Semantics-based Recommender Systems

Fatih Gedikli[0000−0001−6190−0449] and
Dietmar Jannach[0000−0002−4698−8507]

Abstract

This chapter provides a short overview on semantics-based recommender systems. These are systems that generate personalized suggestions in a way that they match past user preferences regarding *semantic* item properties.

1 Synonyms

Content-based Recommender Systems; Recommendation with Side Information

2 Definition

Semantics-based Recommender Systems generate personalized suggestions in a way that they match past user preferences regarding *semantic* item properties. The term 'semantic', as defined by the Oxford English Dictionary, relates to the study of *meaning* in language. Essentially, this chapter is about teaching recommender systems to *understand* information about the recommendation domain provided in natural language, employing techniques from Natural Language Processing (NLP).

Fatih Gedikli
University of Applied Sciences Ruhr West, Germany, e-mail: fatih.gedikli@hs-ruhrwest.de

Dietmar Jannach
University of Klagenfurt, Austria, e-mail: dietmar.jannach@aau.at

Preprint, to be published in Encyclopedia of Social Network Analysis and Mining, 3rd Edition, Springer, forthcoming, 2026

3 Introduction

Personalized recommendations have become a ubiquitous element of our online user experience. Today, recommender systems (RS) are successfully applied in various application settings, including e-commerce, media streaming, news, tourism, or social media [Ricci et al., 2022], where they can create substantial value both for consumers and service providers [Jannach and Zanker, 2021].

A multitude of technical approaches to implement recommender systems has been proposed over the last three decades. Commonly, these approaches are categorized in terms of the types of information they rely on to generate personalized item suggestions [Jannach et al., 2011]. *Collaborative filtering* (CF) techniques operate on the basis of patterns in the collective behavior or preference profiles of a larger group of users. These techniques process larger datasets containing user behavior logs or explicit consumer ratings to predict the relevance of individual items for a given user. *Content-based filtering* (CB) approaches, in contrast, leverage knowledge about the preferences of individual users regarding certain items, and they recommend items that are similar to those that a given consumer liked in the past. *Knowledge-based techniques* are used in niche application use cases of recommender systems, and they commonly implement recommendation rules that are manually designed by domain experts. *Hybrid approaches*, finally, combine two or more approaches, often by considering more than one type of information in parallel, e.g., both collaborative signals as well as meta-data about the items that can be recommended.

Collaborative filtering is the most widely studied type of approach in the literature, and it is also in widespread use in practical settings. While such techniques are often unmatched in making accurate relevance predictions in particular in offline experiments, content-based and hybrid approaches have their place as well. Pure content-based techniques may for example be favorable in practical settings in which user preferences are narrow and stable over time. In [Jannach and Hegelich, 2009], for example, an A/B test in the mobile games domain revealed that a content-based approach was the most effective one in terms of Key Performance Indicators such as conversion rates and purchases. In other domains, approaches that combine collaborative filtering and content-based filtering turn out to be highly effective. A very common pattern in such hybrids is to augment the collaborative filtering process with what is called *side information* of different types, see [Sun et al., 2019] for a related survey. Such combined approaches work particularly well for certain domains where (a) the features (or content) of the items matter, e.g., in the news domain [Kirshenbaum et al., 2012], or (b) when there is not enough data to reliably apply a pure collaborative filtering technique, e.g., in cold-start settings where frequently new items are added to the catalog.

Historically, as we discuss also in Section 5, content-based filtering has its roots in applications where the items are text documents, like news articles or electronic messages. In such applications, the actual content of the documents, i.e., the natural language text, was considered as the information that characterizes the item. Later on, also various forms of meta-data (e.g., the genre of a movie) where subsumed under the term *content* for these historical reasons.

In those cases where the actual content of the documents was considered, the probably most common way of transforming the document for further processing was to use a Term-Frequency Inverse Document Frequency (TF-IDF) encoding. This encoding determines the importance of individual terms of the document in terms of their appearance frequency, with a discounting factor for generally frequent terms. While such an encoding has shown to be effective in many use cases, a central limitation of such an approach is that the *semantics* of the terms are not considered. For example, in such an encoding synonyms are not taken into account. This may results in the effect that documents that are factually similar in their content, are actually considered not very similar, just because different words are used.

Given these limitations, a variety of approaches were put forward in recent years that target at taking the semantics of the document terms or the item properties into account in the recommendation process. These important developments are spurred by different technological advances. Among other aspects, we in particular observe a continuing increase of publicly available data sources that encode 'world knowledge' in a machine-processible form. Furthermore, advances in machine learning enable novel forms of elaborate semantically-rich document representations (*embeddings*), most recently in the form of Large Language Models (LLMs). We discuss the main classes of approaches in Section 6.

4 Key Points

- Semantics-aware recommendation approaches can help to overcome key limitations of collaborative filtering techniques.
- Semantics can be extracted from both external sources (*exogenous knowledge*) and directly from the text documents (*endogenous knowledge*).
- Research in semantics-aware recommender systems is fueled by disruptive advances in the area of natural language processing (including LLMs) and the increasing availability of ontological databases containing 'world knowledge'.

5 Historical Background

Content-based filtering methods have strong roots in the fields of Information Retrieval (IR) and Information Filtering [Belkin and Croft, 1992]. Various technical approaches developed in the IR field, for example, in terms of document representations, were later used in content-based recommender systems. A main difference between an IR and a recommendation setting is that in typical IR settings, the retrieval of documents is based on explicit query terms that are provided by a proactive user. In an RS setting, in contrast, the starting point is an individual user profile maintained by the system, and the proactive recommendation of items is guided by their match with the user profile.

From a historical perspective, traditional technical approaches in classical content-based recommendation methods include TF-IDF for document representation, decision trees and nearest-neighbor methods for classification and ranking, or the use of relevance feedback and Rocchio's Algorithm as a learning approach, see [Pazzani and Billsus, 2007]. Later on, with the rise of the 'Semantic Web' in the 2000s, various works started to leverage structured knowledge sources like DBPedia[1] or databases that implement Linked (Open) Data[2] principles. Around the same time, the World Wide Web started to develop into a more interactive, participatory Web (Web 2.0), where user-generated content became available in abundant amounts. Correspondingly, various approaches were proposed that try to extract valuable side information from this sources, e.g., in the form of user-provided tags or reviews for the recommendable items. The most recent years in semantics-aware approaches are particularly shaped by the rise of deep learning methods. These developments led to the emergence of more semantically-rich document representation or the consideration of various types of side information in graph-based models. Most recently, with the development of continuously more powerful LLMs, which are also able to process multi-modal content, new opportunities arise to build the next generation of semantics-aware recommender systems.

6 Semantics-based Recommender Systems: Key Technical Approaches

In Semantics-aware Recommender Systems, the essential strategy involves leveraging descriptive attributes of items to create customized recommendations. This approach relies on two key methods for deriving content[3] features from items: (1) *endogenous*, which focuses on internal data, and (2) *exogenous*, involving the use of external data sources [Musto et al., 2022].

1. **Endogenous Feature Extraction**: This technique is introspective, focusing on leveraging the inherent data associated with an item and the user's profile to construct an understanding of the recommendable items and the target user. It involves a deep analysis of the item's intrinsic properties, such as text, images, audio, or video content. Endogenous methods are thus agnostic to data that is available outside, such as on the web.
2. **Exogenous Feature Extraction**: In contrast, this approach extends beyond the bounds of a given dataset of users and items, embracing external data sources such as Open Knowledge Repositories, Knowledge Graphs, and various forms of metadata. Exogenous feature extraction is akin to looking outward for complementary insights that enrich the recommendation process. It might include

[1] https://www.dbpedia.org/

[2] https://en.wikipedia.org/wiki/Linked_data

[3] We use the term 'content' here in the traditional, generic sense as in the literature.

leveraging data from social media, user-generated content, expert opinions, or global databases to provide a more rounded and informed recommendation.

In the following sections we provide a more detailed discussion of each strategy, showing their roles, advantages, and potential challenges in the domain of personalized content recommendation. In addition, we will illustrate these strategies with examples drawn from existing literature.

6.1 Word Embedding Techniques

One of the most important inventions that have helped modern-day LLMs become successful are *word embeddings*. The extraction of word embeddings from large text corpora represents a central endogenous feature extraction method. The basic idea is to map all the words of a vocabulary to numbers such that semantically similar words have similar numbers. Instead of mapping each word to a single number, words are mapped to vectors of real numbers in a high-dimensional space. Such vectors, which are referred to as embeddings, are learned from the text data itself, capturing syntactic and semantic word relationships based on the words' context in the text. The most common methods to generate word embeddings are algorithms like *Word2Vec*, *GloVe*, and *fastText*.

- **Word2Vec** (developed by a team at Google) generates word embeddings by using neural network models, either through a Continuous Bag-of-Words (CBOW) model or a Skip-Gram model (SG) [Mikolov et al., 2013].
- **GloVe** (Global Vectors for Word Representation, developed by Stanford researchers) is an unsupervised learning algorithm for generating word embeddings by aggregating global word-word co-occurrence statistics from a corpus [Pennington et al., 2014].
- **fastText**, developed by Meta, extends Word2Vec to consider subword information (like n-grams), making it better at handling rare words [Bojanowski et al., 2017].

Generally, the core idea of these approaches is to learn the *meaning* of a word through the neighboring words, which is called the context of the word. Of course, this observation was also made long ago in linguistics. John Rupert Firth, a prominent linguist from the 1950s, famously stated, "You shall know a word by the company it keeps" [Firth, 1957], underlining the significance of *context* in language analysis. Since the basic ideas of all three methods mentioned above are similar, we will discuss Word2Vec in more detail here.

Word2Vec utilizes a two-layer neural network architecture to develop representations of words in a high-dimensional vector space. The architecture features an initial input layer, where each node corresponds to a unique word (or phrase) in the vocabulary V. Therefore, the vocabulary size $|V|$ corresponds to the number of nodes in the input layer. Following this, the second and final output layer addresses the task of building word vectors. The size of these vectors, N, is a user-defined parameter that determines the dimensionality of each word representation. In general, a larger value for N enhances the depth of the word representation, enabling it to encapsulate more subtle linguistic nuances such as synonymy or polysemy. However, a larger dimensionality also leads to a more complex and expensive training process. Typical values for N are 50, 100, or 300. Note that if N is too large, performance typically decreases after a while. One reason might be that noise in the data is increasingly taken into account in the learning process when the vectors are too large.

Two different methods can be applied to train the network. The Continuous Bag-of-Words (CBOW) architecture tries to predict the current word based on the words in the context, i.e., the words before and after the current word. Let us assume that the training corpus contains the sentence "`The cat is on the mat.`". In this example CBOW would try to predict the target word *is* given the surrounding words: "`The cat ? on the mat.`". The Skip-Gram (SG) architecture, on the other hand, tries to predict the surrounding words given the current word: "`? ? is ? ? ?.`". This prediction task forces the model to learn representations of words that capture their meanings and relationships with other words in the language. For example, in CBOW, if the model frequently sees 'cat' and 'mat' around 'is', it starts to learn that these words have a contextual relationship. In SG, given 'is', the model tries to predict 'cat', 'mat', and other surrounding words, thereby understanding the word's usage in various contexts.

Each architecture has its strengths and weaknesses, and their performance can vary based on several factors. It can be observed that SG produces more training examples compared to CBOW. Therefore, CBOW is in general faster to train but less accurate than SG, since SG can produce more training examples for rare words.

6.1.1 Extending Word Embedding Concepts: From Words to Items, Products, and Documents

The principles underlying Word2Vec have proven to be widely applicable and extend far beyond processing individual words. This versatility has given rise to various adaptations of the original model, each tailored to different types of data. Notable among these are Item2Vec [Barkan and Koenigstein, 2016], Prod2Vec [Vasile et al., 2016], Paragraph2Vec, and Doc2Vec [Le and Mikolov, 2014]. Item2Vec and Prod2Vec apply the Word2Vec methodology to the realm of recommendation systems, where items or products are represented as vectors, facilitating the discovery of similarities and patterns in user preferences and behaviors. Similarly, Paragraph2Vec and Doc2Vec, also known as sentence embeddings and document embeddings respectively, extend the concept to larger blocks of text. These models encapsulate the semantic essence of paragraphs and entire documents, enabling a nuanced understanding and comparison of longer text forms. This generalization of the Word2Vec concept across multiple levels underscores its foundational role in the field of semantic analysis and its broad applicability in various domains of AI and machine learning.

6.2 Open Knowledge Sources and Knowledge Graphs

In the following, we will discus different types of data that are used by Exogenous Feature Extraction techniques in the literature.

6.2.1 Social Web

The Social Web allows users to create and share a large amount of different types of content such as pictures, videos, bookmarks, blogs, comments, or tagging data. It enables users to collaborate with other users on new types of Web applications called Social Web platforms. Examples of such platforms include X (Twitter), a microblogging service where users share short messages and media; Instagram, a photo and video sharing application; or Reddit, a network of communities based on people's interests[4]. Extracting useful data from the large amount of user-contributed data available in the Social Web represents a challenging topic which however also opens new opportunities for recommender system research. It is worth noting that NLP techniques can also process non-textual content like images, music, or videos, by utilizing their automatically-transcribed versions. In this context, Deep Learning models have shown to be particularly effective at tasks such as generating titles for images or creating automated subtitles from a video's audio track, which can then be utilized for NLP-based tasks in recommender systems.

One important source of information that is frequently leveraged in the literature are user-contributed *tags*. Tags are a popular means for users to organize and retrieve items of interest in the Social Web. As the application areas of tags are manifold, they play an important role for recommender systems. They can be used to categorize items, express preferences about items, retrieve items of interest, predict users' preference, and so on.

Item-specific Tag Preferences. In [Vig et al., 2010] and [Gedikli and Jannach, 2013], the concept of item-specific tag preference data was introduced. The intuition behind this idea is that the same tag may have a positive connotation for the user in one context and a negative in another. For example, a user might like action movies featuring the actor Bruce Willis, but at the same time, the user might dislike the performance of Bruce Willis in romantic movies. Based on such an approach, users are able to evaluate an item in various dimensions and are thus not limited to the one single overall vote anymore. According to the study presented in [Vig et al., 2010], users particularly appreciated this new feature, which was measured in increased user satisfaction. In [Gedikli and Jannach, 2013], the authors present different novel recommendation schemes to take item-specific tag preferences into account when generating rating predictions. The results show that the accuracy can be further improved by exploiting item-specific tag preference data. In a related work, tags can also be used to predict the target user's preferences. In [Nguyen and Riedl, 2013], for

[4] https://x.com, https://www.instagram.com, https://www.reddit.com

example, the authors present two recommender algorithms, that incorporate tagging data to better assess the users' preferences.

Tag-based Explanations. Tag information may not only enhance recommender algorithms but may also serve as a basis for better *explanations*, a key research area in recommender systems. Vig et al. [2009] introduced "tagsplanations", using tag relevance (tag-item relationship) and tag preference (user-tag relationship) for explanations. For example, 'love' as a tag might indicate a movie's theme and a user's interest in such movies. Gedikli et al. [2014] developed explanation interfaces using personalized and non-personalized tag clouds, contrasting them with previous keyword-style explanations. Personalized tag clouds, informed by user preferences, use color-coding to indicate a user's likely opinion on the item's features represented by tags.

Overall, as can be seen, there are many possible applications for social web (tagging) data. Shoja and Tabrizi [2019] offer an overview of of tag-aware recommender systems. This article systematically reviews various aspects of these systems and sketches future research directions and opportunities.

6.2.2 Semantic Web and Knowledge Graphs

The advent of the Semantic Web (Web 3.0) had a significant impact on the ways in which user-generated content is promoted and shared. Linked data, a cornerstone of the Web 3.0 enables a more interconnected and semantic web experience. The idea is to transform the traditional *web of documents* into a *web of data*, where machines can understand the relationship between entities such as: *The entity 'cat' is an instance of the entity 'animal'*. The Semantic Web Technology Stack provides a vast array of tools and protocols designed to enhance the accessibility and interoperability of data across different systems. This includes technologies such as RDF (Resource Description Framework), OWL (Web Ontology Language), and SPARQL (SPARQL Protocol and RDF Query Language), which collectively establish a framework for integrating, sharing, and querying data in a semantically rich and meaningful way. Although the complexity of these technologies poses challenges for widespread adoption, their potential is significant, particularly when combined with knowledge graphs.

A *knowledge graph* is a graph which consists of real-world entities, their attributes, and the links between them. Prominent examples of such knowledge graphs are DBpedia[5] and Wikidata[6], which offer extensive, interlinked datasets. These graphs enable a deep semantic interpretation that can significantly enhance the accuracy and relevance of recommender systems, offering both a broader and more contextually rich data analysis. The Linked Open Vocabularies (LOV) project provides a comprehensive overview of the available Linked Open Data on the web.[7]

[5] https://www.dbpedia.org

[6] https://www.wikidata.org

[7] https://lov.linkeddata.es/dataset/lov/

The adoption of Linked Open Data (LOD) in recommender systems has shown promising improvements in their overall performance. Additionally, the integration of LOD into recommender systems significantly mitigates the cold-start problem commonly associated with these systems. The ESWC 2014 Challenge on Book Recommendations was an event held as part of the Extended Semantic Web Conference (ESWC) in 2014. This challenge focused on the application of Semantic Web technologies and LOD in the context of book recommendations. Participants were tasked with developing innovative recommender systems that utilized semantic web data and techniques to improve the quality and relevance of book recommendations [Di Noia et al., 2014]. This approach is not limited to book recommendations but extends to various domains where knowledge graphs play a pivotal role. For instance, in the work by Novo and Gedikli [2023], a knowledge graph is constructed by extracting entities such as organizations, people, and places from news articles. These named entities are crucial for detecting semantically similar news articles and identifying news story chains [Gedikli et al., 2021]. The semantically extracted data from this process is then effectively utilized for enhancing news recommendation systems.

6.2.3 Graph Neural Networks

The graph data structure on which knowledge graphs are based is particularly versatile and suitable for integrating data from different media sources. This flexibility underscores the potential of Graph Neural Networks (GNNs) in recommender systems. For instance, recent applications like Spotify's music recommendation engine leverage GNNs to analyze complex user-song interaction networks [Patil, 2023]. YouTube's video recommender leverages deep learning, possibly including GNNs, to recommend videos based on user interactions and viewing patterns [Covington et al., 2016]. Again, embeddings are used as key technology to encode videos, user sessions and even contextual data such as the user's geographic region and device.

The emergence of Graph Neural Networks (GNNs) is largely attributed to advancements in graph representation learning (GRL) [Gao et al., 2023]. GRL focuses on transforming graph elements like nodes or edges into low-dimensional vectors that encapsulate the complex structural interconnections within graphs. For instance, in a network of news articles, nodes could represent various entities extracted from the content. These graph embeddings are crucial as they provide a means to quantify and analyze the relational patterns and properties of these entities within the graph structure, making them essential for tasks like node classification, link prediction, or clustering in GNNs. It can be seen that embeddings also play a crucial role in exogenous methods. They bridge the gap between complex graph structures and the practical applicability of GNNs in various domains.

Generally, GNNs are characterized by their ability to combine different forms of content data, such as text, images and user interactions, into a unified graph-based framework. This unique capability enables a more comprehensive analysis and understanding of the data, increasing the effectiveness of recommendation systems.

For a thorough investigation of these concepts in GNN-based recommender systems, Gao et al. [2023] provide a comprehensive review of the literature dealing with the integration of diverse data in GNN architectures and their application in recommender scenarios.

7 Key Applications

Semantics-based recommender systems and hybrid approaches that leverage side information of various types have been in numerous application settings. The examples discussed in the early survey in Pazzani and Billsus [2007] among others comprise news recommendation and web site recommendation based on content analysis, as well as an e-commerce setting, where users can state their preferred item categories. In particular the latter example points to a potential advantage of semantics-based methods in terms of explanability and user control over the recommendations, see also [Musto et al., 2022].

In principle, however, semantics-based recommender systems are not limited in terms of application use cases, even though the academic literature mostly focuses on their application in the media domain (news, movies, music) and e-commerce. The timely overviews in [Musto et al., 2022, Sun et al., 2019] list numerous recent works on semantics-aware approaches and hybrids that leverage various types of side information. Generally, key application use cases are those in which data sparsity is high, where the item catalog is frequently changing, where explanability is a concern, and where relevant information sources about the items are available, either endogenously in the form of textual descriptions or exogenously in the form of structured or unstructured (external) knowledge about the items.

8 Future Directions

A number of trends and future directions of semantics-aware recommender systems are discussed in [Musto et al., 2022] and [Lops et al., 2019]. In [Lops et al., 2019], current trends are categorized as *data-related* trends and *algorithm-related* ones. The data-related trends comprise the continued use of Linked Data, User-Generated Content, Mutimedia Content, and Heterogeneous Information Networks. Algorithmic trends include the increased use of Deep Learning methods, new metadata encodings and Meta-Path based approaches.

In terms of future directions, a vast potential still lies in the extraction of semantic information from novel information sources, in particular from multimedia content. Furthermore, various approaches were put forward in most recent years that try to leverage the ‘world knowlege’ encoded in LLMs for enhanced recommender systems, see [Lin et al., 2023, Wu et al., 2023]. One example of such a work is [Harte et al., 2023], where marked performance improvements can be achieved in a sequential recommendation setting by incorporating external item embeddings in the recommendation process. Finally, the latest developments in the area of LLMs open entirely new perspectives for the development of conversational and multi-modal recommender systems.

9 Cross-References

References

Oren Barkan and Noam Koenigstein. Item2Vec: Neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2016.

Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12):29–38, 1992.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Paul Covington, Jay Adams, and Emre Sargin. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA, 2016.

Tommaso Di Noia, Iván Cantador, and Vito Ostuni. Linked open data-enabled recommender systems: Eswc 2014 challenge on book recommendation. In *Semantic Web Evaluation Challenge - SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014*, volume 475, pages 129–143, 05 2014. ISBN 978-3-319-12023-2.

J. R. Firth. *Studies in Linguistic Analysis*. Wiley-Blackwell, 1957.

Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Trans. Recomm. Syst.*, 1(1), 2023.

Fatih Gedikli and Dietmar Jannach. Improving recommendation accuracy based on item-specific tag preferences. *ACM Trans. Intell. Syst. Technol.*, 4(1):11:1–11:19, 2013.

Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should I explain? A comparison of different explanation types for recommender systems. *Int. J. Hum. Comput. Stud.*, 72(4):367–382, 2014.

Fatih Gedikli, Anne Stockem Novo, and Dietmar Jannach. Semi-automated identification of news story chains: A new dataset and entity-based labeling method. In *Proceedings of the 9th International Workshop on News Recommendation and Analytics (INRA 2021) in conjunction with the 15th ACM Conference on Recommender Systems (RecSys 2021)*, September 2021.

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. Leveraging large language models for sequential recommendation. In *17th ACM Conference on Recommender Systems (Late Breaking Results)*, 2023.

Dietmar Jannach and Kolja Hegelich. A case study on the effectiveness of recommendations in the mobile internet. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 205–208, 2009.

Dietmar Jannach and Markus Zanker. Impact and value of recommender systems. In Francesco Ricci, Bracha Shapira, and Lior Rokach, editors, *Recommender Systems Handbook*. Springer US, 2021.

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems – An Introduction*. Cambridge University Press, 2011.

Evan Kirshenbaum, George Forman, and Michael Dugan. A live comparison of methods for personalized article recommendation at Forbes.com. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 51–66, 2012.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196. PMLR, 22–24 Jun 2014.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. How can recommender systems benefit from large language models: A survey. arXiv preprint, https://arxiv.org/abs/2306.05817, 2023.

Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. Trends in content-based recommendation. *User Modeling and User-Adapted Interaction*, 29:1–1, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

Cataldo Musto, Marco de Gemmis, Pasquale Lops, Fedelucio Narducci, and Giovanni Semeraro. *Semantics and Content-Based Recommendations*, pages 251–298. Springer US, New York, NY, 2022.

Tien T. Nguyen and John Riedl. Predicting users' preference from tag relevance. In *User Modeling, Adaptation, and Personalization*, pages 274–280, 2013.

Anne Stockem Novo and Fatih Gedikli. Named entities as key features for detecting semantically similar news articles. *International Journal of Semantic Computing*, 17(4):633–649, 2023.

Shone Patil. GNN Spotify Recommender Website. https://shonepatil.github.io/GNN-Spotify-Recommender-Website/, 2023. Accessed: 29 January 2024.

Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems: Techniques, Applications, and Challenges*, pages 1–35. Springer US, 2022.

Babak Maleki Shoja and Nasseh Tabrizi. Tags-aware recommender systems: A systematic review. In *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD)*, pages 11–18, 2019.

Zhu Sun, Qing Guo, Jie Yang, Hui Fang, Guibing Guo, Jie Zhang, and Robin Burke. Research commentary on recommendations with side information: A survey and research directions. *Electronic Commerce Research and Applications*, 37:100879, 2019.

Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 225–232, 2016.

Jesse Vig, Shilad Sen, and John T Riedl. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 47–56, 2009.

Jesse Vig, Michael Soukup, Shilad Sen, and John T Riedl. Tag expression: tagging with feeling. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, pages 323–332, 2010.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation. arXiv preprint, https://arxiv.org/abs/2305.19860, 2023.