforts in reducing its effects. In topic modeling techniques (like LDA/LSA) there exists a sparse and volatile co-occurrence of patterns [47]. Filter Bubble continues to be a very important issue, for which some secondary technics have been developed for addressing it [31, 1, 33]. These technics are using genetic algorithms, in which users can be provided with a set of diverse recommendations and a wide range of alternatives [64]. In other certain cases, items that are too similar to something the user has seen before, are filtered out like in Daily-Learner RS [14] Finally, as far as the vulnerability to the issue of Synonymy on most Content-based approaches, we were unable to draw

from the literature solid conclusions, except Knowledge graphs that include detailed ontologies which have very good performance.

The comparison illustrated in Table 1 is based on the research performed by [45], [31], [63], [1], [47], [62], [78], [72], [33], [65], [37], [40], [39], [14], and [3].

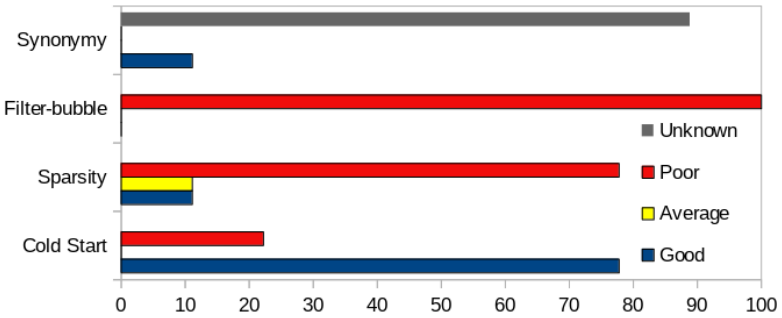

Figure 8. The percentage of Content-based RS categories that perform good, average, poor and with unknown performance over the Cold Start and Sparsity open issues studied in this research.

Figure 8 depicts the percentage of RS sub-categories that perform Good, Average, Poor and Unknown performance over the Cold Start and Sparsity open issues studied in this research based on the results of Table 1. According to this figure it holds that Cold Start is well faced (with good performance) by about 30% of the RS categories. The figure clearly identifies the Diversity (Filter Bubble) problem as the most prevalent along with the Sparsity problem.

8. Conclusions

In this work, we presented a review of the recommendation approaches proposed in the entire research area of Content-based RS, in order to let the reader acquire a quick and complete understanding of this research area. More specifically, to facilitate understanding, we provided a RS taxonomy, i.e., a categorization of each approach based on the tools and techniques employed. Hybrid systems, which appear to be the current trend in recent research, were assigned to a single category depending on their core mechanism, and not to categories used complementary to the main approach. For each such category, we presented an overview of its most representative systems, while avoiding presenting most of their implementation details. We remark that this study either (briefly) presented or included references to i) well-known papers, as well as, ii) the most representative of the recently published papers in each category, according to our knowledge.

Finally, we provided a comparison of recommender system categories according to their ability to efficiently handle some of the well-known RS difficulties. We hope that the present work will be useful to the RS research community, given that RS

research is a multi-disciplinary field based on diverse techniques from various fields of Information Science. Thus, a careful organization of the available approaches is essential in order to obtain an overall view of the field.

Acknowledgements

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T2EDK-03135).

References

[1] Adomavicius G. and Tuzhilin A. 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, June 2005.

[2] Agarwal D. and Chen B.-C. flda: matrix factorization through latent dirichlet allocation. In *WSDM*, pages 91–100. ACM, 2010.

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

[4] Aggarwal C. C. et al. *Recommender systems*. Springer, 2016.

[5] Bauman K., Liu B., and Tuzhilin A. Estimating customer reviews in recommender systems using sentiment analysis methods. In *Conference on Information Systems and Technology (CIST)*, 2015.

[6] Begelman G., Keller P., and Smadja F. Automated tag clustering: Improving search and exploration in the tag space. *Proceedings of the 15th International Conference on World Wide Web*, 2006.

[7] Bergamaschi S., Po L., and Sorrentino S. Comparing topic models for a movie recommendation system. In *WEBIST (2)*, pages 172–183. SciTePress, 2014.

[8] Blei D. and McAuliffe J. Supervised topic models. In *Advances in Neural Information Processing Systems*, 2007.

[9] Blei D. M., Ng A. Y., Jordan M. I., and Lafferty J. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[10] Brin S. and Page L. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*, pages 1–7, 1998.

[11] C. A. Thompson M. H. G. and Langley P. A personalized system for conversational recommendations. *Proceedings of the Journal of Artificial Intelligence Research 21.1*, pages 393–428, 2004.

[12] Catherine R. and Cohen W. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 325–332, New York, NY, USA, 2016. ACM.

[13] Christopher D. Manning H. S., Prabhakar Raghavan. Scoring, term weighting and the vector space model. *Introduction to Information Retrieval, Cambridge University Press*, 2008.

[14] Daniel B. and J. P. M. User modeling for adaptive news access. *User Modelling and User-Adapted Interaction*, 10:147 – 180, 2000.

[15] de Gemmis M., Lops P., Semeraro G., and Basile P. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 163–170, New York, NY, USA, 2008. ACM.

[16] Deerwester S., Dumais S., Furnas G., Landauer T., and Harshman R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407, 1990.

[17] Ehrlinger L. and Wöß W. Towards a definition of knowledge graphs. In *SEMANTiCS (Posters, Demos, SuCCESS)*, 2016.

[18] Eirinaki M., Gao J., Varlamis I., and Tserpes K. Recommender systems for large-scale social networks: A review of challenges and solutions. *Future Generation Comp. Syst.*, 78:413–418, 2018.

[19] Elahi M., Ricci F., and Rubens N. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

[20] F. Fouss A. Pirotte J. R. and Saerens M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering, 19(3)*, pages 355–369, 2007.

[21] Farinella T., Bergamaschi S., and Po L. A non-intrusive movie recommendation system. In *On the Move to Meaningful Internet Systems: OTM 2012*, pages 736–751, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[22] Farrugia J. Model-theoretic semantics for the web. In Hencsey G., White B., Chen Y. R., Kovács L., and Lawrence S., editors, *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 29–38. ACM, 2003.

[23] Garner R. An abstract view on syntax with sharing. *Journal of Logic and Computation*, 22(6):1427–1452, 09 2011.

[24] Gori M. and Pucci A. Itemrank: a random-walk based scoring algorithm for recommender engines. *IJCAI Conference*, pages 2766–2771, 2007.

[25] Grad-Gyenge L., Filzmoser P., and Werthner H. Recommendations on a knowledge graph. In *1st International Workshop on Machine Learning Methods for Recommender Systems*, 05 2015.

[26] Griffiths T. L., Steyvers M., and Tenenbaum J. Topics in semantic representation. *Psychological Review*, in press, 2007.

[27] He Y. A bayesian modeling approach to multi-dimensional sentiment distributions prediction. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 1:1–1:8, New York, NY, USA, 2012. ACM.

[28] Horrocks I. and Patel-Schneider P. F. Three theses of representation in the semantic web. In Hencsey G., White B., Chen Y. R., Kovács L., and Lawrence S., editors, *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 39–47. ACM, 2003.

[29] Hotho A., Jäschke R., Schmitz C., and Stumme G. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, pages 411–426, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[30] Huang Z., Chung W., Ong T.-H., and Chen H. A graph-based recommender system for digital library. *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65–73, 2002.

[31] Iaquinta L., d. Gemmis M., Lops P., Semeraro G., Filannino M., and Molino P. Introducing serendipity in a content-based recommender system. In *2008 Eighth International Conference on Hybrid Intelligent Systems*, pages 168–173, Sep. 2008.

[32] Illig J., Hotho A., Jaschke R., , and Stumme G. A comparison of content-based tag recommendations in folksonomy systems. *Knowledge Processing and Data Analysis*, pages 136–149, 2011.

[33] Isinkaye F., Folajimi Y., and Ojokoh B. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261 – 273, 2015.

[34] Joachims T. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, US, Nashville, US*, pages 143–151, 1997.

[35] Jones K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[36] Julien Subercaze F. L., Christophe Gravier. Hashgraph an expressive and scalable twitter users profile for recommendation. *EEE/WIC/ACM International Conference on Web Intelligence (WI'13)*, pages 101–108, 2013.

[37] Khusro S., Ali Z., and Ullah I. *Recommender Systems: Issues, Challenges, and Research Opportunities*, pages 1179–1189. Springer Singapore, Singapore, 2016.

[38] Kleinberg J. Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46(5)*, pages 604–632, 1999.

[39] Kumar B. and Sharma N. Approaches, issues and challenges in recommender systems: A systematic review. *Indian Journal of Science and Technology*, 9(47), 2016.

[40] Lau A., Tsui E., and Lee W. An ontology-based similarity measurement for problem-based case reasoning. *Expert Systems with Applications*, 36(3, Part 2):6574 – 6579, 2009.

[41] Leung C. Sentiment analysis of product reviews. In *Encyclopedia of Data Warehousing and Mining, Second Edition (4 Volumes)*, pages 1794–1799. IGI Global, 2009.

[42] Lin J. On the dirichlet distribution by jiayu lin. *Submitted to the Department of Mathematics and Statistics of Queen's University Kingston, Ontario, Canada in conformity with the requirements for the degree of Master of Science*, 2016.

[43] Marinho L. B., Nanopoulos A., Schmidt-Thieme L., Jäschke R., Hotho A., Stumme G., and Symeonidis P. *Chapter 19, Social Tagging Recommender Systems*, chapter Folksonomies as Hypergraphs, pages 615–644. Springer US, Boston, MA, 2011.

[44] Mladenic D. Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems and their Applications*, 14(4):44–54, July 1999.

[45] Nguyen T. T., Hui P.-M., Harper F. M., Terveen L., and Konstan J. A. Exploring the filter bubble: The effect of using recommender systems on content diversity. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 677–686, New York, NY, USA, 2014. ACM.

[46] Oramas S., Ostuni V. C., Noia T. D., Serra X., and Sciascio E. D. Sound and music recommendation with knowledge graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–21, 2016.

[47] Ovsjanikov M. and Chen Y. Topic modeling for personalized recommendation of volatile items. In Balcazar J. L., Bonchi F., Gionis A., and Sebag M., editors, *ECML/PKDD (2)*, volume 6322 of *Lecture Notes in Computer Science*, pages 483–498. Springer, 2010.

[48] Panagiotakis C., Papadakis H., Papagrigoriou A., and Fragopoulou P. Improving recommender systems via a dual training error based correction approach. *Expert Systems with Applications*, page 115386, 2021.

[49] Pang B., Lee L., and Vaithyanathan S. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 79–86, USA, 2002. Association for Computational Linguistics.

[50] Papadakis H., Panagiotakis C., and Fragopoulou P. Scor: A synthetic coordinate based recommender system. *Expert Systems with Applications*, 79:8–19, 2017.

[51] Park L. A. F. and Ramamohanarao K. An analysis of latent semantic term self-correlation. *ACM Trans. Inf. Syst.*, 27, 2009.

[52] Pasquale Lops M. d. G. and Semeraro G. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–106. Springer-Verlag, Berlin, Heidelberg, 2010.

[53] Paulheim H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508, 12 2016.

[54] Popescu A.-M. and Etzioni O. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[55] R. M. and P. T. TextRank: Bringing order into texts. *Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[56] Rao V., V R. K., and Padmanabhan V. Divide and transfer: Understanding latent factors for recommendation tasks. In *RecSysKTL*, volume 1887 of *CEUR Workshop Proceedings*, pages 1–8. CEUR-WS.org, 2017.

[57] Rendle S., Balby Marinho L., Nanopoulos A., and Schmidt-Thieme L. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 727–736, New York, NY, USA, 2009. ACM.

[58] R.Manjula A. C. Content based filtering techniques in recommendation system using user preferences. *International Journal of Innovations in Engineering and Technology*, 7, 2016.

[59] Robertson S. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60, 2004.

[60] Salton G., Wong A., and Yang C. A vector space model for automatic indexing. *Communications of the ACM, vol. 18, no. 11*, pages 613–620, 1975.

[61] Salton G. Y. C. S. On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372, 1973.

[62] Schein A. I., Popescul A., Ungar L. H., and Pennock D. M. Methods and metrics for cold-start recommendations. In Järvelin K., Beaulieu M., Baeza-Yates R. A., and Myaeng S., editors, *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*, pages 253–260. ACM, 2002.

[63] Shahabi C. and Chen Y.-S. Web information personalization: Challenges and approaches. In Bianchi-Berthouze N., editor, *Databases in Networked Information Systems*, pages 5–15, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[64] Sharma L. and Gera A. A survey of recommendation system research challenges. In *International Journal of Engineering Trends and Technology (IJETT)*, 2013.

[65] Sharma L. and Gera A. A survey of recommendation system: Research challenges. In *International Journal of Engineering Trends and Technology (IJETT)*, 2013.

[66] Shepitsen A., Gemmell J., Mobasher B., and Burke R. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 259–266, New York, NY, USA, 2008. ACM.

[67] Shi Y., Larson M., and Hanjalic A. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, May 2014.

[68] Shimazu H. Expertclerk: Navigating shoppers' buying process with the combination of asking and proposing. *International Joint Conferences on Artificial Intelligence*, pages 1443–1448, 2001.

[69] Sun Y., Han J., Yan X., Yu P. S., and Wu T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *In VLDB' 11*, 2011.

[70] Symeonidis P., Nanopoulos A., and Manolopoulos Y. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 43–50, New York, NY, USA, 2008. ACM.

[71] Terán L., Mensah A. O., and Estorelli A. A literature review for recommender systems techniques used in microblogs. *Expert Systems with Applications*, 103:63–73, 2018.

[72] Thivakaran T. and Nedunchelian R. Recommendation system for the long tail problem using factorization through latent dirichlet allocation. In *Middle-East Journal of Scientific Research 23*, 2015.

[73] Tso-Sutter K. H. L., Marinho L. B., and Schmidt-Thieme L. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1995–1999, New York, NY, USA, 2008. ACM.

[74] Turney P. D. and Pantel P. From frequency to meaning: vector space models of semantics. *Journal of artificial intelligence research, vol. 37, no. 1*, pages 141–188, 2010.

[75] Vanderwal T. Off the top: Folksonomy entries. *http://www.vanderwal.net*, 2007.

[76] Vig J., Sen S., and Riedl J. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI 2009, Sanibel Island, Florida, USA, February 8-11, 2009*, pages 47–56, 2009.

[77] Wang Q., Mao Z., Wang B., and Guo L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29:2724–2743, 2017.

[78] Wang X., Wang D., Xu C., He X., Cao Y., and Chua T.-S. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, 2018.

[79] Warnestal P. User evaluation of a conversational recommender system. In *Proceedings of the 4th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2005.

[80] Xie W., Dong Q., , and Gao H. A probabilistic recommendation method inspired by latent dirichlet allocation model. *Mathematical Problems in Engineering*, 2014, 2014.

[81] Xie W., Ouyang Y., Ouyang J., Rong W., and Xiong Z. User occupation aware conditional restricted boltzmann machine based recommendation. In *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on*, pages 454–461. IEEE, 2016.

[82] Yamamoto Y., Kumamoto T., and Nadamoto A. Role of emoticons for multidimensional sentiment analysis of twitter. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '14, pages 107–115, New York, NY, USA, 2014. ACM.

[83] Yu X., Ren X., Sun Y., Gu Q., Sturt B., Khandelwal U., Norick B., and Han J. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 283–292, New York, NY, USA, 2014. ACM.

[84] Z. S. Syed T. F. and Joshi A. Wikipedia as an ontology for describing documents. *Proceedings of the Second International Conference on Weblogs and Social Media, AAAI Press*, 2008.

[85] Zanardi V. and Capra L. Social ranking: Uncovering relevant content using tag-based recommender systems. In *Proceedings of the 2008 ACM Conference*

on Recommender Systems, RecSys '08, pages 51–58, New York, NY, USA, 2008. ACM.

[86] Zhang F., Yuan N. J., Lian D., Xie X., and Ma W.-Y. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 353–362, New York, NY, USA, 2016. ACM.

[87] Zhang Y., Liu R., and Li A. A novel approach to recommender system based on aspect-level sentiment analysis. *4th National Conference on Electrical, Electronics and Computer Engineering*, 2016.

[88] Ziani A., Azizi N., Schwab D., Aldwairi M., and Chekkai N. Recommender system through sentiment analysis. *2nd International Conference on Automatic Control, Telecommunications and Signals*, 2017.

[89] Zisopoulos C., Karagiannidis S., Demirtsoglou G., and Antaris S. Content-based recommendation systems. 11 2008.

Received 13.07.2022, Accepted 12.10.2022