Semantics and Content-Based Recommendations

Cataldo Musto, Marco de Gemmis, Pasquale Lops, Fedelucio Narducci, and Giovanni Semeraro

1 Introduction

Content-based recommender systems (CBRSs) basically rely on descriptive features to build a representation of items and users which is used to generate personalized recommendations. Such recommendations may regard *both* items provided with a textual description (e.g., the plot of a movie) as well as items that are themselves 'textual' (e.g., a news article).

Typically, content-based recommendations are obtained by matching up the attributes of the target user profile, in which preferences and interests are stored, with the attributes of the items. The result is a relevance score that represents the target users' level of interest in those items. In other terms, CBRSs are based on the assumption that user preferences remain stable over time (even when such preferences are constructed during the interaction with the system, as it happens in *conversational approaches*), since they suggest items similar to those a target user already liked in the past.

However, early CBRS models were based on keyword-based approaches exploiting simple term-counting. Accordingly, early models were not able to obtain a complete comprehension of the textual content describing the items nor to encode semantic relationships between terms. In particular, early CBRS show clear limits due to properties of natural language elements, such as:

C. Musto (✉) · M. de Gemmis · P. Lops · G. Semeraro
Università degli Studi di Bari, Bari, Italy
e-mail: cataldo.musto@uniba.it; marco.degemmis@uniba.it; pasquale.lops@uniba.it; giovanni.semeraro@uniba.it

F. Narducci
Politecnico di Bari, Bari, Italy
e-mail: fedelucio.narducci@poliba.it

F. Ricci et al. (eds.), *Recommender Systems Handbook*,
https://doi.org/10.1007/978-1-0716-2197-4_7

- POLYSEMY, multiple meanings for one word;
- SYNONYMY, multiple words with the same meaning;
- MULTI-WORD EXPRESSIONS, a sequence of two or more words whose properties are not predictable from those of the individual words;
- ENTITY IDENTIFICATION OR NAMED ENTITY RECOGNITION, the difficulty to locate and classify elements mentioned in text into predefined categories;
- ENTITY LINKING OR NAMED ENTITY DISAMBIGUATION, the difficulty of determining the identity (often called the *reference*) of entities mentioned in text.

This is a sharp limitation, since a keyword-based syntactic representation is often not enough to correctly catch the preferences of the users, as well as the informative content conveyed by the items. Of course, a sub-optimal comprehension of the informative content leads to a sub-optimal representation of the items and, in turn, to recommendations which are not accurate. As shown in several literature [86], it is necessary to improve such a representation in order to fully exploit the potential of content-based features and textual data.

With no doubt, we can state that *semantics* represents the theoretical foundation to proceed in this direction, by implementing more advanced models that allow machines to better understand information provided in natural language . In this research line, *semantics-aware recommender systems* represent one of the most innovative lines of research in the area of recommender systems. Indeed, as we stated in our previous contribution to the handbook [37], thanks to these representations it is possible to give meaning to information expressed in natural language and to obtain a deeper comprehension of the information conveyed by textual content.

The goal of this chapter is to integrate and extend the content previously presented in the chapter *"Semantics-aware Content-based Recommender Systems"* [37]. In particular, the goal of this chapter is to focus on *novel* research directions in the area.

The chapter is organized as follows: first, we start the discussion with an historical perspective in the area, then we provide an overview of the main techniques to incorporate semantics into items and user profiles. These approaches can be broadly split in two categories: *exogenous* and *endogenous* approaches. The former relies on the integration of external knowledge sources, such as ontologies, encyclopedic knowledge and data from the Linked Data cloud, while the latter relies on a lightweight semantic representation based on the hypothesis that the meaning of a word depends on its usage in large corpora of textual documents [57, 70].

Both the approaches can be exploited to cope with the issues of keyword-based syntactic representation: as an example, *word sense disambiguation techniques based on linguistic resources*, such as WordNet, can tackle polysemy, synonymy and multi-word expressions. Similarly, techniques to link items to knowledge graphs, such as those based on the exploitation of *ontologies* and *Linked Open Data*, can be helpful to deal with entity identification and entity linking. The same principle holds for *endogenous* representations, since the lightweight semantics learnt based on distributional semantics models can effectively tackle ambiguity issues.

Further details on the methods are provided in Sect. 2. However, this chapter only sketches these techniques since they represent the focus of the previous edition of our work [37]. Indeed, as previously said, a significant part of the chapter is devoted to recent trends in the area of content-based recommendations. Such trends regard: (1) techniques investigating new methods for *representing* content-based features; (2) techniques investigating new *sources* to gather content-based features; (3) new *use cases* for content-based features, such the exploitation of *content* to generate explainable recommendation and to build conversational recommender systems .

2 Content-Based Recommender Systems

This section reports an overview of the basic principles for building CBRSs and describes the evolution of the techniques adopted for representing items and user profiles.

2.1 The Architecture of a Content-Based Recommender System

The high level architecture of a content-based recommender system is depicted in Fig. 1. The recommendation process is performed in three steps, each of which is handled by a separate component:

- CONTENT ANALYZER—The main responsibility of this component is to represent the content of items (e.g., documents, product descriptions, etc.) coming from information sources in a form suitable for the next processing steps. It extracts features (keywords, n-grams, concepts, ...) from item descriptions and produces a structured item representation stored in the repository *Represented Items*. Early CBRSs adopt relatively simple retrieval models, such as the Vector Space Model (VSM), a spatial representation of text documents [166]. In that model, each document is represented by a vector in a multidimensional space, where each dimension corresponds to a term from the overall vocabulary of a given document collection [9, 165]. As we will show in the next sections, this model is often replaced by more recent and more effective *embedding* techniques. This representation is the input to the PROFILE LEARNER and FILTERING COMPONENT. For more details on NLP techniques for recommender systems we also suggest to refer to Chap. "Natural Language Processing for Recommender Systems";
- PROFILE LEARNER—This module collects data representative of the user preferences and builds the user profile, a model that generalizes the observed data. Preferences on items are collected as ratings on discrete scale and stored in a repository (*Feedback*). Usually, the generalization strategy is realized through

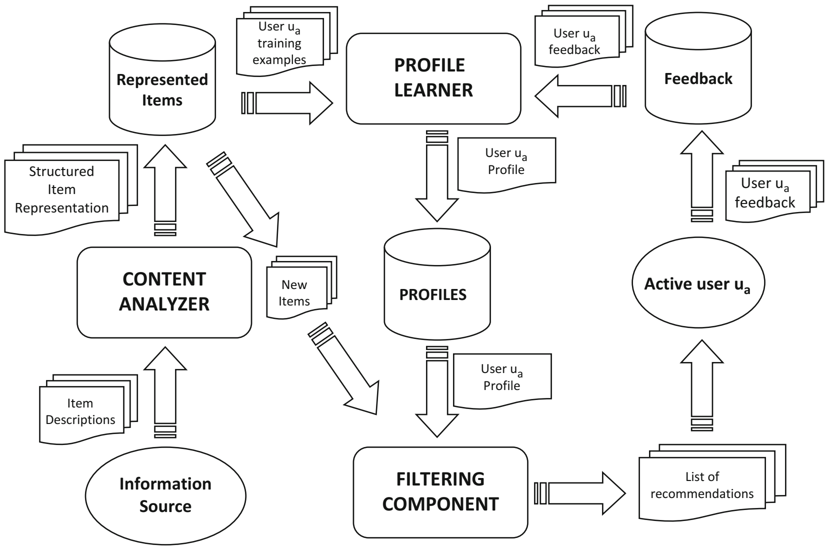

Fig. 1 High level architecture of a content-based recommender

supervised machine learning algorithms [121], which infer the *user profile* from items and corresponding ratings.

- FILTERING COMPONENT—This module predicts whether a new item is likely to be of interest for the *active user*, the one for whom recommendations are to be calculated. It matches the features in the user profile against those in the item representations and produces a binary or continuous relevance judgment, the latter case resulting in a ranked list of potentially interesting items [77]. Ratings can be gathered on generated recommendations, then the learning process is performed again on the new training set, and the resulting profile is adapted to the updated user interests. The iteration of the feedback-learning cycle over time enables the system to take into account the dynamic nature of user preferences.

2.2 Semantics-Aware Content Representation

The Vector Space Model [166] can be useful to develop very simple intelligent information systems, but in order to cope with the above mentioned issues inherently related to natural language and its ambiguity, semantic techniques are crucial to shift from a *keyword-based* to a *concept-based* representation of items and user profiles.

Figure 2 depicts a classification of semantic techniques.

Endogenous approaches exploit large corpora of documents to infer the usage of a word, i.e. its *implicit* semantics. The main ideas behind these methods is described in Sect. 2.2.1.

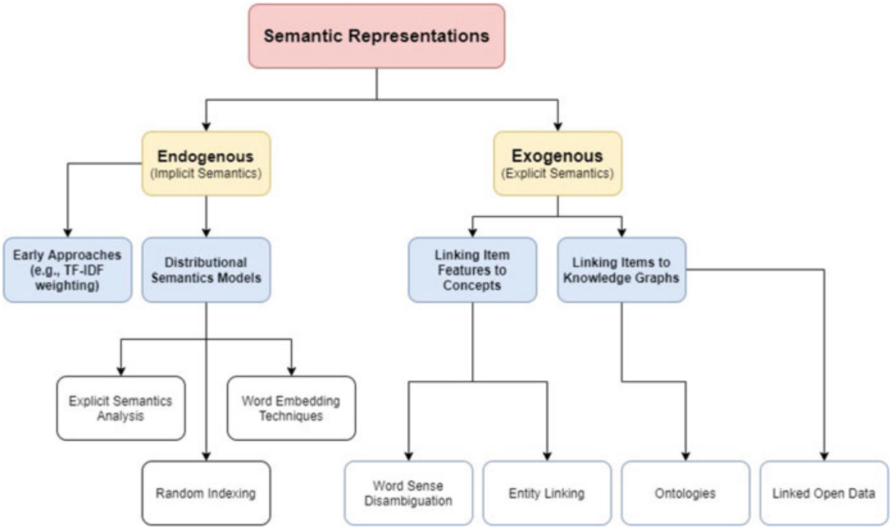

Fig. 2 Classification of semantic representation techniques

Exogenous approaches rely on *external* knowledge sources, such as machine readable dictionaries, taxonomies, thesauri or ontologies, for representing items and user profiles. Section 2.2.2 focuses on the description of these techniques for an *explicit* representation of the semantics.

As previously said, this section provides just a brief overview of these techniques; we suggest to refer to the previous edition of this chapter for a deeper analysis [37]. A complete overview of semantics-aware representation strategies is discussed in [105].

2.2.1 Endogenous Semantics

Techniques for *endogenous semantics representation* fall into the general class of Distributional Semantics Models (DSMs), that were originally introduced in computational linguistics and cognitive sciences [169].

These data-driven approaches rely on the so-called *distributional hypothesis*, which states that *"Words that occur in the same contexts tend to have similar meanings"* [70], thus the algorithms that follow these approaches extract information about the meaning of a word by analyzing its usage in large corpora of textual documents.

DSMs are based on Wittgenstein's idea that the meaning of a word is its use in the language [204], and that semantically similar words also share similar contexts of usage [162].

	c1	c2	c3	c4	c5	c6	c7	c8	c9
beer		✓	✓			✓	✓		
wine		✓	✓			✓	✓	✓	
spoon	✓			✓				✓	✓
glass	✓	✓	✓		✓				✓

Fig. 3 A term-context matrix

These models, also known as *geometrical models*, learn similarities and connections in a totally unsupervised way. Indeed, they represent each term that occurs in a corpus as a vector in a high-dimensional vector space called `WordSpace` [106].

Given a corpus, usually the `WordSpace` is built by means of a *term-context matrix*, as the one presented in Fig. 3. Each row represents one term of the vocabulary (obtained by applying an NLP pipeline), while each column is *a context of usage*. Every time a term is used in a particular context, this information is encoded in the matrix. As an example, according to Fig. 3 the term *beer* is used in contexts c_2, c_3, c_6 and c_7. We can imagine a context as a fragment of text in which the word occurs. Thus, each term is represented by a *vector* (the corresponding row in the matrix), modeled in a vector space whose dimensions are the columns of the term-context matrix.

Given a `WordSpace`, a vector space representation of the documents called `DocSpace` can be also computed. A `DocSpace` can be obtained by following different strategies: as shown by Sahlgren [164], a document representation may be calculated as the centroid vector of the vector space representation of the words that appear in the document, or as their *weighted sum*. Next, given a `WordSpace`, the similarity of two terms can be estimated by analyzing the overlap between their usage. According to the example in Fig. 3, we can state that words *beer* and *wine* are very similar since they share a large number of contexts. In practice, the similarity between words is estimated as the proximity between vectors that represent those words in the `WordSpace`, in accordance with the *similarity-is-proximity* metaphor [164]. Thus, it can be computed in several ways, such as cosine similarity, Manhattan and Euclidean distances, or relative entropy-based measures [122].

We need to further clarify the definition of *context*. Generally speaking, it is *a fragment of text in which a word appears*. In the simplest formulation, the context is the *whole document*. In that case, the *term-context matrix* corresponds to the *term-document matrix* in the classical Vector Space Model [166]. Finer-grained options are possible: the context could be a paragraph, a sentence, a window of surrounding words or even a single word. A survey about different strategies to handle the concept of context is provided in [188].

Among the approaches for *endogenous semantics representation*, it is worth to mention the Explicit Semantic Analysis (ESA). ESA, which became very popular in the early 2010s, builds a semantics-aware representation of words in terms of Wikipedia concepts [61]. In ESA, the representation of the terms is based on the so-called *ESA matrix*, whose rows correspond to the Wikipedia vocabulary (i.e., the set of distinct terms found within Wikipedia articles, after applying basic NLP operations), while columns correspond to Wikipedia pages. Each row represents a term and is called *semantic interpretation vector*. It contains the list of *concepts* (Wikipedia pages) associated to the term, along with the corresponding weights.

The semantics of a document is typically obtained by computing the centroid of the semantic interpretation vectors associated with the individual terms occurring in that document. ESA showed good performance in tasks as text categorization [59], semantic similarity computation [60] and recommendation [143].

One of the main issues with DSMs is represented by the tremendous increase of the size of the term-context matrix as the context gets smaller. *Word Embedding* techniques have been developed to manage that issue. They *project* the original vector space into a smaller *but substantially equivalent* one, thus returning a more compact `WordSpace`. Differently from *pure* DSMs, such as ESA, the new dimensions of the reduced vector space are not human understandable anymore. Popular Word Embedding techniques are Latent Semantic Analysis [95] and Random Indexing [164]. These representations recently got new interest after the introduction of Word2Vec [117] and, as we will show in Sect. 3.1, the use of Word Embedding techniques is today one of the most active research lines in the area of semantics-aware content-based recommender systems.

Word2Vec is a technique that exploits *neural networks* to learn a vector space representation of words. It was first proposed by Tomas Mikolov et al. [117], and it gained a lot of attention in the last years due to the simplicity of the approach and to the effectiveness it obtained in several tasks, including recommendation [129]. In a nutshell, this approach is used to learn (small) word embeddings by exploiting a *two-layer neural network* which is fed with examples gathered from a corpus of textual data to learn the contexts and the linguistic usage of words to generate the embeddings.

A toy example of the neural network exploited by Word2Vec is reported in Fig. 4. Given a corpus of textual data, we define an input layer of size $|V|$, that corresponds to the dimension of the vocabulary of the terms. This means that each term that appears in the corpus is mapped to an element in the input layer. Next, an output layer $|N|$ is created. In this case, N is the size of the embedding we want to obtain at the end of the learning process. The value of N is a parameter of the model and has to be properly tuned. Clearly, the greater the value, the more complex the learning process since a larger number of weights in the network has to be learnt, but the better the resulting representation.[1]

The edges connecting the nodes in the network have different weights. They are initially randomly set and they are updated through the training process. The final

[1] The equation *'larger vectors, better representation'* is typically valid. However, when the dimension becomes too large a decrease in the performance can be noted.

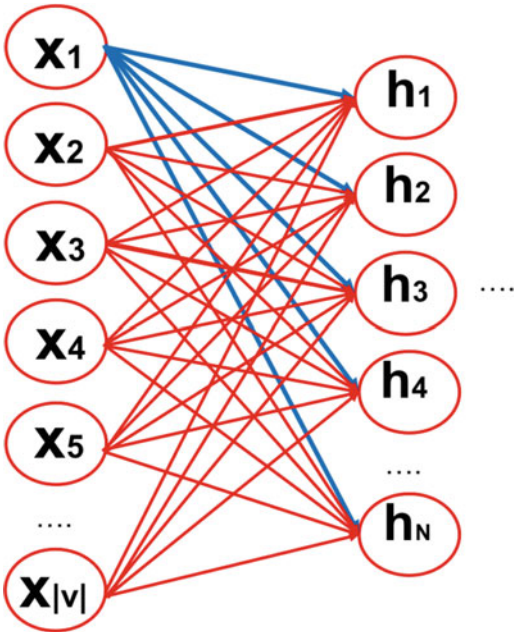

Fig. 4 Structure of the network

representation of a term is the set of weights that connects its corresponding node in the input layer to all the nodes in the output layer. Formally, given a term t_k its representation is given by $[w_{t_k,v_1}, w_{t_k,v_2}, w_{t_k,v_n}]$.

Such a discussion makes immediately emerge the importance of the training process in Word2Vec, since the network needs to acquire input examples to properly learn linguistic regularities and to update the weights in the network (and, in turn, the resulting representation) accordingly.

The training of the network can be carried out by exploiting two different methodologies, that is to say the Skip-Gram methodology (SG) and the Continuous Bag of Words (CBOW). The choice of the most suitable technique is a design choice: according to Mikolov, SG works well when the training set is small and shows a good accuracy even on rare terms, whereas CBOW is several time faster than SG and is more accurate for frequent words.

More details about Word2Vec can be found also in [118], while a thorough discussion about *endogenous methods* can be found in Chapter 3 of the book [105].

2.2.2 Exogenous Semantics

Approaches for *exogenous* semantics representations rely on the linguistic, cultural and background knowledge that is encoded and made available through *external* knowledge bases.

The main difference between *endogenous* and *exogenous* techniques for semantics-aware representation lies in the nature of the knowledge bases they rely on. In the first case, the semantics is obtained by exploiting *unstructured* data (corpora), and is directly inferred from the available information. In the second, the semantics *comes from the outside*, since it is obtained by mining and exploiting data which are previously encoded in *structured* and external knowledge sources.

Most popular structured knowledge sources today available are:

WordNet [54, 119, 120]. It is a lexical database for the English language, made by cognitive scientists, freely available online[2] and extensively used in NLP research [178]. The goal of WordNet is to model *meanings* that can be expressed through the known *word forms*, and to represent the lexical relations that exist among them. The basic building block of WordNet is the SYNSET (SYNonym SET), that encodes a specific *meaning* (concept) though the set of synonym words that can be used to express it.

BabelNet [145]. It is a large-scale multilingual encyclopedic dictionary and semantic network.[3] It integrates heterogeneous resources such as WordNet, Wikipedia, Wikidata [192] (described later), Wiktionary and other lexical databases. In a nutshell, the knowledge encoded in BabelNet is represented through a labeled directed graph. *Nodes* are *concepts* extracted from WordNet and Wikipedia (synsets and Wikipages), while *edges* among nodes encode the *semantic relations* coming from WordNet, as well as semantically unspecified relations from hyperlinked text coming from Wikipedia.

The Linked Open Data (LOD) cloud [17]. This term was introduced to identify the huge number of datasets released through the Linked Open Data initiative, a project started in the late 2000s that inherits some of the concepts and the ideas of the *Semantic Web*. The LOD project is grounded on two cornerstones: (1) each resource available on the Web should be uniquely referred to through an `URI`; (2) data have to be encoded and linked by using `RDF`, acronym of Resource Description Framework. The *nucleus* of the LOD cloud is commonly represented by `DBpedia` [8], the `RDF` mapping of Wikipedia that acts as a *hub* for most of the `RDF` triples made available in the LOD cloud.

Wikidata [192]. It is a free, collaborative and multilingual database, built with the goal of turning Wikipedia into a fully structured resource. While `DBpedia` is almost automatically built by mapping in `RDF` format the information contained in the Wikipedia infoboxes, Wikidata entries are collaboratively created and maintained by both Wikidata editors and automated bots. Due to its collaborative nature Wikidata is continously updated, while `DBpedia` is usually updated only twice a year.

As described in Fig. 2, two strategies can be adopted to exploit the data available in the knowledge sources to build a semantics-aware representation of items:

[2] http://wordnet.princeton.edu.

[3] http://babelnet.org.

(a) linking item features to concepts or (b) linking items to a knowledge graph.

The goal of the first group of techniques is to associate each feature with its *correct semantics* and to identify more complex concepts expressed in the text. Word Sense Disambiguation (WSD) techniques fall in this category because they tackle the problem of correctly identifying which of the senses of an ambiguous word is invoked in a particular use of the word itself [110]. Several WSD algorithms have been developed; for instance, in [170] the authors exploit WordNet to disambiguate item descriptions used to train a content-based recommender system and to build synset-based user profiles. Entity Linking (EL) methods [155] are also classified in the first group. EL is the task of associating the *mention* of an *entity* in a text to an *entity* of the real world stored in a knowledge base [33]. A systematic review of other techniques and algorithms for EL is provided in [172].

Techniques included in the second group directly link items to nodes in a knowledge graph rather than mapping word forms to word meanings or entities. There is no need to process any textual content because either the item is directly linked to the Linked Open Data cloud or an *ontological* representation of the domain of interest is built, so that items are modeled in terms of *classes* and *relations* that exist in the ontology. More details about exogenous methods can be found in Chapter 4 of the book [105], while a deeper discussion on using LOD and Knowledge Graphs for recommender systems will be provided in Sect. 3.2 of this chapter.

2.3 Strengths and Weaknesses of Content-Based Recommendations

The adoption of the content-based recommendation strategies (especially in their semantics-aware forms) has several advantages when compared to the collaborative one:

- USER INDEPENDENCE—Content-based recommenders exploit solely ratings provided by the active user to build her own profile. Instead, collaborative filtering methods need ratings from other users in order to find the "nearest neighbors" of the active user or to build sophisticated machine learning models. In other terms, they are less prone to *data sparsity* issues and can be more effective when a little amount of data is available.
- TRANSPARENCY—Explanations on how the recommender system works can be 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. Conversely, collaborative systems are black boxes since the only explanation for an item recommendation is that unknown users with similar tastes liked that item. More details about recent *explainable* content-based recommendation methods are provided in Sect. 3.4;

- NEW ITEM—Content-based recommenders are capable of recommending items not yet rated by any user. As a consequence, they do not suffer from the first-rater problem, which affects collaborative recommenders which rely solely on users' preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the system would not be able to recommend it.

Nonetheless, content-based systems have several shortcomings:

- LIMITED CONTENT ANALYSIS—Content-based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend. No content-based recommendation system can provide suitable suggestions if no descriptive features of the items is available. Of course, recent advances in the area (e.g., pre-trained language models used to train word embeddings and structured features available in knowledge graphs) have partially mitigated this issue, but the *need for content* is still a mandatory requirement for these approaches;
- OVER-SPECIALIZATION—Content-based recommenders have no inherent methods for finding something unexpected. The system suggests items whose scores are high when matched against the user profile, hence the user is going to be recommended items similar to those already rated. This drawback is also called *serendipity* problem, to highlight the tendency of the content-based systems to produce recommendations with a limited degree of novelty. To give an example, when a user has only rated movies directed by Stanley Kubrick, she will be recommended just that kind of movies. A "perfect" content-based technique would rarely find anything *novel*, limiting the range of applications for which it would be useful.
- NEW USER—As other recommendation paradigms, also content-based recommender systems suffer of *cold start*. Indeed, enough ratings have to be collected before a content-based recommender system can really understand user preferences and provide accurate recommendations. Therefore, when few ratings are available, as for a new user, the system will not be able to provide reliable recommendations.

3 Recent Developments and New Trends

Recent developments in the area of content-based recommender systems can be roughly split into three research directions: first, as for *endogenous* techniques, the most relevant advances concern methods based on embeddings and distributed representations. Next, as for *exogenous* techniques, recent research focused on methods exploiting the information encoded in knowledge graphs . Finally, it is worth mentioning methods based on multimedia features and user-generated content. In the next sections, relevant work recently presented in these areas will be discussed. Moreover, we will also show how content can be used to improve the user experience by means of natural language explanations and conversational interfaces.

3.1 Embeddings and Distributed Representations

As introduced in Sect. 2.2.1, approaches for *endogenous semantics representation* exploit textual content and produce a vector space representation of the items to be recommended as well as of the users.

Nowadays, it is very common to refer to these representation as *embeddings*. This term can be further specialized into *word embeddings* and *sentence embeddings*, depending on whether a representation for each word or for each sentence is built. Due to the effectiveness of these techniques, whose spread has been largely discussed in recent years [69], there are many approaches that exploit embeddings and distributed representations for recommendation tasks. In this section we will provide an overview of relevant work in the area. The section is organized in two parts: First, we introduce early approaches that directly use word and sentence embeddings to feed recommendation models. Next, following the recent trend of deep learning architecture, we show how distributed representations can be used, together with deep neural networks, to generate accurate semantics-aware content-based recommendations.

3.1.1 Recommender Systems Based on Word and Sentence Embeddings

The early attempts in the area put their roots in the area of *pure* distributional semantics models (DSMs). In particular, these attempts exploited a DSM to learn a vector space representation of users and items and they *directly* used this representation in a recommendation model.

As an example, McCarey et al. [114] evaluate the effectiveness of Latent Semantic Indexing (LSI) [38] in a content-based recommendation scenario. Similarly, Musto et al. [124, 125] propose an extension of the classical VSM, called *enhanced Vector Space Model* (eVSM), that exploits Random Indexing to build a dense vector space representation of users and items. As shown in the experiments, eVSM overcame other classical content-based filtering techniques, and the findings were confirmed by subsequent experiments where the same approach is evaluated in a context-aware recommendation scenario [127].

By following these attempts, in [111] the authors exploited word embedding techniques to infer the vector-space representations of venues based on venue descriptions and reviews data. This work confirmed the previously presented outcomes, since the experiments showed that the use of content features and word embeddings significantly enhanced the accuracy of venue recommendations.

Next, a significant boost to the research in the area was noted after the introduction of Word2Vec [117]. This technique, based on the principles of *distri-*

butional semantics models, exploits a two-layer neural network to learn a vectorial representation of users and items. In this resarch line, Ozsoy et al. [147] proposed the use of Word2Vec to learn word embeddings representing items and user profiles. Their experiments showed that CBRSs based on word embeddings can obtain results comparable to those obtained by other content-based approaches and by algorithms for collaborative filtering based on matrix factorization. Next, in [173] the authors employed Word2Vec to compute the vectors of tags in Tumblr and recommended Tumblr blogs to the users. Next, a context-aware recommender method that extracts contextual information from textual reviews using a word embedding based model is proposed in [181] and a similarity measure inspired by Word2Vec, which is then used to learn the similarity between items, is presented in [116].

A comparative analysis among different word embedding techniques in a content-based recommendation scenario is presented in [130]. In particular, the work compares Latent Semantic Indexing, Random Indexing, and Word2Vec to establish the most effective technique. Results of the experiments in a *movie* and *book* recommendation scenarios show the good performance of the Word2Vec strategy, with the interesting outcome that even a smaller word representation could lead to accurate results. Furthermore, it emerged that the effectiveness of word embedding approaches is directly dependent on the sparsity of the data. This is an expected behavior since content-based approaches can better deal with cold-start situations, and with very sparse datasets they perform better than collaborative filtering or matrix factorization baselines. However, as discussed in [23] and [26], it is necessary to point out that the performance of Word2Vec-based models is strictly dependant on hyperparameter tuning. As shown in both these works, the performance of Word2Vec-based approach are strictly dependant of the optimization of the parameters, so it is fundamental to devote the necessary attention to this step.

Due to the effectiveness shown by Word2Vec, several research exploited Word2Vec to encode sequences of actions or sequences of events (instead of sequences of words) to learn a vector space representation of the items. Even if these approaches do not exploit *content-based features*, we deemed as relevant to briefly discuss them in order to provide a complete overview of the effectiveness and of the flexibility of word embedding approaches. The first work that exploited this analogy is Item2Vec [12], where items are used as words and baskets are used as sentences. Similarly, Grbovic at al. [65] adapted Word2Vec to generate product recommendations (i.e., prod2vec). They treated purchase history of a user as the sentence and each product as the word. This approach is further extended is MetaProd2Vec [190], which is based on Prod2Vec and incorporates side information in both the input and output space of the neural network.

Next to Word2Vec, several work propose recommendation methods based on Doc2Vec [96]. Doc2Vec is a neural approach that shares the same principles of Word2Vec and focuses on the representation of *sentences* and *documents*. As an example, in [90] Doc2Vec is used to learn an embedding representing a news article, based on the text and the title of the news. Next, such a representation is used to feed a hybrid recommendation approach. Similarly, in [179] the authors propose an hybrid approach based on the combination of two techniques inspired by Doc2Vec, that it to say, *user2vec*, which uses item descriptions and usage histories to model users and *context2vec* which uses further metadata on items and users in an attempt to incorporate context into the model. In both the cases, the experiments showed that these approaches outperform all the baselines. Good performance of this technique also emerged in [34], where Doc2Vec is used to represent items in a digital library recommendation scenario. Finally, in [3] the authors exploit Paragraph2Vec to generate a vector space representation of reviews. Next, the resulting feature vectors (the neural embeddings) are used in combination with the rating scores in a hybrid probabilistic matrix factorization algorithm. The proposed methodology is then compared to three other similar approaches on six datasets in order to assess its performance. As shown by the results, the exploitation of reviews embeddings led to an improvement of the performance.

3.1.2 Deep Learning Models Based on Word and Sentence Embeddings

Due to the effectiveness shown by recommendation strategies based on pure word and sentence embeddings, several research investigated how to encode distributed representation into more complex recommendation models. One of the first attempts in this research line was due to Kula, who proposed in [94] a hybrid matrix factorisation model representing users and items as linear combinations of their content features' latent factors. As shown by the experiments, the model outperforms both collaborative and content-based models in cold-start or sparse interaction data scenarios.

Next, in parallel with the growth of deep learning architectures and neural models, most of the research effort has been devoted to inject content information into deep architectures. In this area, several work tried to exploit Recurrent Neural Networks (RNNs) [163], which are particularly effective to model sequences of inputs (e.g., audio signals), to encode textual context as well.

As an example, in [180] the authors learn a vector-space representation of textual content based on a Long-Short Term Memory (LSTM) network [79], a specialization of RNNs. The experiments carried out by the authors confirmed again the effectiveness of the model, since the results showed that the use of LSTM networks can further improve the accuracy of the recommendations. This is due to the fact that such architectures are able to learn embeddings that also encode the dependencies between words. These findings have been further investigated in [135], where the authors present a deep content-based recommender system that exploits Bidirectional Recurrent Neural Networks (BRNNs) to learn an effective representation of the items to be recommended based on their textual description. In particular, BRNNs extend RNNs by encoding information about both the preceding and following words, thus leading to a more precise representation. Moreover, the authors further extended such a representation by introducing structured features extracted from the Linked Open Data (LOD) cloud, as the genre of a book, the director of a movie and so on. In the experimental session the effectiveness of the approach is evaluated in a top-N recommendation scenario, and the results showed that the approach obtained very competitive results by overcoming the baselines.

The use of LSTMs to model textual content in CBRS is also investigated in Almahairi et al. [4], who used LSTMs to model textual content (textual *reviews*, in that case) to feed a collaborative recommendation algorithm and by Bansal et al. [10], which applies a bidirectional *Gated Recurrent Unit (GRU)* network to encode item description and an embedding for each tag associated to the item.

Next to the approach based on the exploitation of Recurrent Neural Networks, we can also cite a plenty of research which is based on the use of Convolutional Neural Networks. As an example, in [211], the authors present Deep Cooperative Neural Networks (DeepCoNN), a deep model that consists of two parallel neural networks coupled in the last layers. One of the networks focuses on learning user behaviors exploiting reviews written by the user, while the other one learns item properties from the reviews written for the item. Experimental results demonstrated that DeepCoNN significantly outperforms all the baselines. A similar approach based on the processing of item reviews is also presented in [30] and in [103], where a dual attention mutual learning between ratings and reviews for item recommendation, named DAML, is proposed. In this case, the authors utilize local and mutual attention of the convolutional neural network to jointly learn the features of reviews to enhance the interpretability of the proposed model.

The use of Convolutional Neural Networks is also investigated in [194], where a deep knowledge-aware network (DKN) that incorporates knowledge graph representation for news recommendation is presented, and in [205], where the authors use a CNN network to learn hidden representations of news articles based on their titles.

Generally speaking, all these works gave evidence of the effectiveness of embeddings and distributed representation for recommendation tasks. For the sake of completeness, it is worth mentioning recent attempts in the area of *contextual word representations*. They differ from classic word embeddings since they are able to learn a *context-aware* representation of words which depends on the other words which occur in the sentences. Accordingly, they are able to better handle *ambiguity issues* in content representation. This process is possible because of the use of large pre-trained language models, which can learn highly transferable and task-agnostic properties of a language [51]. The adoption of *contextual word representation* techniques in recommendation tasks is a relatively new direction. As an example, in [71] the authors present a comparative evaluation among several techniques, such as BERT [44], SciBERT [16], ELMo [152], USE [25] and InferSent Sentence Encoders [35]. Experiments show that the sole consideration of semantic information from these encoders does not lead to improved recommendation performance over the traditional BM25 technique [160], while their integration enables the retrieval of a set of relevant papers that may not be retrieved by the BM25 ranking function. Overall, the best results were obtained by USE. Similarly, Stakhiyevich and Huang in [177] proposed an approach for building user profiles to be used in a personalized recommender system, based on Embeddings from Language Models (ELMo) and user reviews. In this case, cosine similarity has been adopted to calculate the similarity between user interests and item categories over their contextual word representations. Finally, Cenikj et al. in [24] investigate, with a preliminary study, the possibility to enhance a graph-based recommender system for Amazon products with two state-of-the-art representation models: BERT and GraphSage. The results obtained by the authors are encouraging in following the idea to merge graph and contextual word representation techniques.

3.2 Linked Open Data and Knowledge Graphs

Linked Open Data provide a great potential to effectively feed filtering algorithms with *exogenous* semantic representations of items. One of the first attempts to leverage Linked Open Data to build recommender systems is *dbrec* [149], a music recommender system based on the *Linked Data Semantic Distance* (LDSD) algorithm [150], which computes the semantic distance between artists referenced in `DBpedia`. Semantic distance can be seen as a way to compute the *relatedness* between two nodes in a knowledge graph (in this case, two artists), and it is obtained as the linear combination of *direct relationship* (i.e., the amount of direct links between the nodes, such as overlapping properties) and *indirect relationship* (i.e., the amount of shared links through other resources). Given a dataset of artists gathered from `DBpedia`, LDSD is computed off-line and it is used to provide users with music recommendations. In particular, semantically related artists, that is to say, artists with a low LDSD score, are suggested. As shown in the experiments reported in [149], recommendations based on LDSD provide competitive results with respect to a music recommender system based on Last.fm, thus giving one of the earliest evidences of the effectiveness of recommendation strategies based on knowledge graphs.

By following this research line, in [126], `DBpedia` is used to enrich the playlists extracted from a Facebook profile with new related artists. Each artist in the original playlist is mapped to a `DBpedia` node, and other similar artists are selected by taking into account shared properties, such as the genre and the musical category of the artist. Another simpler approach to define a CBRS exploiting Linked Open Data is presented in [46]. The ontological information, encoded via specific properties extracted from `DBpedia` and `LinkedMDB` [72], is adopted to perform a *semantic expansion* of the item descriptions, in order to catch implicit relations, not detectable just looking at the nodes directly linked to the item. In that work, the authors used Support Vector Machines to learn user profiles, a technique which tends to be fairly robust with respect to overfitting and can scale up to considerable dimensionalities. The evaluation of different combinations of properties revealed that more properties lead to more accurate recommendations, since this seems to mitigate the limited content analysis issue of CBRSs.

Besides the above mentioned approaches to catch implicit relations which allow to increase the number of common features between items, more sophisticated approaches may be exploited, in order to implement more complex reasoning

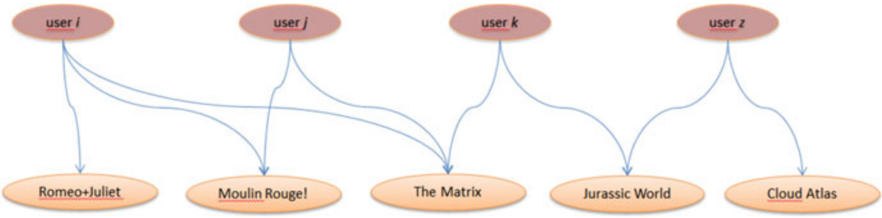

Fig. 5 Basic bipartite graph representing users, items, and their preferences

over the graphs. In particular, graph-based recommender systems model *users* and *items* as *nodes* in a graph, and *edges* connect users with items according to their preferences. An example of such a data model is reported in Fig. 5.

This basic representation is very similar to that usually adopted for collaborative filtering, and recommendations can be produced by assigning to each item $i \in I$ a *relevance score*.

Given such a formulation, the problem of providing a user with a recommendation can be tackled by exploiting an algorithm that assigns a score to an item node in the graph, such as the *PageRank* algorithm [148]. As an example, given a graph-based data model, the PageRank algorithm can be run and PageRank scores can be sorted in descending order. Next, the k nodes with the highest PageRank scores can be returned by the algorithm as recommendations. However, PageRank has the main problem of being *not personalized*, that is to say, the PageRank score of the item nodes (and, accordingly, the recommendations returned by the algorithm) only depends on the topology as well as on the connections that exist in the graph. A well-known variant of the PageRank, called *PageRank with Priors* [73] can be adopted to tackle this issue, since it allows to get a bias towards some nodes, specifically, the preferences of a specific user. As described in [131, 134], this algorithm can be really effective for recommendation tasks since it can adapt her own behavior on the preferences of the target user. In this scenario, PageRank with Priors is executed for each user and the nodes in the graph are ranked according to their PageRank score, as it happens for classical PageRank algorithm. The list of *item* nodes, not yet *voted* by the target user are provided as recommendations. In this setting, Linked Open Data can be used to enrich graphs by introducing *additional* nodes and edges, in order to come up with more effective representations including new connections resulting from the properties encoded in the LOD cloud. Ideally, we can run this enrichment step again and again, in order to introduce in the graph *non-direct* relationships. However, in [128] it has been shown that the introduction of non-direct relationships leads to an exponential growth of the PageRank running time, without a significant improvement in the precision of the recommendation process. Indeed, a simple and straightforward question may emerge from such a scenario: *Is it necessary to inject all the available properties?* and *Are all the properties equally important to provide users with accurate recommendations?* Hence, similarly to what happens in other settings, e.g., machine learning problems, it is necessary to investigate to what extent each property modeled in the graph improves the accuracy of the recommendation strategy, in order to filter out non-useful connections and select only the most meaningful properties. Hence, a possible strategy to automatically identify the most promising LOD-based features is to exploit *feature-selection* methodologies (e.g., Principal Component Analysis, Support Vector Machines, Chi-Squared Test, Page Rank, Information Gain, Information Gain Ratio, Minimum Redundancy Maximum Relevance) adopted in machine learning, whose goal is to improve the prediction performance of the predictors and to provide faster and more cost-effective predictors. For a more detailed discussion on the impact of feature-selection techniques on accuracy and diversity of recommendations we suggest to refer to [131].

Another way to exploit the information stored in a Knowledge Graph (KG) is to extract *topological features* that can be obtained by mining the bipartite and tripartite graph-based data model, as shown in [133]. Such features, encoding some structural characteristics of the data model can be used next to feed a recommendation framework with these new and interesting features.

In [133], an extensive experimental evaluation has been performed to assess the accuracy of different classification algorithms, namely *Naïve Bayes*, *Logistic Regression* and *Random Forests,* trained with item representations based on different groups of features, including the topological ones. One of the main outcomes is that bipartite and tripartite features have performance comparable to that of textual features (simple item descriptions) or LOD-based features (extracted from the LOD cloud). Given that the process that computes textual and LOD-based features requires a quite complex NLP pipeline or a mapping of items to `DBpedia`, topological features represent a more lightweight (they are very few) and therefore a more viable alternative for representing items. On the other side, the benefit of injecting the exogenous knowledge coming from the Linked Open Data cloud particularly emerged when data are sparse.

Recently, several work tried to combine the information encoded in knowledge graphs into machine learning and deep learning models. As an example, in [103] the authors proposed KRED, a knowledge-aware recommender systems exploiting an *enhanced* representation based on KGs. In particular, the authors start from a vector-space representation of the item (a news article, in this case) and then enrich the embedding by aggregating information from their *neighborhood* (extracted from the knowledge graph, of course) of the entities mentioned in the article. In this way, extra information coming from the other nodes directly connected to the target one can be encoded in the representation. This task is carried out by an *information distillation layer*, that aggregates the entity embeddings under the guidance of the original item representation, and transforms the item vector into a new one. Next, in [15] the authors present SemAuto, a recommendation model that puts together knowledge graphs and deep learning techniques. SemAuto is based on *autoencoders*, a type of neural network used to unsupervisedly learn a representation for a set of data. An autoencoder is typically split into two parts, a reduction side, that aims to reduce the noise of original representation, and a reconstructing side, that tries to generate from the reduced encoding a representation as close as possible to its original input. In this setting, Bellini et al. introduce the idea of enhancing autoencoders by means of knowledge graphs. In particular, SemAuto is inspired by fully-connected autoencoders and tries to make the model explainable by labeling neurons in hidden layers with the entities available in the knowledge graphs. In other terms, nodes in the hidden layers are replaced by nodes and connections available in a KG. According to authors' claims, this is supposed to improve both the predictive accuracy as well as the explainability of the model. As shown by the results, SemAuto actually outperforms state-of-the-art recommendation algorithms on both accuracy and diversity.

Moreover, a recent trend concerns the application of Graph Neural Networks (or Graph Convolutional Neural Networks [209]) in the area of RSs. In particular, GCNs aim to generalize convolutional neural networks to non-Euclidean domains (such as graphs). In this area, Wang et al. [196] present Knowledge Graph Convolutional Networks (KGCN) for recommender systems. KGCN learn a representation based on: (1) interactions encoded in the user-item matrix; (2) descriptive encoded in a knowledge graph. Generally speaking, KGCN learn a h-order representation of an entity as a mixture of its initial representations and the representation of its neighbors up to h hops away. In order to reduce the computational load of the method, the authors adopt a sampling method to select just a fixed size of neighbors instead of using its full neighbors. In other terms, KGCN is proposed to capture high-order structural proximity among entities in a knowledge graph. So the final representation of an entity is dependent on itself as well as its immediate neighbors, and also takes into account users' personalized and potential interests. As shown in the experiments, KGCN outperform state-of-the-art baselines in movie, book, and music recommendation. A similar intuition is investigated in [197], where the authors present KGAT (Knowledge Graph Attention Network). In this case, *attention mechanisms* are used to model the high order connectivities in KG in an end-to-end fashion. The core of the approach lies in the definition of an attentive embedding propagation layer, which adaptively propagates the embeddings from a node's neighbors to update the node's representation. For further details on recommender systems based on knowledge graphs we suggest to refer to the survey by Guo et al. [68].

Finally, several work investigated *graph embedding* techniques to improve the quality of recommendation algorithms based on knowledge graphs. In particular, graph embedding techniques take a graph as input and produce a vector-space representation of the nodes encoded in the graph. Intuitively, these models merge the flexibility and the effectiveness of vector-space representation with the richness of a graph-based representation. Such an intuition is investigated in [136], where two popular graph embedding techniques, that is to say, *node2vec* [66] and *Laplacian Eigenmaps* [14], are compared in a recommendation task based on a classification framework. In other terms, the vectors returned by the graph embedding techniques are used as positive and negative examples to build a classification model for each user. As shown by the experiments, the information extracted from knowledge graphs significantly improved the effectiveness of the recommendation models. Overall, the approach based on *node2vec* obtained the better results. A similar intuition was proposed by Zhang et al. in [208]. In this case, the authors exploit *TransR*, a popular graph embedding method, together with stacked denoising autoencoders and stacked convolutional autoencoders, to learn a vector-space representation which is then used to feed the final integrated framework, which is termed as Collaborative Knowledge Base Embedding (CKE), that jointly learns the latent representations in collaborative filtering as well as items' semantic representations from the knowledge base. Also in this case, the experiments confirmed the effectiveness of the approach since the proposed method outperformed several widely adopted state-of-the-art recommendation methods.

To conclude, we can clearly state that this overview of recommender systems based on knowledge graphs confirmed that these methodologies can be helpful to obtain a more precise representation of user and items, that leads in turn to a more accurate generation of the recommendations. Moreover, as we will show in the next section, the use of knowledge graphs also allows to build effective *natural language explanations* supporting the recommendations, thus making these information sources as particularly relevant to design and implement semantics-aware content-based recommender systems.

3.3 User-Generated Content and Multimedia Features

In the last decade, we observed a number of works that use new types of side information, e.g., user-generated content (UGC) or metadata for recommendation.

With the emergence of the participatory Web (Web 2.0), various types of UGC became available, such as product reviews, user tags and forum discussions. Most of the early works that tried to leverage this information in the recommendation context focused on user-provided *tags*, which were used to enhance user and item profiles, or to explain recommendations to users [18, 88, 191]. In another stream of works, researchers focused on *user reviews* and tried to extract various types of information from them, including semantics or user opinions and sentiments, that can be used in the recommendation process [28]. More details are provided in Sect. 3.3.1.

On the other side, content-based recommender systems dealing with non-textual objects were commonly based on the use of metadata for representing descriptions. However, advances in audio, image and video analysis made it possible to represent multimedia objects by features that were extracted from the objects themselves. Those advanced representations could be effectively adopted to define advanced content-based recommender systems [42]. In Sect. 3.3.2, some pointers to the most relevant work concerning recommender systems leveraging multimedia content are presented and discussed.

3.3.1 Content-Based Recommender Systems Leveraging User-Generated Content

Recommender systems focused mostly on using user ratings or item metadata to generate recommendations, but another source of information that has been used to generate more relevant recommendations are *tags*. Tags are keywords that typically describe characteristics of the objects they are applied to, and can be made up of one or more words. Users are free to apply any type and any number of tags to an object, resulting in true bottom-up classification. Social tagging has been applied to many domains such as music, movies, books, and web sites [50, 55, 187, 199], where tags can be used as an additional resource to generate better recommendations. Tags have been effectively integrated into collaborative and content-based algorithms for item recommendations in several different ways. In collaborative filtering based on nearest neighbor algorithms, tags are exploited to aid in the calculation of the user/item similarities, or in some cases even replace the user-item rating matrix entirely, by relying on the less sparse user-tag and item-tag matrices to compute the similarities [91, 142, 187]. Tags have been also integrated into collaborative filtering algorithms based on latent factor models. In [108], the authors propose a tag-augmented version of matrix factorization, which integrates the latent factors of the item tags and ratings to provide a better approximation of the lower-rank user-item matrix. Tags have been also used to link different domains together for generating cross-domain recommendations [55]. Another possible way to include tags is in graph-based algorithms, which include tags as an additional node type and generate recommendations based on the resulting tripartite network, containing users, items and tags. As an example, the FolkRank algorithm runs personalized PageRank to assign a weight to each network node [80].

Tags are also suitable to be included in content-based recommender systems [21, 104]. Being free annotations, tags tend to suffer from syntactic problems, like polysemy and synonymy. Semantic content-based approaches have been proposed to address this problem. In [39], WordNet is adopted to perform Word Sense Disambiguation to content as well as tags, and a Naïve Bayes classifier is exploited to learn a probabilistic model of the user's disambiguated interests. This semantic user profile is then matched against the semantic item representations to locate the most relevant items for the active user.

However, the use of tags may be inadequate, especially when the target user has little historical data, or when the overall data sparsity level is high. A well consolidated approach is to leverage users' reviews for supporting their preferences and improve the recommendation process. User-generated reviews encapsulate rich semantic information such as the possible explanation of the users' preferences and the description of specific item attributes [28], thus they represent a rich source of information about the users' preferences, and can be exploited to build fine-grained user profiles. Therefore, a variety of review-based recommender systems have been proposed in the last years, which incorporate information extracted from user-generated textual reviews into the user modeling and recommending process. This may be beneficial to deal with the problem of data sparsity, cold-start for new users, and even in the case of dense data condition, may help to learn users' latent preference factors by considering the aspect opinions mentioned in the reviews [28]. In the survey by Chen et al. [28], various elements of valuable information that can be extracted from user reviews and that can be utilized by recommender systems have been identified. They range from quite simple information, such as frequently used terms, discussed topics, and overall opinions about reviewed items, to more complex information, such as specific opinions about item features, comparative opinions, reviewers' emotions, and reviews helpfulness.

Among the previous elements, opinions and sentiments expressed by users in their reviews about specific features or *aspects* of the reviewed items represent a promising approach to improve the recommendation process [62]. Aspect-based recommender systems leverage those opinions about aspects to provide improved personalized recommendations. Indeed, when items are evaluated with the same rating value, these systems are able to capture particular strengths and weaknesses of the items and, based on this information, better estimate their relevance for the target user [13, 132].

Aspect-based recommender systems include three main tasks, namely aspect extraction, to identify references to item aspects in user reviews, aspect polarity identification, to identify if the opinion on the aspects is positive, negative or neutral, and aspect-based recommendation, to exploit the extracted aspect opinion information to provide enhanced recommendations. In [78], a thorough investigation of the problem is presented, by separately addressing the three tasks.

Several methods for extracting opinions about items aspects have been proposed in literature [78]. They are classified as:

- vocabulary-based methods, that make use of lists of aspect words, as in [1]
- word frequency-based methods, in which words that have a high appearance frequency are selected as aspects. The methods to identify references to aspects in the reviews range from simple approaches based on the words frequently used in the reviews of a specific domain, to more complex ones based on language models [167], and on the comparison of the use of language when talking about a specific domain with respect to a general topic, in order to identify aspects mentioned in the reviews more often than usual [22]
- syntactic relation-based methods, where syntactic relations between words of a sentence are the basis for identifying aspect opinions, as in the Double Propagation algorithm in [153], which exploits syntactic relations between the words in a review to identify those that correspond to aspects
- topic model-based methods, where topic models are used to extract the main aspects from user reviews. Standard LDA models are modified so different generation distributions can focus on specific parts of the reviews [184, 185].

In the recommendation process aspects may be used in different ways. In [78], aspect-based recommender systems are categorized in approaches:

- enhancing item profiles with aspect opinion information, as in [48], where an item profile is composed of aspects with sentiment and popularity scores, and a case-based recommender matches the user's profile with items whose profiles are highly similar and produce greater sentiment improvements

- modeling latent user preferences on item aspects, as in [112], where a matrix factorization model incorporates hidden topics as a proxy for item aspects. The model aligns latent factors in rating data with latent factors in review texts
- deriving user preference weights from aspect opinions, as in [102], where the weight of an aspect in the user profile is determined by means of two factors, namely how much the user concerns about the aspect, and how much quality the user requires for such aspect. The value of concern is related to the frequency of comments of a user on specific aspects in his/her reviews, while the value of requirement increases when the user frequently rates an aspect lower than other users across different items
- incorporating aspect-level user preferences into recommendation methods, as in [132], where a multi-criteria user- and item-based collaborative filtering algorithm incorporating aspect opinion information is presented and evaluated. Similarity between users or items are computed according to the opinions expressed in the reviews. In [99], the authors propose to use latent multi-criteria ratings generated from user reviews to provide recommendations and capture latent complex heterogeneous user preferences. The latent rating generation process is based on two different models: the one-stage model, which utilizes document hashing to directly compute latent ratings, and the two-stage model, which utilizes a variational autoencoder to map user reviews into latent embeddings and subsequently compresses them into low-dimensional discrete vectors that constitute latent multi-criteria ratings.

3.3.2 Content-Based Recommender Systems Leveraging Multimedia Content

Traditional content-based recommender systems are usually fed by text, e.g. the description of a product, the plot of a movie, or the synopsis of a book [37, 104]. Even though audio or visual content are also associated with text to describe the items, they are usually not taken into account to generate recommendations, albeit they might have impact on user preferences.

Similarly to the architecture of classical content-based recommender systems dealing with textual content, a recommender leveraging multimedia content, such as audio or video, analyzes the items in order to represent them in a feature space, even though the pipeline for processing that kind of content is more complex.

One of the most popular tasks in which audio has been used is music recommendation, ranging from the classical task of recommending songs to a user based on her previous interests, to automatic playlist generation or continuation [19, 154].

Research in music recommender systems is becoming more and more interesting, also due to the increasing availability of music streaming services [168]. The challenge to recommend music relies on the fact that music tastes depends on a variety of factors, ranging from personality and emotional state of users [56], to contextual factors, such as weather conditions [20].

Content-based methods have shown to be useful when user feedback information is scarce, as in cold-start scenarios. As a source of content features to recommend music, social tags have been extensively used [92], even though features extracted from audio signals have also been used. The work in [6] presents a hybrid mood-aware music artist recommender system integrating both artists' and users' mood as well as audio features, such as timbre, tempo, loudness, and key confidence attributes to compute artists' pairwise similarities. This is a two-stage recommender system which identifies candidate artists based on the comparison of the mood of the user and the artist, and then re-ranks the list using artist similarity based on audio content of the artists' most popular songs. Deep learning approaches have been increasingly adopted for music recommendations, in particular in content-based systems which learn latent song or artist representations from the audio signal or from textual metadata. In [189], a CNN is adopted to represent each music item using 50-dimensional latent factors vectors, learned from log-compressed Mel spectrograms of music audio. The resulting latent factor representation of items is used together with latent user factors in a standard collaborative filtering fashion, and experiments on the Million Song Dataset show that this seems a viable method for recommending unpopular music.

Research in image recommendation includes approaches exploiting visual content extracted from images to recommend media items, e.g. paintings, or non-media items, e.g. clothes based on the appearance of photos. For example, the work in [2] describes a recommender system which uses as input the observed painting and generates a list of recommended paintings as output. Recommendations are generated using an algorithm resembling the PageRank strategy, which takes into account the past behavior of individual users, the overall behavior of the entire community of users, and intrinsic features of multimedia objects (low-level and semantic similarities). The past behavior of each individual users is given in terms of the browsing history of objects of that specific user, while the behavior of the whole community of users takes into account the browsing history of any user. The features of the multimedia objects have been used to compute their similarity, using low-level visual features, such as color, texture, and shape, and metadata such as painter, genre, and subject. The system has been effectively evaluated for recommending paintings in a virtual museum scenario containing digital reproductions of paintings in the Uffizi Gallery.

Recommending products leveraging visual content is adopted in several domains, such as fashion, food, and tourism [42]. In the fashion domain, McAuley et al. [113] propose a recommender system to match clothes with accessories by exploiting their images. The authors use freely available data collected from the Amazon, containing millions of relationships between a pool of objects, to identify alternative or complementary pair of products. The gist of the approach is to develop a fashion recommender able to model the human sense of relationship between objects by utilizing the visual appearance of products. More complex works model different levels of a user's preference for different parts of items. In [32], the authors learn a part-based user model based on different partitions of the image to obtain a personalized recommendation model. Experiments on a dataset from Amazon.com including images of helmets, t-shirts, and watches yielded improved results over existing textual or visual recommender systems that disregard appealing differences between parts of products. A similar approach is proposed in [67], where fine-grained facial attributes such as gender, race, eyebrow thickness, skin color, fatness, and hair color are extracted from a frontal face photo to recommend the best-fit eyeglasses using a probabilistic model.

In the food domain, multimodal information such as recipe images or ingredients are taken into account in the recommendation process. The results of a study presented in [49] indicate that preferred recipes could be predicted by leveraging low-level image features and recipe meta-data. Similarly in the tourism domain, visual features extracted from images shared by users are used to model their tastes for enhancing Points of Interests (POI) recommendation. In [195], the authors adopt a CNN pre-trained on ImageNet to extract visual features from images shared on Instagram and a probabilistic matrix factorization algorithm to model interactions between visual content, users and locations, with good results in coping with cold start as well.

Finally, visual characteristics of the content could be also adopted to recommend videos, such as movies, and most of the approaches analyze trailers, movie clips, or posters instead of the full movie. In [40], the authors designed a content-based movie recommender system which takes into account stylistic visual features to distinguish for example between comedy movies, usually made with a large variety of bright colors, and horror films using dark hues. To this purpose, stylistic visual features as shot length, color variation, lighting key, and motion vectors have been adopted, and results of the experiments have shown that low-level visual features provide better recommendations than the high-level features, such as genres.

The work presented in [41] combines audio and visual features with movie metadata, such as genre and cast, into a unique representation, called the Movie Genome, in order to deal with the new item problem. The authors proposed a novel recommendation model, called collaborative-filtering-enriched content-based filtering (CFeCBF), which exploits the collaborative knowledge of videos with interactions (warm items) to weight content information for videos without interactions (cold items). More details about multimedia recommender systems can be found in Chap. "Multimedia Recommender Systems: Algorithms and Challenges".

3.4 Transparency and Content-Based Explanations

The importance of content-based information has been largely discussed throughout this chapter. Moreover, a recent and interesting trend that further confirms the effectiveness of such features lies in their exploitation to *explain* or *justify* a recommendation.

As already discussed in Chap. "Beyond Explaining Single Item Recommendations", the idea of providing intelligent information systems with *explanation* facilities was studied from the early 1990s [89], and gained again attention in the light of the recent *General Data Protection Regulation (GDPR)*,[4] which emphasized and regulated the *users' right to explanation* [64] when people face machine learning-based (or, generally speaking, artificial intelligence-based) systems.

This a very relevant problem, since intelligent systems are becoming more and more important in our everyday lives. Accordingly, it is fundamental that the *internal mechanisms* that guide these algorithms are as clear as possible. The need for *transparent* algorithms is even more felt for RS since, as shown by Sinha et al. [175], the more the *transparency* of the algorithm, the more the *trust* the user puts in the system. Similarly, Cramer et al. [36] proved the relationship between the transparency of a RS and users' acceptance of the recommendations.

The interest of the community in the topic is also confirmed by the spread of several research discussing the positive impact of explanations for recommender systems [63, 93, 115]. In this section we will focus our attention on *content-based features*, thus we will provide an overview of recent approaches to *explain* or *justify* a recommendation based on content.

According to the taxonomy of explanation strategies in RSs provided by Friedrich et al. [58], approaches to generate explanations and justifications can be split into two categories: *white box* methodologies, which generate an explanation which is directly connected to the underlying explanation method and *black box* methodologies, where the explanation strategy is not aware (and is independent) of the underlying recommendation model which is used to generate the suggestions. In the following, we will refer to the first as *explainable recommendations* strategies, while the latter are referred to as *post hoc explanation* strategies.

3.4.1 Generating Explainable Content-Based Recommendations

The idea behind *explainable recommendation strategies* is to encode content-based features in the recommendation model and to exploit them to generate a natural language explanation that supports the recommendation. One of the early attempts in the area is presented in [182], where the authors proposed a model to generate explanations which exploits the overlap between the features of the user profile and those describing the suggestion.

More recently, several methods were devoted to the explanation of the recommendation coming from matrix factorization techniques. As an example, in [207], the authors propose an approach based on the exploitation of users' reviews, since they extract explicit product features and then aligns each latent dimension in order to explain recommendation coming from matrix factorization techniques.

[4] http://ec.europa.eu/justice/data-protection/reform/files/regulation_oj_en.pdf.

However, due to the recent spread of deep learning methods [97], most of the recent research effort focuses on designing *explainable* recommendation methods based on deep neural networks. As an example, in [29] the authors propose NARRE (Neural Attentional Regression model with Review-level Explanations). The core of the approach is an attention mechanism that catches the usefulness of reviews. Such information is used to: (1) predict the interest of the user towards the item and (2) predict the usefulness of each review, simultaneously. Therefore, the reviews labeled as highly-useful are exploited to provide the user with review-level explanations. Similarly, the use of reviews is also investigated in [11], where an approach based on deep neural networks that quantifies the relationship between aspects and reviews is proposed. In particular, the authors build a user-aspect bipartite relation as a bipartite graph. Next, by using dense sub-graph extraction and ranking-based technique an *explainable recommendation* is returned. Finally, in [75] the authors exploited a tripartite modeling of user-item-aspect tuples and used graph-based ranking to find the most relevant aspects of a user that match with relevant aspects of places. These aspects can be used to both drive the recommendation and explanation process.

Finally, several work propose the adoption of *attention mechanisms* as a source to *explain* recommendation. As an example, in [30] the authors propose an Attention-driven Factor Model (AFM), which learns and tunes the user attention distribution over features. This is used to drive both the recommendation and the explanation process, since the features with the highest attention can be used as explanation. Similarly, a method to jointly optimize matrix factorization and attention-based GRU network is proposed in [171]. In this case, the matrix component is used to drive the recommendation while the attention-based mechanism is used to generate a suitable explanation. However, it should be pointed out that some literature [83] argued that *attention* and *explanation* refer to two different concepts, thus the effectiveness of attention modules for explanation purposes tends to be overestimated in current research.

3.4.2 Generating Post Hoc Content-Based Explanations

Differently from *explainable* recommendation methods, that directly encode content-based features in the model and use these features to explain a suggestions, *post hoc* explanations strategies generate an explanation *after* the recommendation process, thus they are completely *independent* from the underlying recommender system.

One of the first attempts is discussed in [191], where the authors used *tags* to generate explanations. As shown in [151], these strategies allow to maintain a good predictive accuracy whilst yielding content-based explanations. In this research line, it is also possible to mention the work proposed in [206], where the authors extract causal rules from user history to provide personalized, item-level, post hoc explanations. In this case, the causal explanations are extracted through a perturbation model and a causal rule learning model. Recently, several work tried to exploit *topic models* to generate content-based explanations independent

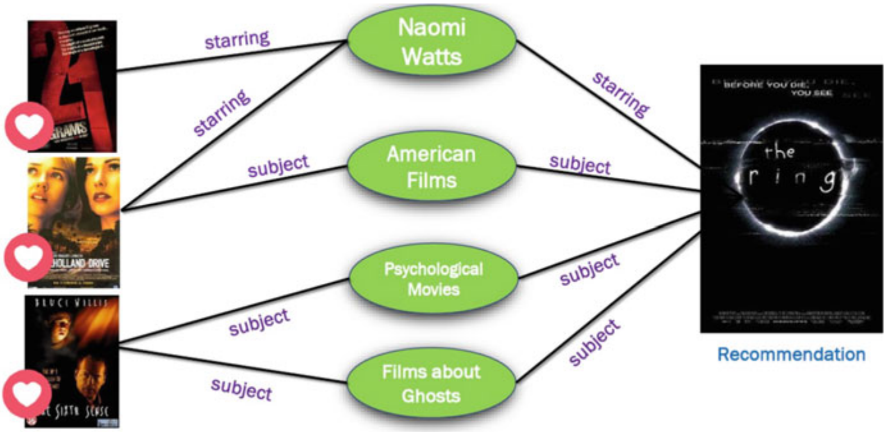

Fig. 6 Graph-based data model to generate explanations

from the recommendation model: as an example, in [161] the authors collect from `Freebase` textual data about movies and design a method to map latent factors to the topic emerging from textual content.

Moreover, one of the most promising research direction lies in the extraction of content-based information from open knowledge sources and knowledge graphs. As an example, ExpLOD [130] is a framework which exploits the information available in `DBpedia` [17] cloud to generate a *natural language explanation*. The methodology is based on a graph in which the items liked by a user are connected to the items recommended through the properties available in `DBpedia`. Basically, this graph connects the items the user liked and those in her recommendation list through the values of the properties describing those items in the knowledge graph. An example of such a data model is reported in Fig. 6. Given such a data module, the explanation is based on a natural language *template* which is filled in by exploiting the *top-K* overlapping properties.

As an example, by referring to the *toy* example reported in Fig. 6, the output of the framework would have been *'I suggest you The Ring, because you often watch movies starred by Naomi Watts and you have also liked psychological movies, such as The Sixth Sense'*. Similar attempts, based on the exploitation of structured features gathered from knowledge graphs are also presented in [5] and in [107], where the authors specifically designed a strategy for the tourism domain.

As shown by the authors, these explanations significantly overcome other simple strategies to generate content-based explanations [138]. These findings are confirmed in a recent work [174], where several strategies to generate post hoc explanations are compared.

Beyond features gathered from knowledge graphs, other attempts exploited review-based features for explanation purposes. In this research line, Muhammad et al. [123] introduced the concept of *opinionated explanations*, that is to say, explanations mined from user-generated reviews. To this end, the authors identify relevant aspects of the items (e.g., bar, service, parking, etc.), and present the most relevant ones. In this case, the authors did not justify the *reasons* behind the choice of highlighting a specific characteristic. In other terms, this approach lets the users be informed about the main characteristics of the items, but does not *explain* why *, e.g.,* the bar or the service are particularly good.

Differently from the approaches in which aspects in the reviews are manually extracted, in [137], the authors propose a framework to automatically generate review-based justifications of the recommendations. In particular, the authors first run Part-of-Speech (POS) tagging algorithm over the set of the reviews to obtain representative concepts. Next, they implement ranking strategies to identify the most relevant aspects that best describe and characterize the item. Finally, excerpts of users' reviews discussing those aspects with a positive sentiment are dynamically combined to fill in a natural language template in the GENERATION module. This represents the final output of the algorithm, which is provided to the user as *justification* of the recommendation she received.

By referring again to the previous toy example concerning 'The Ring', a likely justification would have been: *I recommend you The Ring because people who liked the movie think that the plot delivers some bone-chilling terror. Moreover, people liked The Ring since the casting is pretty good*.[5] As shown by the authors in their research [141], users tended to prefer review-based explanation to feature-based explanation since the identification of relevant reviews excerpts allowed them to discover new information about the recommended items. Further extensions of this framework have been presented in [139], where the authors adopt text summarization techniques to automatically generate a summary of relevant reviews excerpts, and in [140], where a context-aware extension of the framework, which aims to differentiate the justification based on the different *context of consumption* of the items, is discussed.

3.5 Exploiting Content for Conversational Recommender Systems

Conversational Recommender Systems (CRS) are gaining more and more attention in the last few years. This renewed interest probably derives from the massive diffusion of Digital Assistants, such as Amazon Alexa, Siri, Google Assistant, that allow users to execute a wide range of actions through messages in natural language. In fact, the main distinguishing aspect of a CRS, compared to a canonical recommender system, is its capability of interacting with the user during the recommendation process [109] by means of a multi-turn dialog [87]. CRSs make the

[5] In this case, we assume that the aspect extraction module would have identified plot and casting as hallmarks of the movie.

interaction more efficient and natural [198], as they are able to help users navigate complex product spaces by iteratively selecting items for recommendation [176].

During the interaction with a recommender system (canonical or CRS), the following four main phases can usually be identified: preference acquisition, recommendation generation, explanation, and user feedback acquisition. Of course, some steps may be optional. For example, explanations are not provided by all recommender systems. The peculiarity of a CRS lies in the fact that the acquisition of user preferences and needs becomes crucial. Indeed, users do not provide the system with all the preferences in one step. Conversely, they usually provide and refine their feedback during the dialog.

In this context, there are two main scenarios:

1. the user starts the dialog by providing characteristics or features that the ideal item should have [146];
2. the system asks questions on constraints of features and the user answers [27, 186]. These two scenarios can be combined during the dialog, of course. For example, the user provides some initial criteria, then receives an initial set of recommendations, and starts to revise the initial preferences since recommendations do not meet the *ideal* item. This is a typical situation in *critiquing-based* approaches, where the user can add new constraints (*tightening* [158]) or *relax* some others [84, 157]. Further details on the recommendation process in an interactive recommender system are analyzed in [74].

From an architectural perspective, a fundamental distinction can be made between systems that implement a *dialog state tracker* and systems that implement an *end-to-end* architecture [47]. The former require a fine grained definition of internal dialog states and precisely defined user intents. As a consequence, they can hardly scale to large domains characterized by dialogs with high variability in terms of language. This kind of systems usually implement a *pipeline-based* architecture made up of a series of modules, each one with its own specific function [47, 101, 202, 210]. Conversely, end-to-end dialog systems do not rely on explicit internal states, thus they do not need state tracking modules. They combine components that are trained on conversational data and that handle all the steps, from understanding the user message to generating a response.

The next section analyzes the impact of content on these two types of approaches and, more generally, on CRSs. As we will see, the role of textual content is different on the ground of the implemented solution.

3.5.1 Dialog State-Based CRS

A CRS that explicitly manages the dialog state usually consists of the following modules: *Intent Recognizer*, *Dialog Manager*, *Recommender System*, some NLP modules such as an *Entity Recognizer* and a *Sentiment Analyzer*. An example of this modular architecture is implemented in the ConveRSE framework [82] and is shown in Fig. 7.

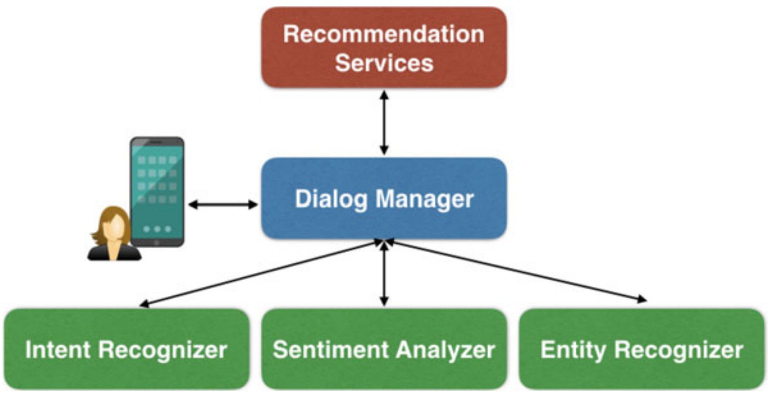

Fig. 7 An example of modular architecture for a CRS

Dialog Manager This is the core component of the architecture since its responsibility is to supervise the whole recommendation process. The *Dialog Manager* (DM) is the component that keeps track of the *dialog state*. It can be viewed as the orchestrator of the system, and strictly depends on the task the dialog agent is dealing with. The DM receives the user message, invokes the components needed for answering to the user request, and returns the message to be shown to the user. When the information for fulfilling the user request is available, the DM returns the message to the client.

According to Williams et al. [203] algorithms for dialog-state-tracking can be based on *hand-crafted rules, generative models,* and *discriminative models*.

Approaches based on *hand-crafted rules* require no training data to be implemented, and this represents a huge advantage in some contexts. Furthermore, designers can incorporate domain knowledge through rule definition. The main weakness of these models is that they consider a single hypothesis for the dialog state. Actually, multiple competing dialog states can occur at any given stage of the conversation, and the choice of the right dialog state represents a key aspect. In fact, the actions the system should take depend on the dialog state. Accordingly, more recent approaches assign a score to each dialog state [193]. However, also the computation of this score requires parameter tuning, which cannot be based on dialog data. This represents an additional limitation for this kind of approaches.

Generative approaches assume that the dialog can be modeled through a Bayesian network. These models make independence assumptions (which generally are invalid) in order to reduce the complexity. For example, they can assume that user errors are equally distributed, while some errors are much more frequent than others [200]. Parameters of generative models are often hand-crafted. Usually, the next dialog state depends on the previous state, but some authors introduce terms that accumulate dialog history [201], or terms that express common-sense relationships between user actions [43].

Discriminative approaches directly model the distribution over the dialog state, given arbitrary and possibly correlated input features [200]. Discriminative approaches can exploit machine-learning algorithms (e.g., maximum entropy linear classifiers, neural networks, web-style ranking models). These models are able to integrate a large number of features and have the goal of optimizing the predictive accuracy. Most of the approaches encode the dialog history in the feature in order to learn a classifier [200]. Handerson et al. [76] apply a deep neural network as classifier. Some other approaches adopt sequential models such as Markov models [156].

Intent Recognizer This component is strictly connected to the Dialog Manager and has the goal of defining the intent of the user from the utterances formulated through a natural language sentence. When the user sends a message, the recommender system must first understand what is the goal of the user and what she wants to express and get by that message. Hence, as first step, the CRS needs to identify the intent of the user.

Four main intents can be identified in a CRS, each one corresponding to a specific step in the recommendation process:

- *preference:* the user is providing a preference on an item or on a feature
- *recommendation:* the user is asking to receive a recommendation
- *critiquing:* the user is providing a feedback on the received recommendation
- *explanation:* the user is asking to receive an explanation on a recommendation.

The Intent Recognizer usually requires a set of utterances for training each intent. From this perspective, intent recognition can be viewed as a text classification task.

Entity Recognizer The aim of the *Entity Recognizer* (ER) module is to find relevant entities mentioned in the user sentence, and to link them to the correct concept in a Knowledge Base (KB).

A classical entity linking approach consists of two steps: *spotting* and *disambiguation*.

The *spotting* step analyzes the text in order to discover candidate entities. Specifically, the algorithm detects a sequence of words (*surface form*) that represent an entity and retrieves all the concepts that can be associated to that surface form. The *disambiguation* step consists in selecting the correct concept for each surface form.

As an example, let us suppose the user writes the following sentence: "I like Michael Jackson and Beat It". "Beat It" is an ambiguous entity since it might be referred to the candidate concept *Beat It by Michael Jackson* or to *Beat It by Sean Kingston*. Hence, this surface form needs to be disambiguated. In order to accomplish this task, a possible strategy the ER might implement is based on the computation of the similarity between the candidate entities (i.e., *Beat It by Michael Jackson* and *Beat It by Sean Kingston* and the other entities in the context (i.e., *Michael Jackson*). In this example, the similarity between *Michael Jackson* and *Beat It by Michael Jackson* will be higher than the one between *Michael Jackson* and *Beat*

It by Sean Kingston. Therefore, *Beat It by Michael Jackson* will be chosen. More details on this ER implementation can be found in [144].

Sentiment Analyzer The *Sentiment Analyzer* (SA) has the goal of assigning the right sentiment polarity (i.e., positive and negative) to the entities mentioned in the sentences (e.g., singer, song, genre) and identified by the ER module. The difficulty lies in the fact that the same sentence can contain different sentiment tags (even opposite) as well as different entities. For example, given the sentence "I like Rocket Man, but I don't like Gus Dudgeon", the SA identifies a positive sentiment (i.e., like) and a negative one (i.e., don't like). SA should assign the positive sentiment to the entity *Rocket Man* and the negative sentiment to the entity *Gus Dudgeon*.

Recommendation Services This component collects the services strictly related to the recommendation process. The *recommendation algorithm* is invoked by the DM when all the information required for generating the recommendations are available.

Henceforth, we will analyze some relevant approaches by highlighting the dialog state they are focused on and the exploited content. As regards the *preference acquisition*, it can be easily imagined that preferences are mostly expressed by the user during the dialog in terms of characteristics that an ideal item should have (e.g., "I would like to have dinner in a restaurant with a beautiful sea view"). Actually, a CRS could also acquire the preferences by asking feedback on individual items [45]. However, preferences expressed in terms of item facets are more interesting from our point of view since content plays a crucial role in that case. This modality of preference acquisition is also known as *slot filling* [87] and it establishes a straightforward relation between CRSs and content-based recommender systems. Indeed, while a preference given on a specific item (e.g., "I like the movie American Beauty") can also be exploited by a collaborative recommendation algorithm, preferences in terms of item facets necessarily need that the recommendation algorithm is able to exploit this information for generating recommendations. From this perspective, it also emerges the similarity between CRSs and constraint-based recommender systems as well as knowledge-based ones. In fact, during the dialog with the user, several CRSs perform *filtering* and *re-ranking* of the items through constraint-based techniques [52]. For example, the knowledge-based recommender environment named CWAdvisor [53] proposes a set of questions to the user, each one associated to set of predefined answers. The goal of this step is to acquire customer properties and constraints, and matching them with product properties. In this way the system is also able to *explain* the recommendation by providing the user with the reasons why a given product suits the customer needs and wishes. In a CRS powered by a content-based recommendation algorithm, the idea is basically the same, with the difference that preferences are acquired through a multi-turn dialog, as can be seen in Adaptive Place Adivsor (APA) [183]. The interaction in APA takes the form of a sequence of questions whose goal is to discard items not interesting for the user. Hence, the goal of the dialog is to acquire attribute-value specifications, such as *cuisine = Chinese*. APA ends the search of the *ideal* item for the user when a small number of items match the constraints and are highly similar. Still on the subject of the adopted *recommendation strategy*, Argal et al. [7] propose a hybrid recommendation algorithm that combines collaborative and content-based paradigms in a CRS. The content-based algorithm is exploited when the system does not register any user activity (cold-start stage), otherwise collaborative recommendations are provided. The content-based algorithm performs a matching of user preferences against product representations stored in Elasticsearch.[6] Recently, Li et al. [98] focused their attention on the *critiquing* step and proposed a latent-linear-critiquing model for CRSs. A user is iteratevely provided with item recommendations and attribute descriptions for those items. The user can accept the recommendation or can critique an attribute to generate a new recommendation. The model exploits preferences implicitly revealed from user reviews. User critiques are then transformed in an embeddable term-frequency representation that can be co-embedded with user preferences. Item descriptions are keyphrases from user reviews.

3.5.2 End-to-End Systems

End-to-end systems have become popular thanks to the spread of Deep Learning techniques [47]. End-to-end systems are a promising solution because they do not require the development of specific and complex rules for managing the dialog and its states, which makes them easier to port to a new domain.
However, there are still some challenges to face in order to enable them to handle goal-oriented conversations [47]. One of the main obstacles is the lack of datasets for training deep-learning models. Indeed, in this case content plays an *unusual* role compared to the one we are accustomed to observe in a recommender system. Content, in the form of recorded dialogs, is essential for learning at least the two actions of providing preferences and generating recommendations [87]. Recently, three datasets that contain utterances in the movie, book, and music domains have been released.[7] These datasets consist of real dialogs between users and a CRS, collected during an experimental session with the CRS [81]. However, the number of utterances (5,318 movie, 1,862 book, 2,096 music) is probably not enough for learning end-to-end components.

Ritter et al. [159] proposed one of the first approaches which adopts an end-to-end architecture for conversational agents. In that work, the authors defined a data-driven method to generate responses to open-domain linguistic stimuli, based on phrase-based Statistical Machine Translation (SMT). They demonstrated that SMT techniques perform better than information retrieval approaches on the task of response generation. Recently, Jannach and Manzoor [85] analyzed the utterances generated by two novel end-to-end models for CRSs. More specifically, they compared *DeepCRS* [100] and *KBRD* [31]. DeepCRS has a set of sub-components

[6] https://www.elastic.co/elasticsearch//.

[7] https://github.com/aiovine/converse-dataset.

for sentence encoding, next-utterance prediction, sentiment classification, and recommendation. KBRD has a dialog system based on a transformer-based sequence-to-sequence module and a knowledge graph which provides knowledge about the domain. A switching network connects the two modules. The comparative analysis between KBRD and DeepCRS showed that one third of the system utterances were not meaningful for each system in the given context, and less than two third of the recommendations were meaningful. As a result, the quality of responses and recommendations of end-to-end CRSs showed ample room for improvement at this stage. Both systems have been trained on the ReDial dataset[8] that is annotated dataset consisting of 10,006 dialogues, 182,150 utterances, 956 users, and 51,699 movie mentions. In each conversation users recommend movies to each other and the number of movies mentioned varies. The dataset has been developed by Amazon Mechanical Turk. In each dialog there are two roles: the recommendation seeker and the recommender. The movie seeker has to explain what kind of movie she likes, and asks for movie recommendation. The recommender tries to understand the seeker tastes, and recommends movies. All exchanges of information and recommendations are made using natural language.

4 Discussion and Future Outlook

This chapter covered the recent advances in the area of *semantics-aware content-based recommendations*. As we have shown, most of the current work can be split into three main research lines: (1) techniques investigating new methods for *representing* content-based features; (2) techniques investigating new *sources* to gather content-based features; (3) new *use cases* for content-based features.

In the first group, we have discussed work based on *word embedding techniques*. The use of vector space representations, together with deep architectures, is probably the most popular research direction in the area. In this sense, the adoption of *contextual word representations* (discussed in Sect. 3.1.2), such as BERT, ELMo and so on, is a research direction particularly worthy of attention in the next years. Indeed, these models proved to overcome state-of-the-art techniques on several NLP tasks, thus they are gaining more and more attention in recommendation scenarios as well. In particular, their ability to model the precise meaning of words and sentences can be useful to develop more accurate recommendation strategies.

As for the sources of information, in Sect. 3.2 we introduced approaches based on the exploitation of *knowledge graphs* and techniques based on *user-generated content* and *multimedia features*. In this research line, we expect to see more and more work trying to merge heterogeneous groups of features through deep learning architectures. As an example, we consider as very promising the adoption of *graph embedding techniques*, since they learn a vector space representation that allows to

[8] https://redialdata.github.io/website/.

merge word embeddings with the information gathered from knowledge graphs. In Sect. 3.2 we already discussed some work investigating this intuition, but we expect an increase of the research trying to put together collaborative information (e.g., ratings), content-based information (e.g., review data, descriptive text, multimedia features, if any) and structured features available in *knowledge graphs*. In this sense, research investigating the *best* strategies to combine these features is still at an early stage.

Finally, as new use cases we focused our attention on the usage of content to generate *natural language explanations* and to design *conversational recommender systems*. In this case, we emphasize the importance of explanation facilities and more elaborated interaction models in recommender systems research. On the one side, explanations are fundamental to *open* black-box recommendation models, to make users *aware* of the underlying algorithms as well as to further increase their trust. On the other side, more *natural* interaction strategies, based on conversational recommender systems and dialog mechanisms, can be helpful to take the final step towards the adoption of recommendation algorithms in very *sensitive* domains, such as medicine and finance. In this sense, we expect a significant effort in developing personalized conversational agents for both those domains.

5 Conclusions

All the literature we discussed throughout the chapter gave evidence of the importance of both content-based features and semantic representations. Indeed, the usage of content can help to design and develop more accurate recommendation models in a broad range of domains and applications, as shown in this chapter, since it is able to effectively tackle typical problems of *pure* collaborative models, such as cold start, new item problems and *opacity* of the recommendation models. We think that content-based information and semantics-aware representation strategies will play an increasingly central role in recommender systems research in the next years.

It is our hope that this chapter may stimulate the research community to adopt and effectively integrate the discussed techniques in several recommendation scenarios in order to foster future innovations in the area of content-based recommender systems.

References

1. S. Aciar, D. Zhang, S.J. Simoff, J.K. Debenham, Informed recommender: basing recommendations on consumer product reviews. IEEE Intell. Syst. **22**(3), 39–47 (2007)
2. M. Albanese, A. d'Acierno, V. Moscato, F. Persia, A. Picariello, A multimedia recommender system. ACM Trans. Internet Techn. **13**(1), 3:1–3:32 (2013)

3. G. Alexandridis, T. Tagaris, G. Siolas, A. Stafylopatis, From free-text user reviews to product recommendation using paragraph vectors and matrix factorization, in *Companion Proceedings of the 2019 World Wide Web Conference* (2019), pp. 335–343
4. A. Almahairi, K. Kastner, K. Cho, A. Courville, Learning distributed representations from reviews for collaborative filtering, in *Proceedings of the 9th ACM Conference on Recommender Systems* (2015), pp. 147–154
5. M. Alshammari, O. Nasraoui, S. Sanders, Mining semantic knowledge graphs to add explainability to black box recommender systems. IEEE Access **7**, 110563–110579 (2019)
6. I. Andjelkovic, D. Parra, J. O'Donovan, Moodplay: interactive music recommendation based on artists' mood similarity. Int. J. Hum. Comput. Stud. **121**, 142–159 (2019)
7. A. Argal, S. Gupta, A. Modi, P. Pandey, S. Shim, C. Choo, Intelligent travel chatbot for predictive recommendation in echo platform, in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)* (IEEE, Piscataway, 2018), pp. 176–183
8. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, *DBpedia: A Nucleus for a Web of Open Data* (Springer, Berlin, 2007)
9. R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, vol. 463 (ACM Press New York, 1999)
10. T. Bansal, D. Belanger, A. McCallum, Ask the GRU: multi-task learning for deep text recommendations, in *Proceedings of the 10th ACM Conference on Recommender Systems* (2016), pp. 107–114
11. R. Baral, X. Zhu, S. Iyengar, T. Li, ReEL: review-aware explanation of location recommendation, in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization* (ACM, New York, 2018), pp. 23–32
12. O. Barkan, N. Koenigstein, Item2vec: neural item embedding for collaborative filtering, in *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)* (IEEE, Piscataway, 2016), pp. 1–6
13. K. Bauman, B. Liu, A. Tuzhilin, Aspect based recommendations: recommending items with the most valuable aspects based on user reviews, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13–17, 2017* (ACM, New York, 2017), pp. 717–725
14. M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
15. V. Bellini, T. Di Noia, E. Di Sciascio, A. Schiavone, Semantics-aware autoencoder. IEEE Access **7**, 166122–166137 (2019)
16. I. Beltagy, K. Lo, A. Cohan, Scibert: a pretrained language model for scientific text (2019). arXiv:1903.10676
17. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, DBpedia-a crystallization point for the Web of Data. J. Web Semantics **7**(3), 154–165 (2009)
18. T. Bogers, Tag-based recommendation, in *Social Information Access - Systems and Technologies*. Lecture Notes in Computer Science, vol. 10100 (Springer, Berlin, 2018), pp. 441–479
19. G. Bonnin, D. Jannach, Automated generation of music playlists: survey and experiments. ACM Comput. Surv. **47**(2), 26:1–26:35 (2014)
20. M. Braunhofer, M. Kaminskas, F. Ricci, Location-aware music recommendation. Int. J. Multim. Inf. Retr. **2**(1), 31–44 (2013)
21. I. Cantador, A. Bellogín, D. Vallet, Content-based recommendation in social tagging systems, in *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26–30, 2010*, ed. by X. Amatriain, M. Torrens, P. Resnick, M. Zanker (ACM, New York, 2010), pp. 237–240
22. A. Caputo, P. Basile, M. de Gemmis, P. Lops, G. Semeraro, G. Rossiello, SABRE: a sentiment aspect-based retrieval engine, in *Information Filtering and Retrieval*, ed. by C. Lai, A. Giuliani, G. Semeraro. Studies in Computational Intelligence, vol. 668 (Springer, Berlin, 2017), pp. 63–78
23. H. Caselles-Dupré, F. Lesaint, J. Royo-Letelier, Word2vec applied to recommendation: hyperparameters matter, in *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), pp. 352–356

24. G. Cenikj, S. Gievska, Boosting recommender systems with advanced embedding models, in *Companion Proceedings of the Web Conference 2020, WWW'20* (Association for Computing Machinery, New York, 2020), pp. 385–389
25. D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R.S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al., Universal sentence encoder (2018). arXiv:1803.11175
26. B.P. Chamberlain, E. Rossi, D. Shiebler, S. Sedhain, M.M. Bronstein, Tuning word2vec for large scale recommendation systems, In *Fourteenth ACM Conference on Recommender Systems* (2020), pp. 732–737
27. L. Chen, P. Pu, Preference-based organization interfaces: aiding user critiques in recommender systems, in *International Conference on User Modeling* (Springer, Berlin, 2007), pp. 77–86
28. L. Chen, G. Chen, F. Wang, Recommender systems based on user reviews: the state of the art. User Model. User-Adap. Inter. **25**(2), 99–154 (2015)
29. C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee (2018), pp. 1583–1592
30. J. Chen, F. Zhuang, X. Hong, X. Ao, X. Xie, Q. He, Attention-driven factor model for explainable personalized recommendation, in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), pp. 909–912
31. Q. Chen, J. Lin, Y. Zhang, M. Ding, Y. Cen, H. Yang, J. Tang, Towards knowledge-based recommender dialog system, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, ed. by K. Inui, J. Jiang, V. Ng, X. Wan (Association for Computational Linguistics, Stroudsburg, 2019)
32. H. Chi, C. Chen, W. Cheng, M. Chen, UbiShop: commercial item recommendation using visual part-based object representation. Multim. Tools Appl. **75**(23), 16093–16115 (2016)
33. N. Chinchor, P. Robinson, MUC-7 named entity task definition, in *Proceedings of the 7th Conference on Message Understanding*, vol. 29 (1997), pp. 1–21
34. A. Collins, J. Beel, Document embeddings vs. keyphrases vs. terms: an online evaluation in digital library recommender systems (2019). arXiv:1905.11244
35. A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data (2017). arXiv:1705.02364
36. H. Cramer, V. Evers, S. Ramlal, M. Van Someren, L. Rutledge, N. Stash, L. Aroyo, B. Wielinga, The effects of transparency on trust in and acceptance of a content-based art recommender. User Model. User-Adapt. Interact. **18**(5), 455–496 (2008)
37. M. de Gemmis, P. Lops, C. Musto, F. Narducci, G. Semeraro, Semantics-aware content-based recommender systems, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira (Springer, Berlin, 2015), pp. 119–159
38. S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
39. M. Degemmis, P. Lops, G. Semeraro, P. Basile, Integrating tags in a semantic content-based recommender, in *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23–25, 2008*, ed. by P. Pu, D.G. Bridge, B. Mobasher, F. Ricci (ACM, New York, 2008), pp. 163–170
40. Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, M. Quadrana, Content-based video recommendation system based on stylistic visual features. J. Data Semant. **5**(2), 99–113 (2016)
41. Y. Deldjoo, M.F. Dacrema, M.G. Constantin, H. Eghbal-zadeh, S. Cereda, M. Schedl, B. Ionescu, P. Cremonesi, Movie genome: alleviating new item cold start in movie recommendation. User Model. User Adapt. Interact. **29**(2), 291–343 (2019)
42. Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, Recommender systems leveraging multimedia content. ACM Comput. Surv. **53**(5), 106:1–106:38 (2020)

43. D. DeVault, M. Stone, Managing ambiguities across utterances in dialogue, in *Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue (Decalog 2007)* (2007), pp. 49–56
44. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding (2018). arXiv:1810.04805
45. L.W. Dietz, S. Myftija, W. Wörndl, Designing a conversational travel recommender system based on data-driven destination characterization, in *ACM RecSys Workshop on Recommenders in Tourism* (2019), pp. 17–21
46. T. Di Noia, R. Mirizzi, V.C. Ostuni, D. Romito, Exploiting the web of data in model-based Recommender Systems, in *Proceedings of the Sixth ACM conference on Recommender Systems* (ACM, New York, 2012), pp. 253–256
47. J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, J. Weston, Evaluating prerequisite qualities for learning end-to-end dialog systems (2015). arXiv:1511.06931
48. R. Dong, M. Schaal, M.P. O'Mahony, K. McCarthy, B. Smyth, Opinionated product recommendation, in *Case-Based Reasoning Research and Development - 21st International Conference, ICCBR 2013, Saratoga Springs, NY, USA, July 8–11, 2013. Proceedings*, ed. by S.J. Delany, S. Ontañón. Lecture Notes in Computer Science, vol. 7969 (Springer, Berlin, 2013), pp. 44–58
49. D. Elsweiler, C. Trattner, M. Harvey, Exploiting food choice biases for healthier recipe recommendation, in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7–11, 2017*, ed. by N. Kando, T. Sakai, H. Joho, H. Li, A.P. de Vries, R.W. White (ACM, New York, 2017), pp. 575–584
50. M. Enrich, M. Braunhofer, F. Ricci, Cold-start management with cross-domain collaborative filtering and tags, in *E-Commerce and Web Technologies - 14th International Conference, EC-Web 2013, Prague, Czech Republic, August 27–28, 2013. Proceedings*, ed. by C. Huemer, P. Lops. Lecture Notes in Business Information Processing, vol. 152 (Springer, Berlin, 2013), pp. 101–112
51. K. Ethayarajh, How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, Nov. 2019* (Association for Computational Linguistics, Stroudsburg, 2019), pp. 55–65
52. A. Felfernig, G. Friedrich, D. Jannach, M. Zanker, Constraint-based recommender systems, in *Recommender Systems Handbook* (Springer, Berlin, 2015), pp. 161–190
53. A. Felfernig, G. Friedrich, D. Jannach, M. Zanker, An integrated environment for the development of knowledge-based recommender applications. Int. J. Electron. Commerce **11**(2), 11–34 (2006)
54. C. Fellbaum, *WordNet: An Electronic Lexical Database* (MIT Press, 1998)
55. I. Fernández-Tobías, I. Cantador, Exploiting social tags in matrix factorization models for cross-domain collaborative filtering, in *Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems Co-located with the 8th ACM Conference on Recommender Systems, CBRecSys@RecSys 2014, Foster City, Silicon Valley, California, USA, October 6, 2014*. CEUR Workshop Proceedings, vol. 1245. CEUR-WS.org (2014), pp. 34–41
56. B. Ferwerda, M. Schedl, M. Tkalcic, Personality & emotional states: understanding users' music listening needs, in *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015), Dublin, Ireland, June 29 - July 3, 2015*, ed. by A.I. Cristea, J. Masthoff, A. Said, N. Tintarev. CEUR Workshop Proceedings. CEUR-WS.org, vol. 1388 (2015)
57. J.R. Firth, A synopsis of linguistic theory, 1930–1955. *Studies in linguistic analysis* (Oxford, Blackwell, 1957)
58. G. Friedrich, M. Zanker, A taxonomy for generating explanations in recommender systems. AI Mag. **32**(3), 90–98 (2011)

59. E. Gabrilovich, S. Markovitch, Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge, in *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference* (AAAI Press, Palo Alto, 2006), pp. 1301–1306
60. E. Gabrilovich, S. Markovitch, Computing semantic relatedness using wikipedia-based explicit semantic analysis, in *IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical Intelligence* (2007), pp. 1606–1611
61. E. Gabrilovich, S. Markovitch, Wikipedia-based semantic interpretation for natural language processing. J. Artif. Intell. Res. **34**, 443–498 (2009)
62. G. Ganu, Y. Kakodkar, A. Marian, Improving the quality of predictions using textual information in online user reviews. Inf. Syst. **38**(1), 1–15 (2013)
63. F. Gedikli, D. Jannach, M. Ge, How should I explain? A comparison of different explanation types for recommender systems. Int. J. Hum.-Comput. Stud. **72**(4), 367–382 (2014)
64. B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a right to explanation (2016). arXiv:1606.08813
65. M. Grbovic, V. Radosavljevic, N. Djuric, N. Bhamidipati, J. Savla, V. Bhagwan, D. Sharp, E-commerce in your inbox: product recommendations at scale, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 1809–1818
66. A. Grover, J. Leskovec, node2vec: scalable feature learning for networks, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), pp. 855–864
67. X. Gu, L. Shou, P. Peng, K. Chen, S. Wu, G. Chen, iglasses: a novel recommendation system for best-fit glasses, in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17–21, 2016*, ed. by R. Perego, F. Sebastiani, J.A. Aslam, I. Ruthven, J. Zobel (ACM, New York, 2016), pp. 1109–1112
68. Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* (Early Access). (2020). https://doi.org/10.1109/TKDE.2020.3028705
69. L. Gutiérrez, B. Keith, A systematic literature review on word embeddings, in *International Conference on Software Process Improvement* (Springer, Berlin, 2018), pp. 132–141
70. Z.S. Harris, *Mathematical Structures of Language* (Interscience, New York, 1968)
71. H.A.M. Hassan, G. Sansonetti, F. Gasparetti, A. Micarelli, J. Beel, Bert, elmo, use and infersent sentence encoders: the panacea for research-paper recommendation? in *RecSys (Late-Breaking Results)* (2019), pp. 6–10
72. O. Hassanzadeh, M.P. Consens, Linked movie data base, in *Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009*, ed. by C. Bizer, T. Heath, T. Berners-Lee, K. Idehen. CEUR Workshop Proceedings, vol. 538, CEUR-WS.org (2009)
73. T.H. Haveliwala, Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search. IEEE Trans. Knowl. Data Eng. **15**(4), 784–796 (2003)
74. C. He, D. Parra, K. Verbert, Interactive recommender systems: a survey of the state of the art and future research challenges and opportunities. Exp. Syst. Appl. **56**, 9–27 (2016)
75. X. He, T. Chen, M.-Y. Kan, X. Chen, Trirank: review-aware explainable recommendation by modeling aspects, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (ACM, New York, 2015), pp. 1661–1670
76. J. Henderson, O. Lemon, Mixture model pomdps for efficient handling of uncertainty in dialogue management, in *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15–20, 2008, Columbus, Ohio, USA, Short Papers* (The Association for Computer Linguistics, Stroudsburg, 2008), pp. 73–76
77. L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating Collaborative Filtering Recommender Systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)

78. M. Hernández-Rubio, I. Cantador, A. Bellogín, A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. User Model. User Adapt. Interact. **29**(2), 381–441 (2019)
79. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
80. A. Hotho, R. Jäschke, C. Schmitz, G. Stumme, Information retrieval in folksonomies: search and ranking, in *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11–14, 2006, Proceedings*, ed. by Y. Sure, J. Domingue. Lecture Notes in Computer Science, vol. 4011 (Springer, Berlin, 2006), pp. 411–426
81. A. Iovine, F. Narducci, M. de Gemmis, A dataset of real dialogues for conversational recommender systems, in *CEUR-WS*, *Proceedings of the Sixth Italian Conference on Computational Linguistics*, ed. by R. Bernardi, R. Navigli, G. Semeraro , vol. 2481, (2019). CLiC-it 2019
82. A. Iovine, F. Narducci, G. Semeraro, Conversational recommender systems and natural language: a study through the converse framework. Decis. Support Syst. **131**, 113250 (2020)
83. S. Jain, B.C. Wallace, Attention is not explanation (2019). arXiv:1902.10186
84. D. Jannach, Finding preferred query relaxations in content-based recommenders, in *Intelligent Techniques and Tools for Novel System Architectures* (Springer, Berlin, 2008), pp. 81–97
85. D. Jannach, A. Manzoor, End-to-end learning for conversational recommendation: a long way to go? in *Proceedings of the 7th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems Co-located with 14th ACM Conference on Recommender Systems (RecSys 2020), Online Event, September 26, 2020*, ed. by P. Brusilovsky, M. de Gemmis, A. Felfernig, P. Lops, J. O'Donovan, G. Semeraro, M.C. Willemsen. CEUR Workshop Proceedings, vol. 2682, CEUR-WS.org (2020), pp. 72–76
86. D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, *Recommender Systems: An Introduction* (Cambridge University Press, Cambridge, 2010)
87. D. Jannach, A. Manzoor, W. Cai, L. Chen, A survey on conversational recommender systems (2020). arXiv:2004.00646
88. R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, G. Stumme, Tag recommendations in folksonomies, in *European Conference on Principles of Data Mining and Knowledge Discovery* (Springer, Berlin, 2007), pp. 506–514
89. H. Johnson, P. Johnson, Explanation facilities and interactive systems, in *Proceedings of the 1st International Conference on Intelligent User Interfaces* (ACM, New York, 1993), pp. 159–166
90. D. Khattar, V. Kumar, M. Gupta, V. Varma, Neural content-collaborative filtering for news recommendation, in *CEUR-WS*, Recent Trends in News Information Retrieval, Proceedings of the Second International Workshop on Recent Trends in News Information Retrieval, Co-Located with 40th European Conference on Information Retrieval (ECIR 2018), ed. by D. Albakour, D. Corney, J. Gonzalo, M. Martinez, B. Poblete, A. Valochas , vol. 2079, (2018), pp. 45–50 NewsIR 2018
91. H. Kim, A. Roczniak, P. Lévy, A. El-Saddik, Social media filtering based on collaborative tagging in semantic space. Multim. Tools Appl. **56**(1), 63–89 (2012)
92. P. Knees, M. Schedl, A survey of music similarity and recommendation from music context data. ACM Trans. Multim. Comput. Commun. Appl. **10**(1), 2:1–2:21 (2013)
93. B.P. Knijnenburg, M.C. Willemsen, Z. Gantner, H. Soncu, C. Newell, Explaining the user experience of recommender systems. User Model. User-Adapt. Interact. **22**(4–5), 441–504 (2012)
94. M. Kula, Metadata embeddings for user and item cold-start recommendations (2015). arXiv:1507.08439
95. T.K. Landauer, P.W. Foltz, D. Laham, An introduction to latent semantic analysis. Discourse Process. **25**(2–3), 259–284 (1998)
96. Q. Le, T. Mikolov, Distributed representations of sentences and documents, in *International Conference on Machine Learning* (2014), pp. 1188–1196
97. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**(7553), 436–444 (2015)

98. H. Li, S. Sanner, K. Luo, G. Wu, A ranking optimization approach to latent linear critiquing for conversational recommender systems, in *Fourteenth ACM Conference on Recommender Systems* (2020), pp. 13–22
99. P. Li, A. Tuzhilin, Learning latent multi-criteria ratings from user reviews for recommendations. *IEEE Transactions on Knowledge and Data Engineering* (Early Access). (2020). https://doi.org/10.1109/TKDE.2020.3030623
100. R. Li, S.E. Kahou, H. Schulz, V. Michalski, L. Charlin, C. Pal, Towards deep conversational recommendations, in *Advances in Neural Information Processing Systems* (2018), pp. 9725–9735
101. B. Liu, I. Lane, An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *Interspeech 2017* (2017), pp. 2506–2510. arXiv: 1708.05956
102. H. Liu, J. He, T. Wang, W. Song, X. Du, Combining user preferences and user opinions for accurate recommendation. Electron. Commer. Res. Appl. **12**(1), 14–23 (2013)
103. D. Liu, J. Lian, S. Wang, Y. Qiao, J.-H. Chen, G. Sun, X. Xie, KRED: knowledge-aware document representation for news recommendations (2019). arXiv:1910.11494
104. P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Springer, Berlin, 2011), pp. 73–105
105. P. Lops, C. Musto, F. Narducci, G. Semeraro, *Semantics in Adaptive and Personalised Systems - Methods, Tools and Applications* (Springer, Berlin, 2019)
106. W. Lowe, Towards a theory of semantic space, in *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society* (Lawrence Erlbaum Associates, 2001), pp. 576–581
107. V. Lully, P. Laublet, M. Stankovic, F. Radulovic, Enhancing explanations in recommender systems with knowledge graphs. Proc. Comput. Sci. **137**, 211–222 (2018)
108. X. Luo, Y. Ouyang, Z. Xiong, Improving neighborhood based collaborative filtering via integrated folksonomy information. Pattern Recognit. Lett. **33**(3), 263–270 (2012)
109. T. Mahmood, F. Ricci, Improving recommender systems with adaptive conversational strategies, in *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia* (ACM, New York, 2009), pp. 73–82
110. C.D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*, vol. 999 (MIT Press, Cambridge, 1999)
111. J. Manotumruksa, C. Macdonald, I. Ounis, Modelling user preferences using word embeddings for context-aware venue recommendation (2016). arXiv:1606.07828
112. J.J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in *Seventh ACM Conference on Recommender Systems, RecSys'13, Hong Kong, China, October 12–16, 2013*, ed. by Q. Yang, I. King, Q. Li, P. Pu, G. Karypis (ACM, New York, 2013), pp. 165–172
113. J.J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9–13, 2015*, ed. by R. Baeza-Yates, M. Lalmas, A. Moffat, B.A. Ribeiro-Neto (ACM, New York, 2015), pp. 43–52
114. F. McCarey, M.Ó. Cinnéide, N. Kushmerick, Recommending library methods: an evaluation of the vector space model (VSM) and latent semantic indexing (LSI), in *International Conference on Software Reuse* (Springer, Berlin, 2006), pp. 217–230
115. D. McSherry, Explanation in recommender systems. Artif. Intell. Rev. **24**(2), 179–197 (2005)
116. M.M. Mehrabani, H. Mohayeji, A. Moeini, A hybrid approach to enhance pure collaborative filtering based on content feature relationship (2020). arXiv:2005.08148
117. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in *Advances in Neural Information Processing Systems* (2013), pp. 3111–3119

118. T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2013), pp. 746–751
119. G. Miller, WordNet: An on-line lexical database. Int. J. Lexicography. **3**(4), 235–244 (1990)
120. G. Miller, Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
121. T.M. Mitchell, *Machine Learning*. McGraw Hill Series in Computer Science (McGraw-Hill, New York, 1997)
122. S. Mohammad, G. Hirst, Distributional measures of semantic distance: a survey (2012). CoRR, abs/1203.1858
123. K.I. Muhammad, A. Lawlor, B. Smyth, A live-user study of opinionated explanations for recommender systems, in *Proceedings of the 21st International Conference on Intelligent User Interfaces* (ACM, New York, 2016), pp. 256–260
124. C. Musto, Enhanced vector space models for content-based recommender systems, in *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26–30, 2010*, ed. by X. Amatriain, M. Torrens, P. Resnick, M. Zanker (ACM, New York, 2010), pp. 361–364
125. C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Random indexing and negative user preferences for enhancing content-based recommender systems, in *E-Commerce and Web Technologies - 12th International Conference, EC-Web 2011, Toulouse, France, August 30 - September 1, 2011. Proceedings*, ed. by C. Huemer, T. Setzer. Lecture Notes in Business Information Processing, vol. 85 (Springer, 2011), pp. 270–281
126. C. Musto, G. Semeraro, P. Lops, M. de Gemmis, F. Narducci, Leveraging social media sources to generate personalized music playlists, in *E-Commerce and Web Technologies - 13th International Conference, EC-Web 2012*. Lecture Notes in Business Information Processing, vol. 123 (Springer, Berlin, 2012), pp. 112–123
127. C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Contextual eVSM: a content-based context-aware recommendation framework based on distributional semantics, in *E-Commerce and Web Technologies - 14th International Conference, EC-Web 2013, Prague, Czech Republic, August 27–28, 2013. Proceedings*, ed. by C. Huemer, P. Lops. Lecture Notes in Business Information Processing (Springer, Berlin, 2013), pp. 125–136
128. C. Musto, P. Basile, P. Lops, M. de Gemmis, G. Semeraro, Linked open data-enabled strategies for top-n recommendations, in *Proceedings of the 1st Workshop on New Trends in Content-Based Recommender Systems Co-located with the 8th ACM Conference on Recommender Systems, CBRecSys@RecSys 2014, Foster City, Silicon Valley, California, USA, October 6, 2014.*, ed. by T. Bogers, M. Koolen, I. Cantador. CEUR Workshop Proceedings, vol. 1245. CEUR-WS.org (2014), pp. 49–56
129. C. Musto, G. Semeraro, M. de Gemmis, P. Lops, Learning word embeddings from wikipedia for content-based recommender systems, in *European Conference on Information Retrieval* (Springer, Berlin, 2016), pp. 729–734
130. C. Musto, G. Semeraro, M. de Gemmis, P. Lops, Learning word embeddings from wikipedia for content-based recommender systems, in *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings*, ed. by N. Ferro, F. Crestani, M. Moens, J. Mothe, F. Silvestri, G.M.D. Nunzio, C. Hauff, G. Silvello. Lecture Notes in Computer Science, vol. 9626 (Springer, Berlin, 2016), pp. 729–734
131. C. Musto, P. Basile, P. Lops, M. de Gemmis, G. Semeraro, Introducing linked open data in graph-based recommender systems. Inf. Process. Manag. **53**(2), 405–435 (2017)
132. C. Musto, M. de Gemmis, G. Semeraro, P. Lops, A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews, in *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27–31, 2017*, ed. by P. Cremonesi, F. Ricci, S. Berkovsky, A. Tuzhilin (ACM, New York, 2017), pp. 321–325

133. C. Musto, P. Lops, M. de Gemmis, G. Semeraro, Semantics-aware recommender systems exploiting linked open data and graph-based features. Knowl.-Based Syst. **136**, 1–14 (2017)
134. C. Musto, G. Semeraro, M. de Gemmis, P. Lops, Tuning personalized pagerank for semantics-aware recommendations based on linked open data, in *The Semantic Web - 14th International Conference, ESWC 2017*, ed. by E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, O. Hartig. Lecture Notes in Computer Science, vol. 10249 (2017), pp. 169–183
135. C. Musto, T. Franza, G. Semeraro, M. de Gemmis, P. Lops, Deep content-based recommender systems exploiting recurrent neural networks and linked open data, in *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization* (2018), pp. 239–244
136. C. Musto, P. Basile, G. Semeraro, Hybrid semantics-aware recommendations exploiting knowledge graph embeddings, in *International Conference of the Italian Association for Artificial Intelligence* (Springer, Berlin, 2019), pp. 87–100
137. C. Musto, P. Lops, M. de Gemmis, G. Semeraro, Justifying recommendations through aspect-based sentiment analysis of users reviews, in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization* (2019), pp. 4–12
138. C. Musto, F. Narducci, P. Lops, M. de Gemmis, G. Semeraro, Linked open data-based explanations for transparent recommender systems. Int. J. Hum.-Comput. Stud. **121**, 93–107 (2019)
139. C. Musto, G. Rossiello, M. de Gemmis, P. Lops, G. Semeraro, Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations, in *Proceedings of the 13th ACM Conference on Recommender Systems* (2019), pp. 383–387
140. C. Musto, G. Spillo, M. de Gemmis, P. Lops, G. Semeraro, Exploiting distributional semantics models for natural language context-aware justifications for recommender systems, in *Proceedings of the 7th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems Co-located with 14th ACM Conference on Recommender Systems (RecSys 2020)* (2020), pp. 65–71
141. C. Musto, M. de Gemmis, P. Lops, G. Semeraro, Generating post hoc review-based natural language justifications for recommender systems. User Model. User-Adapt. Interact. **31**, 629–673 (2021)
142. R.Y. Nakamoto, S. Nakajima, J. Miyazaki, S. Uemura, H. Kato, Y. Inagaki, Reasonable tag-based collaborative filtering for social tagging systems, in *Proceedings of the 2nd ACM Workshop on Information Credibility on the Web, WICOW 2008, Napa Valley, California, USA, October 30, 2008*, ed. by K. Tanaka, T. Matsuyama, E. Lim, A. Jatowt (ACM, New York, 2008), pp. 11–18
143. F. Narducci, C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Exploiting big data for enhanced representations in content-based recommender systems, in *International Conference on Electronic Commerce and Web Technologies* (Springer, Berlin, 2013), pp. 182–193
144. F. Narducci, P. Basile, M. de Gemmis, P. Lops, G. Semeraro, An investigation on the user interaction modes of conversational recommender systems for the music domain. User Model. User-Adapt. Interact. **30**(2), 251–284 (2020)
145. R. Navigli, S.P. Ponzetto, Babelnet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (2012)
146. L. Nie, W. Wang, R. Hong, M. Wang, Q. Tian, Multimodal dialog system: generating responses via adaptive decoders, in *Proceedings of the 27th ACM International Conference on Multimedia* (2019), pp. 1098–1106
147. M.G. Ozsoy, From word embeddings to item recommendation (2016). arXiv:1601.01356
148. L. Page, S. Brin, R. Motwani, T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web.* (Stanford InfoLab, Stanford, 1999)
149. A. Passant, dbrec - music recommendations using dbpedia, in *International Semantic Web Conference, Revised Papers*. Lecture Notes in Computer Science, vol. 6497 (Springer, Berlin, 2010), pp. 209–224
150. A. Passant, Measuring semantic distance on linking data and using it for resources recommendations, in *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence* (AAAI, Menlo Park, 2010), pp. 93–98

151. G. Peake, J. Wang, Explanation mining: post hoc interpretability of latent factor models for recommendation systems, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), pp. 2060–2069
152. M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations (2018). arXiv:1802.05365
153. G. Qiu, B. Liu, J. Bu, C. Chen, Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)
154. M. Quadrana, P. Cremonesi, D. Jannach, Sequence-aware recommender systems. ACM Comput. Surv. **51**(4), 66:1–66:36 (2018)
155. D. Rao, P. McNamee, M. Dredze, Entity linking: finding extracted entities in a knowledge base, in *Multi-Source, Multilingual Information Extraction and Summarization* (Springer, Berlin, 2013), pp. 93–115
156. H. Ren, W. Xu, Y. Yan, Markovian discriminative modeling for cross-domain dialog state tracking, in *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7–10, 2014* (IEEE, Piscataway, 2014), pp. 342–347
157. F. Ricci, Q.N. Nguyen, Acquiring and revising preferences in a critique-based mobile recommender system. IEEE Intell. Syst. **22**(3), 22–29 (2007)
158. F. Ricci, A. Venturini, D. Cavada, N. Mirzadeh, D. Blaas, M. Nones, Product recommendation with interactive query management and twofold similarity, in *International Conference on Case-Based Reasoning* (Springer, Berlin, 2003), pp. 479–493
159. A. Ritter, C. Cherry, W.B. Dolan, Data-driven response generation in social media, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP'11* (Association for Computational Linguistics, Stroudsburg, 2011), pp. 583—593
160. S. Robertson, H. Zaragoza, *The Probabilistic Relevance Framework: BM25 and Beyond* (Now Publishers, Norwell, 2009)
161. M. Rossetti, F. Stella, M. Zanker, Towards explaining latent factors with topic models in collaborative recommender systems, in *Database and Expert Systems Applications (DEXA), 2013 24th International Workshop on* (IEEE, Piscataway, 2013), pp. 162–167
162. H. Rubenstein, J.B. Goodenough, Contextual correlates of synonymy. Commun. ACM **8**(10), 627–633 (1965)
163. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
164. M. Sahlgren, The word-space model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. Ph.D. Thesis, Stockholm University (2006)
165. G. Salton, M. McGill, *Introduction to Modern Information Retrieval* (McGraw-Hill, New York, 1983)
166. G. Salton, A. Wong, C. Yang, A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
167. C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, C. Jin, Red opal: product-feature scoring from reviews, in *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11–15, 2007*, ed. by J.K. MacKie-Mason, D.C. Parkes, P. Resnick (ACM, New York, 2007), pp. 182–191
168. M. Schedl, P. Knees, B. McFee, D. Bogdanov, M. Kaminskas, Music recommender systems, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira (Springer, Berlin, 2015), pp. 453–492
169. H. Schütze, Automatic word sense discrimination. Comput. Linguistics **24**(1), 97–123 (1998)
170. G. Semeraro, M. Degemmis, P. Lops, P. Basile, Combining learning and word sense disambiguation for intelligent user profiling, in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6–12, 2007*, ed. by M.M. Veloso (2007), pp. 2856–2861
171. S. Seo, J. Huang, H. Yang, Y. Liu, Interpretable convolutional neural networks with dual local and global attention for review rating prediction, in *Proceedings of the Eleventh ACM Conference on Recommender Systems* (2017), pp. 297–305

172. W. Shen, J. Wang, J. Han, Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Trans. Knowl. Data Eng. **27**(2), 443–460 (2014)
173. D. Shin, S. Cetintas, K.-C. Lee, Recommending tumblr blogs to follow with inductive matrix completion, in *RecSys Posters* (2014)
174. D. Shmaryahu, I. Negev, G. Shani, B. Shapira, Post-hoc explanations for complex model recommendations using simple methods, in *(RecSys 2020), CEUR-WS*, Interfaces and Human Decision Making for Recommender Systems 2020, Proceedings of the 7th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, Co-Located with 14th ACM Conference on Recommender Systems, ed. by P. Brusilovsky, M. de Gemmis, A. Felfernig, P. Lops, J. O'Donovan, G. Semeraro, M. C. Willemsen , vol. 2682, (2020), pp. 26–36 IntRS 2020
175. R. Sinha, K. Swearingen, The role of transparency in recommender systems, in *CHI'02 Extended abstracts on Human Factors in Computing Systems* (ACM, New York, 2002), pp. 830–831
176. B. Smyth, L. McGinty, An analysis of feedback strategies in conversational recommenders, in *The Fourteenth Irish Artificial Intelligence and Cognitive Science Conference (AICS 2003)* (Citeseer, 2003)
177. P. Stakhiyevich, Z. Huang, Building user profiles based on user interests and preferences for recommender systems, in *2019 IEEE International Conferences on Ubiquitous Computing Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)* (2019), pp. 450–455
178. M. Stevenson, *Word Sense Disambiguation: The Case for Combinations of Knowledge Sources* (CSLI Publications, Stanford, 2003)
179. S. Stiebellehner, J. Wang, S. Yuan, Learning continuous user representations through hybrid filtering with doc2vec (2017). arXiv:1801.00215
180. A. Suglia, C. Greco, C. Musto, M. de Gemmis, P. Lops, G. Semeraro, A deep architecture for content-based recommendations exploiting recurrent neural networks, in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP 2017, Bratislava, Slovakia, July 09–12, 2017*, ed. by M. Bieliková, E. Herder, F. Cena, M.C. Desmarais (ACM, New York, 2017), pp. 202–211
181. C. Sundermann, J. Antunes, M. Domingues, S. Rezende, Exploration of word embedding model to improve context-aware recommender systems, in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (IEEE, Piscataway, 2018), pp. 383–388
182. P. Symeonidis, A. Nanopoulos, Y. Manolopoulos, MoviExplain: a recommender system with explanations, in *Proceedings of the Third ACM Conference on Recommender Systems* (ACM, New York, 2009), pp. 317–320
183. C.A. Thompson, M.H. Goker, P. Langley, A personalized system for conversational recommendations. J. Artif. Intell. Res. **21**, 393–428 (2004)
184. I. Titov, R.T. McDonald, A joint model of text and aspect ratings for sentiment summarization, in *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15–20, 2008, Columbus, Ohio, USA*, ed. by K.R. McKeown, J.D. Moore, S. Teufel, J. Allan, S. Furui (The Association for Computer Linguistics, Stroudsburg, 2008), pp. 308–316
185. I. Titov, R.T. McDonald, Modeling online reviews with multi-grain topic models, in *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21–25, 2008*, ed. by J. Huai, R. Chen, H. Hon, Y. Liu, W. Ma, A. Tomkins, X. Zhang (ACM, New York, 2008), pp. 111–120
186. W. Trabelsi, N. Wilson, D. Bridge, F. Ricci, Comparing approaches to preference dominance for conversational recommenders, in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 2 (IEEE, Piscataway, 2010), pp. 113–120.
187. K.H.L. Tso-Sutter, L.B. Marinho, L. Schmidt-Thieme, Tag-aware recommender systems by fusion of collaborative filtering algorithms, in *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16–20, 2008*, ed. by R.L. Wainwright, H. Haddad (ACM, New York, 2008), pp. 1995–1999

188. P.D. Turney, P. Pantel, From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**(1), 141–188 (2010)
189. A. van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in *Advances in Neural Information Processing Systems*, ed. by C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger, vol. 26 (Curran Associates, Red Hook, 2013), pp. 2643–2651
190. F. Vasile, E. Smirnova, A. Conneau, Meta-prod2vec: product embeddings using side-information for recommendation, in *Proceedings of the 10th ACM Conference on Recommender Systems* (2016), pp. 225–232
191. J. Vig, S. Sen, J. Riedl, Tagsplanations: explaining recommendations using tags, in *Proceedings of the 14th International Conference on Intelligent User Interfaces* (ACM, New York, 2009), pp. 47–56
192. D. Vrandecic, M. Krötzsch, Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
193. Z. Wang, O. Lemon, A simple and generic belief tracking mechanism for the dialog state tracking challenge: on the believability of observed information, in *Proceedings of the SIGDIAL 2013 Conference, The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 22–24 August 2013, SUPELEC, Metz, France* (The Association for Computer Linguistics, Stroudsburg, 2013), pp. 423–432
194. H. Wang, F. Zhang, X. Xie, M. Guo, DKN: deep knowledge-aware network for news recommendation, in *Proceedings of the 2018 World Wide Web Conference* (2018), pp. 1835–1844
195. S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, H. Liu, What your images reveal: exploiting visual contents for point-of-interest recommendation, in *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3–7, 2017*, ed. by R. Barrett, R. Cummings, E. Agichtein, E. Gabrilovich (ACM, New York, 2017), pp. 391–400
196. H. Wang, M. Zhao, X. Xie, W. Li, M. Guo, Knowledge graph convolutional networks for recommender systems, in *The World Wide Web Conference* (2019), pp. 3307–3313
197. X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, KGAT: knowledge graph attention network for recommendation, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 950–958
198. P. Wärnestål, User evaluation of a conversational recommender system, in *Proceedings of the 4th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems* (2005), pp. 32–39
199. R. Wetzker, C. Zimmermann, C. Bauckhage, S. Albayrak, I tag, you tag: translating tags for advanced user models, in *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4–6, 2010*, ed. by B.D. Davison, T. Suel, N. Craswell, B. Liu (ACM, New York, 2010), pp. 71–80
200. J.D. Williams, Challenges and opportunities for state tracking in statistical spoken dialog systems: results from two public deployments. IEEE J. Sel. Top. Signal Process. **6**(8), 959–970 (2012)
201. J.D. Williams, S.J. Young, Partially observable markov decision processes for spoken dialog systems. Comput. Speech Lang. **21**(2), 393–422 (2007)
202. J. Williams, A. Raux, M. Henderson, The dialog state tracking challenge series: a review. Dial. Discourse **7**(3), 4–33 (2016)
203. J.D. Williams, A. Raux, M. Henderson, The dialog state tracking challenge series: a review. Dial. Discourse **7**(3), 4–33 (2016)
204. L. Wittgenstein, *Philosophical Investigations* (Blackwell, Oxford, 1953)
205. C. Wu, F. Wu, M. An, J. Huang, Y. Huang, X. Xie, NPA: neural news recommendation with personalized attention, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 2576–2584
206. S. Xu, Y. Li, S. Liu, Z. Fu, Y. Zhang, Learning post-hoc causal explanations for recommendation (2020). arXiv:2006.16977

207. Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval* (2014), pp. 83–92
208. F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), pp. 353–362
209. S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review. Comput. Soc. Netw. **6**(1), 1–23 (2019)
210. T. Zhao, M. Eskenazi, Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning (2016). arXiv:1606.02560 [cs]
211. L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (2017), pp. 425–434