A Survey On Knowledge-Aware News Recommender Systems

Andreea Iana [a,*], Mehwish Alam [b,c] and Heiko Paulheim [a]

[a] *Data and Web Science Group, University of Mannheim, Germany*
E-mails: andreea@informatik.uni-mannheim.de, heiko@informatik.uni-mannheim.de
[b] *FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany*
E-mail: mehwish.alam@fiz-karlsruhe.de
[c] *Karlsruhe Institute of Technology, Institute AIFB, Germany*
E-mail: mehwish.alam@kit.edu

Abstract. News consumption has shifted over time from traditional media to online platforms, which use recommendation algorithms to help users navigate through the large incoming streams of daily news by suggesting relevant articles based on their preferences and reading behaviour. In comparison to domains such as movies or e-commerce, where recommender systems have proved highly successful, the characteristics of the news domain pose additional challenges for the recommendation models. While some of these can be overcome by conventional recommendation techniques, injecting external knowledge into news recommender systems has been proposed in order to enhance recommendations by capturing information and patterns not contained in the text and metadata of articles, and hence, tackle shortcomings of traditional models. This survey provides a comprehensive review of knowledge-aware news recommender systems. A new classification method divides the models into four categories: frameworks based on the vector space model, on semantic similarities, on distance, and on knowledge graph embeddings. Moreover, the underlying recommendation algorithms, as well as their evaluations are analysed. Lastly, open issues in the domain of knowledge-aware news recommendation are identified and potential research directions proposed.

Keywords: News Recommender System, Knowledge Graphs, Ontologies, Semantic Similarity, Knowledge Graph Embeddings, Evaluation Methodology, Survey

1. Introduction

In the past two decades, there has been a shift in individuals' news consumption, from traditional media, such as printed newspapers or radio and TV news broadcasts, to online media platforms, in the form of news websites and aggregation services, or social media. News platforms utilize a form of a recommender system to help users navigate through the overwhelming amount of news published daily by suggesting relevant articles based on their interests and reading behaviour. Recommender systems have proven successful over time in numerous domains [81], ranging from music (e.g. Spotify), movies (e.g. Netflix, MovieLens), or books recommendation [26, 42, 53], to e-commerce (e.g. Amazon, eBay), travel and tourism [10], or research paper recommendation [3].

In comparison to these domains, news poses additional challenges which hinder a direct transfer of traditional recommendation techniques to the task of news recommendation. Firstly, the relevance of news changes quickly within short periods of time and is highly dependent on the time sensitiveness and popularity of articles [59, 123]. Secondly, articles may be semantically related and users' interests evolve dynamically over time, meaning that it is not trivial to accurately capture the preferences of individual users [123]. Thirdly, common limitations of recommender systems (i.e. the cold-start problem, data sparsity, scalability), are further intensified by the greater item churn of news [27], the fact that usually user profiles

[*] Corresponding author. E-mail: andreea@informatik.uni-mannheim.de.

are constrained to a single session [30], and that their feedback is typically collected implicitly, from their reading behaviour rather than explicitly provided during a session [71]. Additionally, news articles contain a large number of knowledge entities and common sense knowledge, which are not incorporated in conventional news recommendation methods [102].

Enhancing classic information retrieval and recommendation methods with external information from knowledge bases has been proposed as a potential solution for some of the aforementioned shortcomings of recommender systems in the news domain. Knowledge graphs are directed heterogeneous graphs which describe real-world entities and their interrelations [82]. Knowledge-aware recommender systems inject information contained in knowledge graphs or domain-specific ontologies to capture information and reveal patterns that are not contained directly in an item's features [46]. In the case of news recommendation, such knowledge-enhanced models have been developed to capture the semantic meaning and relatedness of news, remove ambiguity, handle named entities, extend text-level information with common sense knowledge, discover knowledge-level connections between news, and overcome cold-start and data sparsity issues.

Previous works provide overviews of this field from two directions. On the one hand, surveys such as [46] or [40], focus on knowledge-aware recommender systems applied to a variety of domains, such as movies, books, music, or products. Although a few of the discussed models come from the news domain, none of these works extensively review how external knowledge can be used to enhance news recommendation. On the other hand, a vast number of surveys analyse the news recommendation problem from various angles, including challenges and algorithmic approaches [6, 9, 32, 59, 64, 66, 123], performance comparison in online news recommendation challenges [29, 30], user profiling [47], news features-based methods [84], or impact on content diversity [76]. However, the focus of these studies is not on knowledge-aware recommendation techniques. In contrast to existing studies, this survey focuses on categorizing and examining knowledge-aware news recommender systems, developed either specifically for or evaluated also on the news domain, as a solution for enhancing recommendations and overcoming limitations of traditional recommendation models. The analysis of such systems covers both a review of the algorithmic approaches utilized for computing recommendations, as well as a comparison of evaluation methodologies and a discussion of limitations and research gaps.

The contributions of the paper are threefold:

1. A **new taxonomy** of knowledge-aware news recommender systems is proposed. The recommendation approaches are classified into methods based on the vector space model, on semantic similarities, on distance, and on knowledge graph embeddings.
2. This survey aims to provide a **comprehensive review** of recommender systems for the news domain which utilize knowledge bases as external sources of information. For each category of models, a detailed analysis of the representative models is provided, including relevant comparisons and descriptions of the algorithms, as well as of the evaluation methodologies used.
3. The limitations of existing models and open issues in the field of knowledge-aware news recommender systems are identified and examined, and five potential **future research directions** are proposed in terms of comparability of evaluations, scalability of systems, explainability and fairness of results, and multi-task learning for recommendation.

The rest of the article is structured as follows. Section 2 introduces recommender systems and outlines challenges specific to the news domain, while Section 3 covers related work in news and knowledge-aware recommender systems. Section 4 introduces and defines commonly used notations and concepts, and analyses different aspects of knowledge-aware news recommenders. Section 5 classifies and discusses knowledge-aware news recommender systems, whereas Section 6 investigates various evaluation approaches adopted by the different models. Section 7 discusses open issues identified in the field. Finally, the survey is summarized in Section 8.

2. Challenges in News Recommendation

Recommender systems consist of techniques that filter information and generate recommendations of items deemed potentially interesting for users, based on their preferences and past behaviour, in order to help individuals overcome information overload [89]. User's preferences are learned using either explicit (e.g. ratings) or implicit (e.g. browsing history) feedback [55]. Recommender systems are generally categorized into collaborative filtering, content-based, and hybrid methods, based on the underlying algorithm. Collaborative filtering systems recommend items liked in the past by users with similar preferences to the current user [1]. In content-based algorithms, the recommendations depend only on the user's past ratings of items, meaning that the suggested items will have similar characteristics to the ones preferred in the past by the current user [1]. Hybrid models combine one or more types of recommendation approaches to alleviate the weaknesses of a single technique, such as the cold-start problem (which refers to the difficulty in the computation of the recommendations for new items, without ratings, or new users, without a profile) or the over-specialization issue (i.e. the lack of diversity and serendipity in results) [11].

The unique characteristics of news not only distinguish them from items in domains such as online retail, movies, music, or tourism, where recommender systems have already proven successful, but also impede the straightforward application of conventional recommendation algorithms to the task of news recommendation. A large quantity of news is published every day, with articles being continuously updated. Such a **large volume of data**, spread over short periods of time, combined with the **unstructured format** of news articles, requires more complex analyses and heavier computations [64]. In addition, news are characterized by **short shelf lives** and **high item churn**, as their relevance highly depends on the **recency** articles, since users prefer reading about the latest events that took place [27, 64]. A topic's **popularity** also significantly influences the importance of an article, as stories can become quickly outdated and lose relevance when they are superseded by "breaking news" [59]. For example, while a reader might be concerned with news about the elections in a country for multiple days or even weeks, he will be less likely to be interested in the results of a tennis match a week after a tournament has finished.

Furthermore, the user's interests evolve over time as individuals display both **short-term and long-term preferences**. On the one hand, individuals display long-term interests in certain topics, motivated by their socio-economic and personal background, such as a user being interested in climate change for several years [2, 47]. On the other hand, highly popular news might affect a user's short-term interest, which changes more rapidly, within a short time span [2]. For example, a user might read several news articles related to GameStops's short squeeze after browsing the "latest news" section of a website which announced that Robinhood has limited the buying and trading of GameStop stocks. However, in the news domain, users are usually not required to sign in and create profiles in order to read articles. Moreover, they rarely provide explicit feedback in terms of likes and ratings. In turn, this means that their profiles are either limited to a single session or tracked through browser cookies, and that **feedback is gathered implicitly** by analysing the clicks stored in logs [29, 30]. Overall, these characteristics of users in the news domain pose an additional challenge for creating an accurate user profile for the recommendation algorithms. Additionally, the lack of feedback on news articles and the small amount of data available for user profiling further amplifies the cold-start and data sparsity problems of recommender systems [59, 123].

News articles often describe events that occur in the world, which can be represented in terms of **named entities** that indicate what, when, where the event happened, as well as who was involved [64]. Additionally, news recommendations can also be subjected to **over-specialization** issues as users are being suggested articles semantically similar to the ones already read, but published in different sources and written using terms that are related through semantic relations, such as synonymy or antonymy [66, 123].

3. Related Work

The current section gives an overview of surveys published in the areas of news recommendation and knowledge-aware recommender systems.

3.1. News Recommender Systems

Several surveys on news recommender systems and corresponding issues have been conducted. A comparison and evaluation of content-based news recommenders are performed in [6]. Borges and Lorena [9] first provide a high-level overview of recommender systems in general, including similarity measures and evaluation metrics, followed by an in-depth analysis of six models applied in the news domain. A more general overview and comparison of the mechanisms and algorithms used by news recommendation approaches, as well as corresponding strengths and weaknesses, is provided by Dwivedi and Arya [32].

Özgöbek et al. [123] identify the challenges specific to the news domain and discuss twelve recommendation models according to the targeted problems, without considering evaluation approaches. In contrast to these studies, Karimi et al. [59] provide a comprehensive review of news recommender systems, not only by taking into account a large number of challenges and algorithmic approaches proposed as a solution, but also by discussing approaches and datasets used in evaluating such systems, as well as proposing future research directions from the perspectives of algorithms and data, and the aspect of evaluation methodologies.

Li et al. [64] review issues characterizing the field of personalized news recommendation and investigate existing approaches from the perspectives of data scalability, user profiling, as well as news selection and ranking. Additionally, the authors conduct an empirical study on a collection of news articles gathered from two news websites in order to examine the influence of different methods of news clustering, user profiling, and feature representation on personalized news recommendation. More recently, Li and Wang [66] analyse state-of-the-art technologies proposed for personalized news recommendation, by classifying them according to seven addressed news characteristics, namely data sparsity, cold-start, rich contextual information, social information, popularity effect, massive data processing, and privacy problems. Furthermore, they discuss the advantages and disadvantages of different kinds of data used in personalized news recommendation, as well as open issues in the field [66].

In comparison to the previous general studies, Harandi and Gulla [47] investigate and categorize approaches used for user profiling in news recommendation according to the problems addressed and the types of features utilized. Lastly, Qin and Lu [84] survey feature-based news recommendation techniques, which they categorize into location-based, time-based (i.e. further classified into real-time and session-based), and event-based methods.

Although these surveys provide comprehensive overviews of news recommendation methods, domain-specific challenges and evaluation methodologies, they do not discuss knowledge-aware models or the latest state-of-the-art recommendation methods. In contrast, our survey focuses solely on news recommender systems which incorporate external knowledge to enhance the recommendations and to overcome the limitations of conventional recommendation techniques.

3.2. Knowledge-Aware Recommender Systems

Knowledge graphs, a type of directed heterogeneous networks, describe real-world entities (represented as nodes) and multiple kinds of relations between them (represented as edges), either spanning multiple domains (e.g. Freebase [7], DBpedia [63], YAGO [96], Wikidata [101], Microsoft Satori [122]) or focusing on a particular field (e.g. Bio2RDF [4]) [34, 82]. In addition, such graphs can capture higher-order relations connecting entities with several related attributes [46].

This strong representation ability of knowledge graphs has attracted the attention of the research community working on developing and improving recommender systems for several reasons. Firstly, using knowledge graphs as side information in recommendation models can help diminish common limitations, such as data sparsity and the cold-start problem [46]. Secondly, the precision of recommendations can be improved by extracting latent semantic connections between items, while the diversity of results can be increased by extending the user's preferences taking into account the variety of relations between items encoded in a knowledge graph [40, 112]. Another advantage of using knowledge graphs as background information is improving the explainability of recommendations, to ensure trustworthy recommendation systems, by considering the connections between a user's previously liked items and the generated suggestions, represented as paths in the knowledge graph [112].

Guo et al. [46] provide a detailed review and analysis of knowledge graph-based recommender systems, which are classified into three categories, according to the strategy employed for utilizing the knowledge graph, namely embedding-based, path-based and unified methods. In addition to comparing the algorithms used by the three types of methods, the authors also analyse how knowledge graphs are utilized to create explainable recommendations. Lastly, the survey clusters relevant works according to their application and introduces the datasets commonly used for evaluation in each category [46].

Recent advancements in deep learning techniques for graph data, in the form of Graph Neural Networks (GNN) [112, 120], have given rise to new knowledge-aware, deep recommender systems. Gao et al. [40] are the first to provide a comprehensive overview of GNN-based knowledge-aware deep recommender (GNN-KADR) systems, in which they analyse recommendation techniques, discuss how challenges such as scalability or personalization are addressed, and briefly summarize the domain-specific datasets and metrics used for evaluation, before suggesting a number of possible directions for future research.

Gao et al. [40] categorize GNN-KADRs depending on the type of graph neural network components used for recommendation. More specifically, graph neural networks are comprised of an aggregator, which combines the feature information of a node's neighbourhood to obtain the context representation, and an updater, which uses this contextual information together with the input information for a given graph node in order to compute its new embedding. According to Gao et al. [40], aggregators are divided into relation-unaware (i.e. the relation information between nodes is not encoded in the context representation) and relation-aware aggregators (i.e. the information contained in different relations is considered in the context representation). The latter category is further split into relation-aware subgraph aggregator and relation-aware attentive aggregator, depending on how the relations in the knowledge graph are modelled in the framework [40]. The first subcategory creates multiple subgraphs for each relation type found in a node's neighbourhood graph, while the second encodes the semantic information contained in the edges of the knowledge graph using weights which measure how related different knowledge triples are to the target node [40]. Similarly, updaters are also categorized into three clusters, namely context-only updaters (i.e. only the node's context representation is used to produce its new embedding), single-interaction updaters (i.e. both the target node's current embedding, as well as its context representation are used to obtain its updated representation), and multi-interaction updaters (i.e. different binary operators combine multiple single-interaction updaters), where the first two groups of updaters are more often encountered [40].

GNN-based recommender systems are investigated also by Wu et al. [112], who classify the recommendation models based on whether the models consider the item's ordering (i.e. general vs. sequential methods) and on the type of information used (i.e. without side information, social network-enhanced, and knowledge graph-enhanced). According to the proposed taxonomy of Wu et al. [112], knowledge-aware models can be found only in the group of general recommender systems. In this category, four representative recommendation frameworks are examined from the aspects of graph simplification, multi-relation propagation, and user integration.

The research commentary of Sun et al. [97] consists of an extensive, systematic survey of recent advancements in recommender systems that use side information. The models, mostly collaborative filtering techniques, are analysed from two perspectives. On the one hand, Sun et al. [97] categorize the models according to the evolution of fundamental methodological approaches into memory-based and model-based frameworks, where the latter category is further split into latent factor models, representation learning models and deep learning models. On the other hand, the recommender systems are classified based on the evolution of side information used for recommendation, into models using structural data and models using non-structural data. The first group includes information in the form of flat features, network features, feature hierarchies, and knowledge graphs, whereas the second consists of text, image, and video features [97].

In the surveys discussed above, knowledge-aware news recommender systems are rarely analysed. In comparison to these works, the current survey focuses on the investigation of approaches for injecting external knowledge only into the news recommendation model. To this end, it provides a categorization and an extensive overview of the knowledge-aware recommender systems developed either for or evaluated also in the news domain.

4. Definitions and Categorization

This section firstly introduces and defines commonly used concepts and notations. Afterwards, it provides an overview of knowledge-aware news recommender systems according to multiple criteria.

4.1. Definitions

Firstly, a minimal set of concepts and notations referred to in the rest of the article are defined. Bold uppercase characters denote matrices, while bold lowercase characters generally indicate vectors. The notations used throughout this article are illustrated in Table 1, unless specified otherwise.

Definition 1. An **ontology** is defined as a set of *k concepts* [54]:

$$\mathcal{O} = \{c_1, c_2, ..., c_k\} \qquad (1)$$

In many cases, concepts are distinguished into classes $\mathcal{C}$ and relations $\mathcal{R}$, so that $\mathcal{O} = \mathcal{C} \cup \mathcal{R}$, and $\mathcal{C} \cap \mathcal{R} = \emptyset$.

Table 1

Commonly used notations

Notations	Descriptions
$\lvert \cdot \rvert$	Set size
$\odot$	Element-wise product
$\oplus$	Vector concatenation
$tanh$	Hyperbolic tangent function
$\sigma(\cdot)$	Nonlinear transformation function (e.g. sigmoid)
X^T	Transpose of matrix X
$\mathcal{G}$	A knowledge graph
c_i	Concept i in the ontology $\mathcal{O}$
e_i	Entity i in $\mathcal{G}$ (either head or tail)
$\mathcal{N}(e_i)$	Neighbours of entity e_i
k	Dimension of knowledge graph embedding
$\mathbf{e}_i \in \mathbb{R}^{k \times 1}$	Embedding of entity e_i in $\mathcal{G}$
$\mathbf{r} \in \mathbb{R}^{k \times 1}$	Embedding of relation r in $\mathcal{G}$
u_i	Profile of user i
v_j	Profile of item j
$\mathcal{U} = \{u_1, u_2, ..., u_M\}$	Set of users
$\mathcal{V} = \{v_1, v_2, ..., v_N\}$	Set of items
M	The number of users in $\mathcal{U}$
N	The number of items in $\mathcal{V}$
$\hat{y}_{u,v}$	User u's probability of interacting with item v
d	Dimension of a feature vector
$\mathbf{u}_i \in \mathbb{R}^{d \times 1}$	Feature vector of user u_i
$\mathbf{v}_j \in \mathbb{R}^{d \times 1}$	Feature vector of item v_j
$\mathbf{V}_i, \mathbf{W}_i, \mathbf{w}_i$	Trainable weight matrices and vectors
$\mathbf{b}_i$	Trainable bias vectors

Definition 2. The semantic neighbourhood of a concept c_i is defined as the set of concepts which are directly related to concept c_i, including itself [54]:

$$N(c_i) = \{c^i_1, c^i_2, ..., c^i_k\} \tag{2}$$

Definition 3. A **knowledge graph** (KG) $\mathcal{G} = (V, E)$ is a labeled directed graph, where the nodes represent *entities*. Edges are of the form $\langle e_h, r, e_t \rangle \in E$, and indicate a relationship $r \in \mathcal{R}$ from head entity e_h to tail entity e_t, where $e_h, e_t \in V$. Edges can be interpreted as subject-property-object triple facts [13]. Often, entities are assigned one or more types, defined by type statements of the form $\langle e, t \rangle$, where $e \in E$ and $t \in \mathcal{C}$.

4.2. *Categorization of Knowledge-Aware News Recommender Systems*

Knowledge-aware news recommendation models can be investigated according to multiple criteria, ranging from the used knowledge resource to the output types or addressed challenges.

4.2.1. *Types of Recommendation Techniques*

News recommendation systems generally adopt one of the three main techniques for predicting whether a user will interact with a certain article, namely content-based, collaborative filtering, and hybrid. However, content-based approaches are the most widely used in the field of news recommendation [59].

4.2.2. *Knowledge Base*

The knowledge resources used by knowledge-aware recommender systems can be grouped into domain ontologies and knowledge graphs. In the remainder of the paper, these will be referred to as *knowledge bases* (KB), if the type of resource is not explicitly specified. The former category can be further split into self-constructed ontologies – built either from combining smaller domain ontologies or subsets of large knowledge bases (e.g. DBpedia, Hudong encyclopedia) or directly from news articles (i.e. financial domain ontology using information from Yahoo! Finance [54])

– and controlled vocabularies utilized in the news domain, such as the IPTC News Codes[1] [99]).

In the latter category, one can distinguish between open source and commercial knowledge graphs. In the first subgroup, cross-domain knowledge graphs such as Wikidata, DBpedia, and Freebase are widely used in news recommender systems. Freebase [7] was initially launched by Metaweb in 2007, and later acquired by Google in 2010, before being shut down in 2015 [82]. The latest version of Freebase, available at Google's Data Dumps[2] has been estimated to contain 1.9 billion triples [43]. Wikidata [101], a collaboratively edited knowledge graph, containing several language editions of Wikipedia, as well as data previously contained in Freebase [82], consists of 92 million items[3] and over 1174 million statements[4]. DBpedia [63] is a knowledge graph built by extracting structured data from various language versions of Wikipedia, and contains in its most recent and largest version, DBpedia Largest Diamond, 220 million entities and 1.45 billion triples[5].

WordNet [74], a large English lexicon containing nouns, verbs, adjectives and adverbs grouped into synsets (i.e. sets of synonyms), which are further interconnected via semantic relations of antonymy, hyponymy, meronymy, troponomy, or entailment, is often used in knowledge-aware news recommender systems for word disambiguation. More specifically, each term in WordNet is associated with a set of senses, which denote the set of possible meanings that the word might have. For example, the noun "Jupiter" can refer to either the planet in the solar system or the supreme god of the Romans. WordNet 3.0[6] contains 117,659 synsets and 206,941 word-sense pairs.

In the subgroup of commercial knowledge bases, Satori [122], the knowledge graph proposed by Microsoft, is the most often utilized one, especially by recent deep learning-based news recommender systems. Although very little information about the data contained in Satori is publicly available, it was estimated to contain in 2012 approximately 300 million entities and 800 million relations [82].

4.2.3. *Structure of Knowledge Base*

News recommendation models utilize knowledge bases by exploiting their different structures in order to extract either semantic, structural, or both types of information. The majority of older knowledge-aware news recommender systems exploit mostly the semantic information contained in a knowledge graph or ontology, by extracting concepts or entities that appear in a news article, which will be denoted as *concepts/entities only* models for the rest of this article. This basic set of knowledge entities can be further enriched in two ways. Firstly, the entity set can be expanded with the neighbourhoods of extracted entities in the knowledge graph by considering the paths between entities (denoted as *entities+paths*). Secondly, concepts extracted from an ontology can be enhanced by considering the ontological structures, or different types of relations between nodes in an ontology, such as synonymy or hyponymy relationships in semantic lexicons (denoted as *concepts+KB structure*). The structural information of an ontology is exploited in a different manner also by recommendation models which define the similarity between news articles based on the distance between the encompassed concepts or entities (the latter is denoted as *entities+KB structure*). Differently from these categories of models, the newer deep-learning-based recommendation techniques exploit simultaneously both the semantic and the structural information encoded in knowledge graphs, by means of knowledge graph embeddings (denoted as *entities+KG structure*).

4.2.4. *Outputs*

Two main outputs can be distinguished in news recommendation models, namely click-through rate (CTR) prediction and item ranking. Models classified in the first group aim to predict the probability that the user will click on the target article, whereas methods in the second group recommend the top *N* most similar articles to the articles previously read by the user.

4.2.5. *Addressed Challenge*

In addition to enhancing the accuracy of recommendations, knowledge-aware news recommender systems aim to address different challenges of the news domain or limitations posed by conventional recommendation techniques. Several articles, written in different manners, using semantically related terms, can describe the same piece of news, and numerous words have different meanings depending on the context in which they are used. While humans can easily distinguish ambiguous words, or words connected via certain semantic relations, such as synonyms, this constitutes a challenge for recommendation models using text representations. Knowledge-aware recommender systems pro-

[1] https://iptc.org/standards/newscodes/
[2] https://developers.google.com/freebase
[3] https://www.wikidata.org/wiki/Wikidata:Statistics
[4] https://wikidata-todo.toolforge.org/stats.php
[5] https://www.dbpedia.org/resources/knowledge-graphs/
[6] https://wordnet.princeton.edu/documentation/wnstats7wn

pose to remove such ambiguity from text by representing an article using only disambiguated knowledge entities or concepts from a controlled vocabulary, instead of all the terms. In turn, this leads to faster computations, since the model is required to consider a limited number of concepts or entities, which is significantly smaller than the total number of words contained in an article. Moreover, the semantic meaning of news, as well as the semantic relatedness of concepts (i.e. news describing similar or related concepts might indicate different interests of a user) can be captured by further considering the relations between the different concepts found in an article.

News articles contain a large number of named entities, used to denote information regarding the events described, such as the location, actors involved, time, or what the event refers to. However, named entities are not taken into account in traditional text-based recommendation models. In contrast, knowledge-aware techniques handle named entities by extracting them from the text and enriching them with external information encoded in knowledge graphs. Furthermore, using external information for recommendation can help overcome the data sparsity and cold-start problems, as articles can be connected using relations in the knowledge graph between the entities extracted from text, such that new items without user feedback can also be included in the recommendations.

Moreover, injecting external knowledge into the recommendation model has three additional benefits. Firstly, it extends text-level information with common sense knowledge which is encoded in knowledge graphs, but cannot be extracted only from an article's text. For example, a user reading the titles of the news articles in Figure 1 will probably know that *Elon Musk* and *Robinhood* were participants in the *GameStop short squeeze* event that affected *GameStop*, or that the *New York Stock Exchange* is located on *Wall Street*. However, a text-based recommendation model does not possess such common knowledge information. Additionally, using external information also helps the model discover latent knowledge-level connections between the news, such as the fact that the two snippets in the example from Figure 1 are connected, although they do not appear related when considering only the words in their titles. Lastly, exploiting the knowledge-level and semantic connections between news can improve the diversity of recommendations, as the model learns to avoid recommending articles that are too semantically similar, even if they are published in different sources and have a different writing styles.

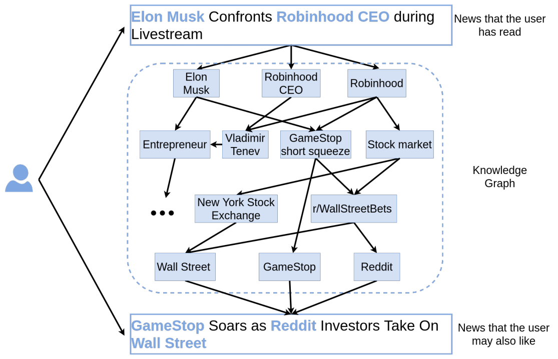

Fig. 1. Illustration of a knowledge graph-enhanced news recommender system (reproduced from [102]).

5. Knowledge-Aware News Recommendation Models

Knowledge-aware news recommendation systems can be classified into four categories based on how external knowledge is injected in the recommendation model, on the way in which item-item recommendations are computed, as well as on the utilized structures of the knowledge base. This taxonomy, illustrated in Figure 2, distinguishes between methods based on the vector space model, on semantic similarities, on distance, and on knowledge graph embeddings.

To facilitate readers reviewing the literature, the surveyed models are listed in Table 2 according to the aforementioned criteria. It should be noted that papers which only introduce the idea for a recommender system, without presenting the underlying algorithm or the framework's architecture, as well as works lacking an evaluation of the presented system were excluded from this survey. Furthermore, only models designed specifically for news recommendation, or evaluated on several downstream tasks, including the news domain, have been considered.

The following subsections analyse the knowledge-aware recommender systems presented in Table 2 according to the taxonomy introduced above. For each category of models, the overall framework, as well as the representative models are investigated.

5.1. Methods based on the vector space model

Recommender systems classified in this category adopt a content-based recommendation approach and compute the similarity between news articles using different variants of the Vector Space Model (VSM) [93], modified to take into consideration side information from a knowledge base.

Table 2
Overview of knowledge-aware news recommendation approaches. The first column categorizes the papers according to the new taxonomy. The next two columns list the model's abbreviated name and its publication year. The fourth column indicates the type of recommendation technique. The fifth column specifies the external knowledge resource used, while the sixth column shows which knowledge base structures are utilized by the respective model. The seventh column shows the framework's output type. The last column lists the challenges addressed by injecting external knowledge in the recommendation model. "Accuracy" is not explicitly mentioned as a challenge, as all discussed models aim to improve recommendations on this measure. The abbreviations used in the table are the following: VSMM=Methods based on the vector space model, SSM=Methods based on semantic similarities, DM=Methods based on distance, KGEM=Methods based on knowledge graph embeddings, RS=Recommender system, CB=Content-based, CF=Collaborative filtering, H=Hybrid, DO=domain ontology.

Category	Model	Year	RS Type	KB	KB Structure	Output	Addressed Challenge
VSMM	CF-IDF [44]	2011	CB	DO	concepts/entities only	Item ranking	-
	SF-IDF [20]	2012	CB	WordNet	concepts/entities only	Item ranking	Capture semantic meaning
	SF-IDF+ [75]	2013	CB	WordNet	entities+paths	Item ranking	Capture semantic relationships between synsets
	Bing-SF-IDF+ [22]	2015	CB	WordNet	entities+paths	Item ranking	Capture semantic meaning
	OF-IDF [86]	2015	CB	DO	entities+paths	Item ranking	Eliminate ambiguity
SSM	Semantic context-aware recommendation [14, 16]	2008	Hybrid	DO (IPTC + Wikipedia)	entities+paths	Item ranking	Capture semantic relatedness of concepts
	Social tags enriched recommendations [18]	2008	CB	Wordnet, Wikipedia DO (IPTC + Wikipedia)	entities+paths	Item ranking	Data sparsity, cold-start problem
	Semantic relatedness [41]	2009	CB	DO	entities+paths	Item ranking	Capture semantic relatedness of concepts
	RSR [54]	2010	CB	DO	entities+paths	Item ranking	Capture semantic meaning, faster computations, eliminate ambiguity
	Hybrid context-aware recommendation [17]	2011	Hybrid	DO (IPTC + Wikipedia)	entities+paths	Item ranking	Data sparsity, cold-start problem
	RSR 2 [37]	2012	CB	DO	entities+paths	Item ranking	Capture semantic meaning, faster computations, eliminate ambiguity
	SS [20]	2012	CB	WordNet	concepts/entities only	Item ranking	Capture semantic meaning
	BingSS [21]	2013	CB	WordNet	entities+paths	Item ranking	Handle knowledge entities
DM	ePaper [73, 94]	2008	CB	IPTC News Codes	concepts+KB structure	Item ranking	-
	Magellan [31]	2011	CB	DO	concepts+KB structure	Item ranking	-
	OBSM [85]	2013	CB	Ontologies based on DBpedia and Hudong	concepts+KB structure	Item ranking	Eliminate ambiguity
	SED [58]	2019	CB	Freebase	entities+KB structure	Item ranking	Cold-start problem
KGEM	CETR [117]	2017	CF	Wikidata	entities+KG structure	CTR	Extend text-level information
	DKN [102]	2018	CB	Microsoft Satori	entities+KG structure	CTR	Extract knowledge-level news connections, extend text-level information with common sense
	Fine-grained DKN with self-attention [38]	2018	CB	KG	entities+KG structure	CTR	Extract knowledge-level news connections
	RippleNet [103]	2018	CB	Microsoft Satori	entities+KG structure	CTR	Data sparsity, cold-start problem
	RippleNet-agg [104]	2019	CB	Microsoft Satori	entities+KG structure	CTR	Data sparsity, cold-start problem
	MKR [106]	2019	CB	Microsoft Satori	entities+KG structure	CTR Item ranking	Data sparsity, cold-start problem
	IGNN [83]	2019	CB	Wikipedia	entities+KG structure	CTR	Mine high-order connectivity of users and news
	Saskr [25]	2019	CB	News-specific KG	entities+KG structure	CTR	Extract knowledge-level news connections, diversity, cold-start problem, handle knowledge entities
	KRED [70]	2020	CB	Microsoft Satori	entities+KG structure	CTR	Handle knowledge entities
	TEKGR [62]	2020	CB	Microsoft Satori	entities+KG structure	CTR	Handle knowledge entities, capture topical relatedness between news

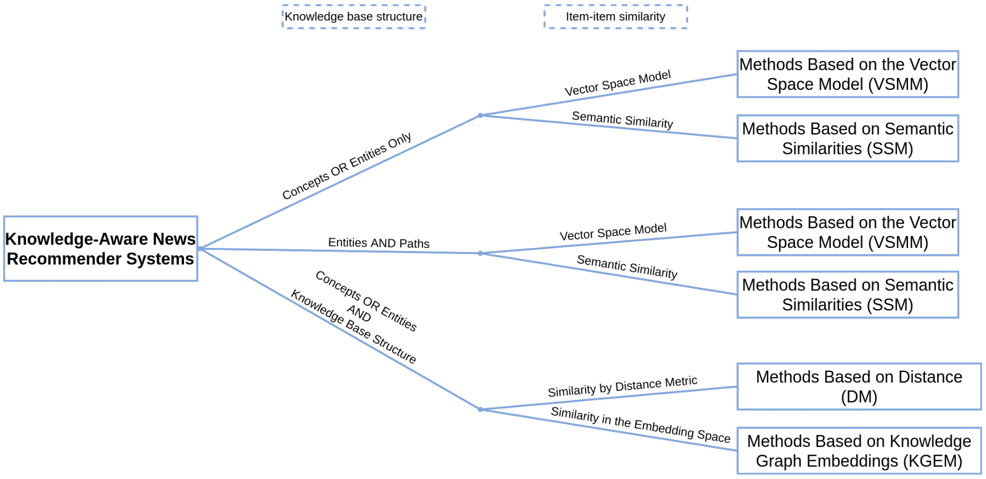
Knowledge base structure
Item-item similarity
Knowledge-Aware News Recommender Systems
Concepts OR Entities Only
Entities AND Paths
Concepts OR Entities AND Knowledge Base Structure
Vector Space Model
Semantic Similarity
Vector Space Model
Semantic Similarity
Similarity by Distance Metric
Similarity in the Embedding Space
Methods Based on the Vector Space Model (VSMM)
Methods Based on Semantic Similarities (SSM)
Methods Based on the Vector Space Model (VSMM)
Methods Based on Semantic Similarities (SSM)
Methods Based on Distance (DM)
Methods Based on Knowledge Graph Embeddings (KGEM)

Fig. 2. The categorization of knowledge-aware news recommender systems. We divide existing frameworks into four categories, based on the type of knowledge base structure utilized and the way of computing item-item similarity.

5.1.1. Overall framework

The methods based on the vector space model first create a vector representation of both the target article, and the user profile, where the latter comprises of the user's reading history. Afterwards, the models compute the cosine similarity between the two profiles and recommend a list of the top N articles whose similarity scores exceed a predefined threshold. These systems are analysed in terms of two differentiating factors:

- **Profile representation.** The representation of the items and users determines which kind of semantic information is incorporated in the model.
- **Weighting scheme.** Several weighting approaches are used to measure the importance of the components used to represent the news articles.

5.1.2. Representative models

This subsection discusses five representative recommendation techniques based on the vector space model.

Concept Frequency - Inverse Document Frequency (**CF-IDF**) [44] constitutes a variant of the Term Frequency - Inverse Document Frequency (TF-IDF) [92] weighting scheme that uses concepts instead of terms in order to represent news articles. In the framework proposed by Goossen et al. [44], the profile of a user u consists of a set of q concepts from an ontology, namely $u = \{c_1^u, c_2^u, ..., c_q^u\}, c_i^u \in \mathcal{O}$. In turn, each concept c^u from the user profile is represented as a set of k news articles v_j in which it occurs, namely $c^u = \{v_1, v_2, ..., v_k\}$. An article is thus composed of a set of p concepts occurring in it, denoted as $v = \{c_1^n, c_2^n, ..., c_p^n\}, c^n \in \mathcal{O}$.

In the CF-IDF recommender, each user's interests are represented as a vector of CF-IDF weights w^u corresponding to the concepts appearing in the user's previously read articles, as shown in Eq. (3). Analogously, the article's profile is computed according to Eq. (4).

$$\mathbf{u} = [< c_1^u, w_1^u >, ..., < c_q^u, w_q^u >] \tag{3}$$

$$\mathbf{v} = [< c_1^n, w_1^n >, ..., < c_p^n, w_p^n >] \tag{4}$$

The CF-IDF weights are computed similarly to TF-IDF weights. Firstly, the *Concept Frequency* $cf_{i,j}$ calculates the frequency of a concept c_i in an article v_j as the ratio between the number of occurrences in the given article, $n_{i,j}$, and number of occurrences of all concepts appearing in the article, $n_{k,j}$. Since highly frequent concepts are less informative than rarer ones, the *Inverse Document Frequency* idf_i penalizes such concepts by increasing the weight of concepts rarely occurring across all $|D|$ articles in the corpus. For a concept c_i, this is achieved by computing the logarithmically scaled inverse fraction of documents containing the concept. The final weight is given by multiplying the two components according to Eq. (5).

$$cf - idf_{i,j} = cf_{i,j} \times idf_i = \frac{n_{i,j}}{\sum_k n_{k,j}} \times \log \frac{|D|}{|d : c_i \in d|} \tag{5}$$

The main difference between the TF-IDF and CF-IDF lies in the fact that the latter considers only the ontology concepts contained in the text, instead of all the terms. Therefore, it assigns a larger value to the concepts deemed more important, and results in faster computations, as it considers a smaller amount of elements during similarity computations.

The Synset Frequency - Inverse Document Frequency (**SF-IDF**) approach of Capelle et al. [20] modifies the TF-IDF weighting scheme to take into account the semantic meaning of terms in a text. In comparison to CF-IDF, Capelle et al. [20] represent the user's and article's profile as sets of WordNet synsets of the terms appearing the news article. Mathematically, the news item's profile is represented as:

$$v = \{s_1^n, s_2^n, ..., s_p^n\} \quad (6)$$

where s_i^n denotes a WordNet synset of a term from the article, and p the total number of synsets contained in it. The user profile is obtained by aggregating the vector representations v_q of all the Q news articles in his reading history, denoted as:

$$u = \bigcup_{q \in Q} v_q = \bigcup_{q \in Q} \{s_1^u, s_2^u, ..., s_q^u\} \quad (7)$$

The synsets in both profiles are weighted using SF-IDF weights, obtained from TF-IDF by replacing terms with synsets s, i.e. $sf - idf_{s,d} = sf_{s,d} \times idf_{s,d}$. The main advantage of SF-IDF is that the same synset is used to represent words with identical meaning, thus reducing the ambiguity of terms and taking into account their semantic relatedness.

However, SF-IDF yields a limited understanding of the semantics of news. Therefore, **SF-IDF+** [75] additionally considers the semantic relationships between synsets in order to overcome this drawback. This is achieved by extending a set of synsets $S(s)$ with the concepts connected through semantic relationships with the included synsets, as shown in Eq. (8), where s denotes a synset, $R(s)$ represents the set of relationships of this synset extracted from a semantic lexicon, such as WordNet, and $r(s)$ indicates the corresponding synset according to relationship r.

$$S(s) = s + \bigcup_{r \in R(s)} r(s) \quad (8)$$

Hence, the item's and user's profiles, v and u, are extended according to Eqs. (9) and 10, respectively.

$$v = \{S(s_1^n), S(s_2^n), ..., S(s_p^n)\} \quad (9)$$

$$u = \{S(s_1^u), S(s_2^u), ..., S(s_u^q)\} \quad (10)$$

Furthermore, SF-IDF+ not only uses extended synsets instead of synsets, as it is the case for the SF-IDF model, but also assigns different weights w_r to the relationship r connecting a synet with its semantically related synset, as per Eq. (11).

$$sf - idf{+}_{s,d,r} = sf_{s,d} \times idf_{s,d} \times w_r \quad (11)$$

Nonetheless, a shortcoming of the SF-IDF+ recommendation model is not being able to take into consideration named entities, which are prevalent in news articles. Thus, Capelle et al. [22] proposed **Bing-SF-IDF+**, a method which extends the SF-IDF+ technique with named entity similarities computed using Bing page counts. The main assumption made by the authors is that the likelihood of two entities being similar is directly proportional to the amount of times they cooccur on websites [22]. The Bing-SF-IDF+ similarity score combines two elements, namely the Bing component which measures the similarity between pairs of named entities, and the SF-IDF+ component which computes the similarity between synsets.

The SF-IDF+ profiles and weights are built and calculated according to Eqs. (8)-(11). For the Bing component, new user and item profiles are built using sets of named entities extracted from the text with a named entity recognizer, denoted as follows:

$$v = \{e_1^n, e_2^n, ..., e_k^n\} \quad (12)$$

$$u = \{e_1^u, e_2^u, ..., e_l^u\} \quad (13)$$

where e^n and e^u denote a named entity in the profile of a news article, and of a user, respectively. The total number of named entities in the article's and user's profile is indicated by k, and respectively, l. All possible pairs of named entities from the two profiles are combined into a vector $\mathbf{V}$, as per Eq. (14).

$$\mathbf{V} = (< e_1^n, e_1^u >, ..., < e_k^n, e_l^u >), \forall e^n \in v, e^u \in u. \quad (14)$$

Subsequently, The Bing search engine is used to compute the page count $c(e^n, e^u)$ for each pair (e^n, e^u) of named entities in $\mathbf{V}$, namely how many pages found by querying Bing contain either one, or both of the entities in a pair. A page rank-based Point-Wise Mutual Information (PMI) co-occurrence similarity measure is afterwards used to calculate the difference between the actual and the expected joint probability of the occurrence of a pair of named entities in a query on a web search engine [22]. PMI assumes independence between the two named entities and is based on their marginal probabilities, as illustrated in Eq. (15).

$$sim_{PMI}(e^n, e^u) = \log \frac{\frac{c(e^n,e^u)}{N}}{\frac{c(e^n)}{N} \times \frac{c(e^u)}{N}} \tag{15}$$

where $c(e^n)$ and $c(e^u)$ denote the page counts of the named entities e^n and e^u from the unread news article's and user's profiles, respectively, whereas N represents the total number of web pages indexed by Bing. The average of these PMI scores over all pairs of named entities in $\mathbf{V}$ constitutes the Bing similarity score:

$$sim_{Bing}(\mathbf{V}) = \frac{\sum_{(e^n,e^u)\in\mathbf{V}} sim_{PMI}(e^n, e^u)}{|\mathbf{V}|} \tag{16}$$

The SF-IDF+ similarity $sim_{sf-idf+}(u, v)$, and the Bing similarity $sim_{Bing}(u, v)$ scores, namely the cosine similarity of the user and target news article profiles, are then normalized using a min-max normalization between 0 and 1 in order to ensure compatibility of scores. Lastly, the Bing-SF-IDF+ score is defined as the weighted average of the two components' normalized similarity scores, according to Eq. (18).

$$sim_{Bing-sf-idf+}(u, v) = \tag{17}$$
$$\alpha \times \overline{sim}_{Bing}(u, v) + (1-\alpha) \times \overline{sim}_{sf-idf+}(u, v)$$

where $\overline{sim}_{Bing}(u, v)$ and $\overline{sim}_{sf-idf+}(u, v)$ represent the normalized Bing and SF-IDF+ similarity scores, and α is a weight optimized on the training set.

An approach combining CF-IDF and SF-IDF, which aims to address the ambiguity problem by representing news articles using key concepts, synonyms and synsets from a domain ontology, is represented by the **OF-IDF** method proposed by Ren et al. [86]. In this case, a news article is described in terms of key concepts contained in a financial domain ontology. Additionally, the lexical representation of a concept is disambiguated by enriching it with its corresponding synset retrieved from WordNet. Similar to CF-IDF, the concepts in the article's profiles are weighted using an Ontology Frequency-Inverse Domain Frequency scheme. Thus, the article can be represented as a vector of OF-IDF weights w^n associated with the concepts it contains, namely $\mathbf{v} = (w_1^n, w_2^n, ..., w_p^n)$, where the weights are computed as follows:

$$w_{i,j} = OF_{i,j} \times IDF_i = \frac{n_{i,j}}{max_i n_{i,j}} \times \log \frac{N}{n_i} \tag{18}$$

In Eq. (18), $n_{i,j}$ is the number of occurrences of concept c_i in article j, and $1 \leqslant i \leqslant p$, where p denotes the total number of concepts in j. The user's interests in the read news, can be described by means of a user-concept matrix whose rows denote the read articles, columns indicate the concepts appearing in these articles, and the entries correspond to OF-IDF weights. According to Ren et al. [86], such a user-concept matrix can be modified using relevance feedback in order to capture different interaction patterns between a user and a target article. More specifically, the original OF-IDF weights are adjusted depending on whether the user clicked, read and liked, or read and did not like an article. Under this assumption, the user profile is changed as follows:

$$\mathbf{u} = \sum_m(\alpha + \mathbf{S}^\alpha) + \sum_n(\beta + \mathbf{S}^\beta) + \sum_l(\gamma + \mathbf{S}^\gamma) \tag{19}$$

In Eq. (19), the vectors $\mathbf{S}^\alpha, \mathbf{S}^\beta, \mathbf{S}^\gamma$ represent the m articles clicked, n articles read and liked, and respectively, l news read and not liked by the user. These vectors of OF-IDF weights are modified using parameters α, β, γ, where the first two parameters are positive to illustrate the user's interest in an article, while the last one is negative to capture negative feedback.

5.1.3. Summary

News recommendation techniques based on the vector space model and incorporating side information are summarized in terms of two aspects:

- **Profile representation.** Each model constructs two profiles, one representing the unread target article, and the other characterizing the user's interests, as an aggregation of the articles from his reading history. CF-IDF utilizes concepts extracted from the news and contained in a domain ontology to represent articles. In comparison, models such as SF-IDF or OF-IDF, use synsets of terms or concepts enriched with associated synonym sets from semantic lexicons to avoid ambiguity. Another approach, used by SF-IDF+, additionally takes into account relationships between synsets, by extending the original vector representation with concepts referred to by semantic relations characterizing the synsets from the initial profile. Lastly, Bing-SF-IDF+ further improves the technique by including named entities into the vector representations.

- **Weighting scheme.** The models discussed in this section employ a variant of the TF-IDF weighting scheme, modified to incorporate concepts or synsets instead of terms. SF-IDF+ refines the weighting model by assigning different weights to each semantic relation connecting a concept to its semantically related synset. In addition to using SF-IDF+ weights to measure the importance of concepts in a news article, the Bing-SF-IDF+ model computes Bing similarity scores for the user and item profiles based on the page rank-based PMI co-occurrence measure of the named entity pairs contained in the two profiles.

5.2. Methods based on semantic similarities

Models from this group most often adopt a content-based approach for computing recommendations in an item-ranking manner. The similarity between articles, and the preference of a user for a candidate article are determined using semantic similarity metrics.

5.2.1. Overall framework

Similar to recommendation models from the previous section, these techniques rely on vector representations of the articles and the user's reading history. However, the models differ in three major aspects:

- **Profile representation.** Several elements extracted from the text of news are used to construct the vector representation of articles and users.
- **Weighting scheme.** Similar to the previous category of recommendation techniques, the weighting scheme utilized determines how the importance of the vector elements is computed.
- **News-user similarity.** The recommendation is based on the similarity of target articles to the articles from the user's profile, which is calculated using a variety of methods.

5.2.2. Representative models

Several representative recommendation approaches are discussed in the current subsection. Cantandor et al. [14, 16] developed a **semantic context-aware recommendation** model which aims to contextualise the users' interests, such that the model learns to ignore preferences that are out of focus in a particular session, and to place a higher importance on those that are in the semantic scope of the ongoing user activity. The profiles of both user and news articles are described using semantic concepts from a domain ontology, as $\mathbf{u} = (w_1^u, w_2^u, ..., w_q^u)$, and $\mathbf{v} = (w_1^n, w_2^n, ..., w_p^n)$, respectively. The concepts in the user's vector representation are weighted with weights $w_i^u \in [-1, 1]$, which indicate the intensity of the user's interest for concept $c_i \in \mathcal{O}$. A negative weight is equivalent to a dislike for the concept, while a positive value shows that the user is interested in the given concept. Similarly, concepts weights $w_i^n \in [0, 1]$ place the article's representation in the same vector space as the user's preferences [14].

A personalised content retrieval approach assigns a relevance measure $pref(v, u) = cos(\mathbf{v}, \mathbf{u})$ of an item v to user u using the cosine similarity between their vector representations. However, a good recommendation model should be able to differentiate between a user's short and long-term preferences, which could be accomplished by enhancing it with contextualized semantic preferences. More specifically, Cantandor et al. [14] define a semantic runtime context as the background topics $\mathbf{C}_u^t$ under which user u performs a set of activities in the unit of time t. The runtime context, illustrated in Eq. (20), is comprised of a set of weighted concepts from a domain ontology, collected from the articles accessed by the user during a session.

$$\mathbf{C}_u^t[c_i] = \xi \cdot \mathbf{C}_u^{t-1}[c_i] + (1 - \xi) \cdot Req_u^t[c_i] \qquad (20)$$

where $Req_u^t \in [0, 1]^{|\mathcal{O}|}$ is a vector whose elements indicate the extent to which the concepts c_i are relevant to the user's request at time t, which can be defined in several ways, including a query concept-vector, or an average concept-vector [14]. The decay factor ξ determines the speed with which the importance of a concept c_i fades over time, specifically how many actions should be performed before a concept is no longer considered to be in the current semantic context.

Following the construction of the runtime context, Cantandor et al. [14] introduce a semantic preference spreading strategy which expands the user's initial preferences through semantic paths towards other concepts in the ontology. This contextual activation of user preferences constitutes an approximation of conditional probabilities. According to this formulation, the probability that concept $c_i \in \mathcal{O}$ is of interest for a user is determined by the probability that the concept c_i itself, as well as all other concepts $c_j \in \mathcal{O}$ directly connected to it in the ontology, belong to the same topic, and the probability that the related concept c_j is also relevant for the user.

Consequently, the semantic spreading mechanism requires weighting every semantic relation r in the ontology with a value $w(r, c_i, c_j)$ which denotes the probability that concepts c_i and c_j belong to the same topic given the fact that they are connected by relation r. The initial set of user preferences expressed in terms of concepts, $\mathbf{P}_u = \{c_i^u \in \mathcal{O} | w_k^u \neq 0\}$, is expanded as follows:

$$\mathbf{EP}_u[c_j] = \begin{cases} \mathbf{P}_u[c_j], \text{if } \mathbf{P}_u[c_j] > 0 \\ R(\{\mathbf{EP}_u[c_i] \cdot power(c_i)\}_{c_i \in \mathcal{O}, r(c_i, c_j)}), \text{otherwise} \end{cases} \tag{21}$$

where $power(c_i) \in [0, 1]$ represents a propagation power assigned to each concept and $R(\mathbf{X}) = \sum_{S \subset \mathbb{N}_n} \{(-1)^{|S|+1} \times \prod_{i \in S} x_i\}$, with $\mathbf{X} = \{x_i\}_{i=0}^n$, $x_i \in [0, 1]$.

The context-aware personalized recommendation model computes the relevance measure of an item v for user u using the expanded profiles of the user and the article, in the following way: $pref_c(v, u) = \lambda \cdot pref(v, EP_u) + (1 - \lambda) \cdot pref(v, EC_u)$. In this case, the parameter $\lambda \in [0, 1]$ weights the strength of the personalization component with regards to the current context.

The weights spreading strategy addresses both the cold-start and the data sparsity problems, while incorporating contextual information captures the changing utility of a news article to a user based on temporary circumstances. While this model applies to single users, in a later work, Cantandor et al. [17] employ a **hybrid context-aware recommendation** technique which exploits the connections between users and concepts to discover relations among users in a collaborative fashion. The goal in this case is to leverage partial similarities between users with similar preferences in a focused domain, but who are globally dissimilar. On a high level, the strategy is accomplished by clustering users according to layers of preferences shared among them. Hence, the user similarities depend on sub-profiles, which increase the likelihood of extracting conjunctions of rare preferences.

Semantic Communities of Interest (CoI) are derived from the users' relations at different semantic levels [17]. More specifically, each ontology concept c_i^u occurring in a user's reading history is represented as a vector of weights measuring its importance for the user, namely $\mathbf{c}_i^u = (w_{1,i}, w_{2,i}, ..., w_{M,i}) \in [-1, 1]^M$. A hierarchical clustering method is used to determine groups of preferences in the concept-user vector space, and each user is assigned to a concept cluster based on the similarity of his profile to cluster C_q, computed as $sim(u, C_q) = \frac{\sum_{c_i^u \in C_q} w_1^u}{|C_q|}$, where c_i is the concept associated with the w_i^u element in the user's preference vector.

Cantandor et al. [17] propose two recommendation models that utilize the extracted latent communities of interests among users. On the one hand, model *UP* computes a unique ranked list of news articles based on the similarities between news and all semantic clusters, meaning that it compares a user's interests to those of the other users and utilizes these user-user similarities to weight preferences for candidate articles. As such, the preference score of article v to user u is computed using Eq. (22):

$$pref(u, v) = \tag{22}$$

$$\sum_q nsim(v, C_q) \textstyle\sum_{y \neq u} nsim_q(u, y) \cdot sim_q(y, v)$$

Here $sim(v, C_q) = \frac{\sum_{c_i \in C_q} w_i^n}{\mathbf{v}\sqrt{|C_q|}}$ represents the similarity between item v and cluster C_q, and $nsim(v, C_q)$ denotes the normalised similarity over the set of all clusters. Moreover $sim_q(u, y)$ and $nsim_q(u, y)$ are the single and normalised similarities at layer q between users u and y, defined as the cosine similarity of the projections of their corresponding concept vectors onto cluster C_q. Therefore, model *UP* takes into account both the characteristics of news articles, as well as the relations between user, at different semantic layers.

On the other hand, model *UP-q* generates recommendations separately for each layer by computing a ranked list for each semantic cluster. The preference between user and target article is calculated as follows:

$$pref(u, v) = \sum_{y \neq u} nsim_q(u, y) \cdot sim_q(y, v) \tag{23}$$

The recommendations corresponding to the cluster to which the user has the highest similarity will be suggested (q maximizes $sim(v, C_q)$ in Eq. (23)).

The same context-aware and multi-facet, group-oriented hybrid recommendations are also adopted by Cantandor et al. [18] to generate **social tags enriched recommendations**. The authors expand the original user profiles with personal tag clouds collected from two websites (Flickr and del.icio.us). The extracted tags are incorporated into the ontological user profiles by mapping them to ontology concepts.

The **semantic relatedness** model of Getahun et al. [41] compares two articles v_i and v_j using the cosine similarity of their vector representations $\mathbf{v}_l$ comprising of concepts from an ontology and their corresponding weights:

$$\mathbf{v}_l = (< c_1^l, w_1^l >, ..., < c_p^l, w_p^l >), l \in \{i, j\} \qquad (24)$$

In comparison to item profiles of models such as CF-IDF (Eq. (4)), in Eq. (24) the total number of concepts appearing in an article's profile is represented by the number of distinct concepts $p = |CS_i \cup CS_j|$ in the sets denoting the two texts, CS_i and CS_j, respectively. The weight w_i of concept c_i is based on its occurrence in the set of concepts CS_j of the other text. More specifically, if c_i is contained in CS_j, then it receives a weight of $w_i = 1$, otherwise its weight is determined by its maximum enclosure similarity to concept c_j. Mathematically, this condition is expressed as follows:

$$w_i = \begin{cases} 1, \text{if } freq(c_i \text{ in } CS_j) > 0 \\ \max_j(ES(c_i, c_j)), \text{otherwise} \end{cases} \qquad (25)$$

where

$$ES(c_i, c_j) = \frac{|N(c_i) \cap N(c_j)|}{|N(c_i)|} \qquad (26)$$

The advantage of this method is that it takes into account related concepts of a concept appearing in news, by utilizing its global semantic neighbourhood.

The Ranked Semantic Recommendation (**RSR**) [54] model is based on the assumption that reading an article containing a certain concept expands the user's knowledge not only in that particular concept, but also in the concepts related to it. This notion is captured by assigning a rank to each concept from an ontology. For example, a user reading news about a concept represented by the class instance *Robinhood* might also be interested in his CEO *Vladimir Tenev* or in the *GameStop short squeeze* event. Since these instances are in a direct relation to *Robinhood*, the ranks of all three should be increased. Similarly, if a user firstly reads an article containing instances *Robinhood* and *Elon Musk*, then accesses news about *Open AI*, a related concept instance to *Elon Musk*, but not to *Robinhood*, the rank of *Elon Musk* should be increased, while that of *Robinhood* should be decreased. Therefore, the rank of a concept aims to account for the user's changing interests.

Each concept c_i is associated with a set of related concepts $r(c_i) = \{c_1^i, c_2^i, ..., c_k^i\}$, and the union of all concepts related to those in the user profile can be expressed as $R = \bigcup_{c_i^u \in u} r(c_i^u)$. Hence, the extended user profile u_R is obtained by extending the initial set of concepts extracted from the previously read articles with the set of related concepts, namely $u_R = u \bigcup R$.

Another assumption underlying RSR [54] is that the more articles containing concept c_i^u a user reads, the higher his interest in that concept. The weight w_i^u of concept c_i^u is thus defined as the number of articles from the user's reading history that contain the concept. RSR utilizes a rank matrix - rows contain concepts from the initial user profile and columns denote the concepts in his extended profile - to model the interaction between concepts and compute their importance for the user. The rank of a concept $c_{i,j}^u$ from this matrix is obtained by weighting w_i^u using an experimentally determined constant value meant capture the type of relationship between concepts:

$$r_{ij} = w_i^u \times \begin{cases} +1.0 \text{ if } e_j = c_i^u \\ +0.5 \text{ if } e_j \neq c_i^u, e_j \in r(c_i^u) \\ -0.1 \text{ otherwise} \end{cases} \qquad (27)$$

The final rank of every concept in the user's extended profile, denoted $Rank(e_j) = \sum_{i=1}^{q} r_{ij}$, is computed as a sum of the values in the corresponding column in the rank matrix and stored in a vector $\mathbf{v}_u$.

A min-max normalization is applied to the extended user profile to ensure that the ranks are in the range $[0, 1]$, and thus, comparable between the user's and article's profiles. A news article, comprised from a set of concepts, is also represented as a vector of concept ranks $\mathbf{v}_v$, where a concept contained in the user's extended profile and appearing in the unread article is assigned the same rank as in $\mathbf{v}_u$, while one not occurring in the target item has a rank of zero. Lastly, the extent to which an article is relevant to a user is computed as the ratio between the sum of concept ranks from the article's representation and the sum of concepts ranks in the user's profile:

$$sim(v,u) = \frac{\sum_{r_v \in \mathbf{v}_v} r_v}{\sum_{r_u \in \mathbf{v}_u} r_u} \tag{28}$$

The Ranked Semantic Recommendation 2 (**RSR 2**) [37] model improves RSR by considering, in addition to the concepts appearing in the unread news articles, also the concepts related to them. Following the previous example, this means that if a candidate news contains the concept instance *Elon Musk*, the model will also utilize related concept instances such as *Open AI*, *SpaceX*, *Tesla, Inc.*, or *Neuralink* to represent the article. Thus, the original article profile is extended by the set of related concepts, namely $v_E = v \bigcup E$, where $E = \bigcup_{c_i^n \in v} r(c_i^n)$.

Another difference to RSR is that RSR2 uses different weight values to determine the concepts' ranks. The rank of a concept in the extended article representation $\mathbf{v}_{v_E}$ is equivalent to the corresponding concept rank from the extended user profile $\mathbf{v}_u$, if it appears in it or is related to one of its concepts. Otherwise, a concept has a rank of zero. The final similarity measure between the extended article and user profiles is modified to incorporate these changes accordingly.

Analogously to SF-IDF, the Semantic Similarity (**SS**) recommendation model [20] represents a news item using the WordNet synsets of the terms it contains, as shown in Eqs. (6) and (7). Recommendations are generated by comparing the similarity of the synsets in the unread news article to the synsets of all the articles previously read by the user. For this purpose, firstly a vector containing all combinations of synsets from the target article and the union of synsets from the user profile is constructed as follows:

$$V = (< s_1^n, s_1^u >, ..., < s_p^n, s_q^u >) \forall s^n \in v, s^u \in u \tag{29}$$

Furthermore, a subset is created from V for all pairs of synsets sharing the same part-of-speech (POS):

$$W \subseteq V \; \forall (s^n, s^u) \in W : POS(s^n) = POS(s^u) \tag{30}$$

where $POS(s^n)$ and $POS(s^u)$ denote the part-of-speech tag of synset s^n from the item's profile, and synset s^u from the user's profile, respectively.

The final similarity score of an unread article is given by the sum of all combinations' similarity rank $sim(s^n, s^u)$ relative to the total number of combinations $|W|$, illustrated as follows:

$$sim_{SS} = \frac{\sum_{(s^n,s^u) \in W} sim(s^n, s^u)}{|W|} \tag{31}$$

The WordNet taxonomy constitutes a hierarchy of "is-a" relationships between its nodes which, in turn, constitute synsets. As such, Capelle et al. [20] propose five semantic similarity measures to calculate the similarity rank $sim(s^n, s^u)$ for each combination of synsets in W, namely the extent to which two synsets are semantically close. Three of the measures (Jiang and Conrath [57] $sim_{J\&R}$, Resnik [88] sim_R, Lin [68] sim_L) utilize the information content of a node, defined as $IC(s) = -\log \sum_{w \in S} p(w)$. More specifically, this metric can be described as the negative logarithm of the sum of all probabilities of all the words w from synset s. Furthermore, they take into account the lowest common subsumer (LCS) between two nodes, which represents the lowest node dominating the pair [88]. The three metrics are illustrated in Eqs. (32)-(34).

$$sim_{J\&C}(s^n, s^u) = \frac{1}{dist_{J\&C}(s^n,s^u)} \tag{32}$$

$$= \frac{1}{IC(S^n)+IC(s^u)-2\times LCS(s^n,s^u)}$$

$$sim_R(s^n, s^u) = IC(LCS(s^n, s^u)) \tag{33}$$

$$sim_L(s^n, s^u) = \frac{2\times \log p(LCS(s^n,s^u))}{\log p(s^n)+\log p(s^u)} \tag{34}$$

The two remaining metrics, of Leacock and Chodorow [61] $sim_{L\&C}$, and of Wu and Palmer [113] $sim_{W\&P}$, shown in Eqs. (35)-(36), define the similarity based on the path length between nodes. The path length can refer to either the shortest path (denoted *length*) between a pair of nodes or the maximum depth (denoted as D) from the lowest to the top node in the hierarchy.

$$sim_{L\&C}(s^n, s^u) = -\log \frac{length(s^n,s^u)}{2D} \tag{35}$$

$$sim_{W\&P}(s^n, s^u) = \frac{2\times depth(LCS(s^n,s^u))}{length(s^n,s^u)+2\times depth(LCS(s^n,s^u))} \tag{36}$$

Similar to Bing-SF-IDF+, **BingSS** [21] extends the semantic lexicon-driven SS recommendation model by taking into account named entities. The semantic similarity formula from Eq. (31) is modified to take into account only the set of synset pairs $TOP_W^{\beta_{SS}}$ with the highest similarity in W, as follows:

$$sim_{SS} = \frac{\sum_{(s^n,s^u)\in W} sim_{SS}(s^n, s^u) \in TOP_W^{\beta_{SS}}}{|TOP_W^{\beta_{SS}}|} \qquad (37)$$

where β_{SS} constitutes a predefined positive integer, optimized on the test set, which indicates the top-β_{SS} similarities from the pairs of synsets in W. This change is implemented to reflect the assumption that not all named entities occurring in an article are equally relevant for determining the user's interests. For example, for news regarding the stock exchange changes of GameStop, the named entity *New York Stock Exchange* is less relevant for a user interested specifically in GameStop. The BingSS similarity measure introduced in Eq. (16) is modified accordingly to take into account this assumption, as illustrated in Eq. (38).

$$sim_{Bing} = \frac{\sum_{(e^n,e^u)\in \mathbf{V}} sim_{PMI}(e^n, e^u) \in TOP_V^{\beta_{Bing}}}{|TOP_V^{\beta_{Bing}}|} \qquad (38)$$

where $TOP_V^{\beta_{Bing}}$ represents the set of top-β_{Bing} entity pairs with the highest similarity in V (see Eq. 14), and β_{Bing} constitutes a predefined positive integer denoting the top-β_{Bing} similarities from pairs in the set V.

Lastly, the Bing and the SS components are combined in the final BingSS similarity score using a weighted average with predefined weight α:

$$sim_{BingSS} = \alpha \times sim_{Bing} + (1 - \alpha) \times sim_{SS} \qquad (39)$$

5.2.3. Summary

This subsection summarizes the recommendation approaches based on semantic similarities by considering three factors:

- **Profile representation.** Models in this category employ two approaches for representing news and the user's preferences. On the one hand, the semantic context-aware recommendation framework, its later hybrid enhancements, RSR and RSR2 utilize ontology concepts appearing in articles, as well as concepts related to them. On the other hand, SS represents items and users in terms of WordNet synsets, while BingSS additionally considers named entities.
- **Weighting scheme.** The weighting schemes employed to represent elements in the vector representations of articles and users vary in between models. Semantic context-aware techniques use weights in the range $[-1, 1]$ to denote the users' likes and dislikes, while SS-based methods assign weights based on the information content of nodes or the lengths of paths between pairs of nodes in a semantic lexicon. Moreover, the semantic relatedness model defines concept weights in terms of semantic enclosure which considers the global neighbourhood of a concept. The RSR frameworks compute ranks for each ontology concept based on the number of articles containing them and read by the user, as well as on how the concepts are related to each other in the user's reading history.
- **News-user similarity.** Cosine similarity is often employed to determine the preference of a user for an unread news article in context-aware models. Hybrid semantic context-aware models use a weighted combination of cluster-based cosine similarities to determine the news-user similarity. In contrast, RSR-based models compute the article relevance as the ratio of the sum of concepts ranks from the item and user profiles.

5.3. Methods based on distance

Recommenders based on distance take into account not only the semantic relatedness of concepts or entities, but also their background hierarchical structures which indicate how close the concepts or entities are situated in the ontology or knowledge graph.

5.3.1. Overall framework

The majority of methods in this category represent a news article as a set of tuples consisting of the concepts contained in an ontology and their corresponding weights. Formally, this can be written as $v = \{< c_1^n, w_1^n >, ..., < c_p^n, w_p^n >\}$, where $c_i^n \in \mathcal{O}$, w_i^n is the weight of concept $c_i^n (1 \leqslant i \leqslant p)$, and p is the total number of concepts found in the article v. The profile of a user u is constructed by accumulating all the concepts that appear in the articles previously read by the user, denoted as $u = \{< c_1^u, w_1^u >, ..., < c_q^u, w_q^u >\}$, where w_j^u is the average weighting of concept c_j^u in the articles from the user's reading history that contain concept c_j^u, and q denotes the number of concepts in the articles read by the user. The recommendations are computed in a content-based, item-ranking fashion. Two aspects distinguish the distance measure-based frameworks:

- **Weighting scheme.** The concepts comprising the user and news profiles are weighted using different strategies to measure their importance.
- **Item-item similarity.** The similarity between two news articles is computed using several distance measures.

5.3.2. Representative models

In the following, four representative recommendation techniques for this category are investigated.

ePaper[73, 94] weighs the ontology concepts denoting the user's interests according to the user's implicit feedback. More specifically, the weight of a concept c_i^u is given by the number of clicks on articles containing the given concept relative to the total number of clicks in the user's profile. The relevance of an item to a user is defined in terms of the hierarchical distance between the concepts from the associated profiles, which takes into account the amount of common and related concepts included in each profile, as well as the distance between them. Based on a 3-level ontology, ePaper relies on 1-hop (parent-child), and 2-hop (grandparent-grandchild) hierarchical relations between concepts [73]. The relative position of related concepts from the user's and the article's profiles denotes their relationship in terms of specificity.

Three types of partial matches between concepts were defined by Maidel et al. [73] based on hierarchical distance. A *perfect match* is obtained if the same concept appears in both profiles and at the same hierarchical level. For example, both the news and the user profile contain the concept 'artificial intelligence', found at level 1 in the ontology. However, if a concept occurs only in one of the profiles, while its parent or child is included in the other profile, a *close match* is reached. In this case, one can further differentiate between cases when the user's concept (e.g. *artificial intelligence*) is more general than the article's concept (e.g. *deep learning*), and those in which the user's interest is more specific (e.g. user concept is *graph neural networks* and item concept is *deep learning*). Lastly, a *weak match* occurs if the concepts from the two profiles are two levels apart in the hierarchy, such as the user being interested in *graph neural networks*, whereas the article contains the concept *artificial intelligence* . Analogous to the previous match type, two cases are determined by the profile containing the more general concept.

A similarity score S_i assigns different weights based on the type of match of concept c_1^w to the corresponding concepts in the user's profile. Lastly, the Item Similarity (IS) score, shown in Eq. (40), determines how similar the target article is to the user's interests, based on the number of concept matches (given by S_i) and the concepts' weights from the user profile, given by the number of clicks N on the items containing the concept [94].

$$IS = \frac{\sum_{i=1}^{p} N_i \cdot S_i}{\sum_{j=1}^{q} N_j} \tag{40}$$

A different approach is adopted in **Magellan** [31], which uses a Weighted Term Frequency scheme to determine the importance of a candidate news article to a monitored domain. Magellan extracts named entities from news to represent the articles, and operates on their corresponding concepts from an ontology. According to the weighting scheme, the importance of concepts is determined by their centrality and prestige in the ontology. The main assumption underlying the measure of centrality is that the more relations a concept has to other concepts, the higher is its importance in the given domain. Hence, the concept with the highest out-degree, namely the largest number of accumulative out-going connections, is considered the top-ranked individual. Subsequently, the importance of the remaining concepts depends on the distance, measured in the number of hops, and the strength of the relations w_r to the top-ranked concept, as given by the centrality weight $w_{centrality} = \frac{1}{hops} \times \frac{w_r}{hops}$.

The centrality score ensures that concepts with shorter and stronger connections to the top-ranked concept will be assigned a higher importance than those situated further away in the ontology or having weaker relations. The centrality weight is complemented by the prestige of a concept in the ontology, a method which ranks the concepts based on their incoming relations. The more a concept is referred to via different relations by another concept (i.e. the larger its in-degree), the higher its prestige in the ontology. Consequently, the final importance score of a concept is computed as the product of centrality and prestige (denoted as *rank* in Eq. (41)), weighted by a constant value α assigned to the top-ranked concept:

$$w_{importance} = \alpha \times \frac{w_{centrality}}{rank} \tag{41}$$

The final weight w_i of concept i is obtained by combining its importance in the ontology and frequency n_i in the news article, $w = w_{importance} \times n_i$. According to this weighting scheme, Magellan will score higher stories which frequently contain entities with a large importance in the ontology, whereas those which either contain few concepts or only named entities with low importance will be assigned a lower score.

OBSM, the ontology based similarity model proposed by Rao et al. [85], utilizes a TF-IDF weighting scheme for the concepts in the user and news profiles. The similarity between two concepts c_1 and c_2 found in the news depends on their ontological structures, represented in terms of the shortest distance d among concepts in the ontology, the shortest distance δ to their common ancestor closest to the root node, and the height H of the ontology. This concept-concept similarity metric, illustrated in Eq. (42), follows the assumption that two adjacent, more concrete concepts situated at a lower level in the ontology share more common information from their ancestors, and thus, have a higher likelihood to be similar than those found at a higher level. The preference for closer concepts is ensured by the term $(-log_{2H}\frac{d}{2H})$, which is negatively correlated with the concept distance d. In turn, the weight $\frac{e^{\delta}}{e^{\delta}+1}$ will assign higher importance to concepts located at deeper levels in the hierarchy.

$$Csim(c_1, c_2) = \begin{cases} 1, d = 0 \text{ or } isSynonyms(c_1, c_2) \\ \frac{e^{\delta}}{e^{\delta}+1} \cdot (-log_{2H}\frac{d}{2H}), \text{otherwise} \end{cases} \tag{42}$$

The similarity between the profile of a target news article and user is computed in the following way [85]:.

$$sim(u, v) = \frac{1}{p}\sum_{i=1}^{p} \max_{1 \leqslant j \leqslant q}\{Csim(c_i^n, c_j^u) \times w_{i,j}\} \tag{43}$$

where

$$w_{i,j} = \frac{2}{1 + e^{k\tau}}, \text{ with } \tau = \frac{abs(w_i^n, w_j^u)}{max(w_i^n, w_j^u)} \tag{44}$$

According to Eq. (44), two concepts c_i^n and c_j^u whose corresponding weights w_i^n and w_j^u are relatively equal, will result in a higher confidence score $w_{i,j}$. In turn, this means that the two concepts are similarly important in their concepts sets, indicating that the target article might be of interest for the given user. Concepts with different weights in their associated sets are penalized using a smoothing factor k which controls the sensitivity of the confidence function.

In contrast to the previous two models, **SED**, the entity shortest distance over knowledge graphs algorithm proposed by Joseph and Jiang [58], defines item-item similarity as the shortest distance between the subgraphs consisting of named entities extracted from news articles. The approach is threefold [58]. Firstly, all named entities contained in every news article are extracted and linked to the corresponding nodes in a knowledge graph. Secondly, in the subgraph generation phase, each news article is represented as a subgraph containing the linked nodes from the knowledge graph associated with the previously extracted named entities. These subgraphs are expanded with outgoing relations from the L-hop neighbourhood of each node discovered using a breadth first search strategy.

The shortest distance between two entities over the knowledge graph represents the shortest path length between the corresponding nodes, mathematically denoted as $D(e_i, e_j) = min(|p_k|)$, where $|p_k|$ is the length of path k from the set of all paths between the entity pair (e_i, e_j), namely $p_k \in \mathcal{P}(e_i, e_j)$. Based on this definition, the shortest distance between two articles' subgraphs, S_1 and S_2, is computed according to Eq. (45).

$$D(S_1, S_2) = \frac{\sum_{e_i \in S_1} \min_{e_j \in S_2} D(e_i, e_j)}{|S_1|} \tag{45}$$

Lastly, the similarity between two articles is computed as the pair-wise shortest distance over the union of their subgraphs[58], as shown in Eq. (46).

$$\hat{D}(S_1, S_2) = \frac{D(S_1, S_2) + D(S_2, S_1)}{2} \tag{46}$$

This method provides a symmetric average minimum row-wise distance which places a higher importance on the entity pairs with the highest likelihood of co-occurrence in news article. Additionally, a weighted shortest distance between the articles could be used by weighting the edges of the subgraphs and computing the sum of all the weights of the traversed edges [58]. For the weighted SED algorithm, different weighting schemes could be utilized, including the relation weighting scheme, which assigns edge weights based on the amount of shared neighbours of two entity nodes from an article.

5.3.3. Summary

Knowledge-aware news recommender systems based on distance are summarized from two perspectives.

- **Weighting scheme.** Concepts in the item and user profiles are weighted to encode their importance. However, there is not one unique weighting scheme employed by all the models in this category. ePaper weights concepts based on the number of user clicks on articles containing them, while OBSM utilizes classic TF-IDF weights. Magellan weights concepts based on their importance in an ontology, computed using social network measures and their frequency in news articles. In contrast, SED does not represent the user or item profile in terms of concept sets, but as subgraphs of named entities from a knowledge graph.
- **Item-item similarity.** The majority of models previously discussed utilize a type of distance measure to directly calculate the similarity between two news articles. On the one hand, methods such as ePaper or OBSM focus on the hierarchical distance between the concepts contained in the items' profile. On the other hand, SED views article similarity as the degree to which the subgraphs representing news articles overlap. In comparison, Magellan uses a combination of distance measure and term frequency to determine the importance of named entities from news articles and corresponding ontology concepts to a domain, and to rank candidate articles accordingly.

5.4. Methods based on knowledge graph embeddings

In recent years, the rapid advancements in the field of deep learning have also led to a paradigm shift in the domain of news recommendation. State-of-the-art knowledge-aware recommendation models combine textual representation with external information contained in knowledge graphs, encoded by means of knowledge graph embeddings, defined below.

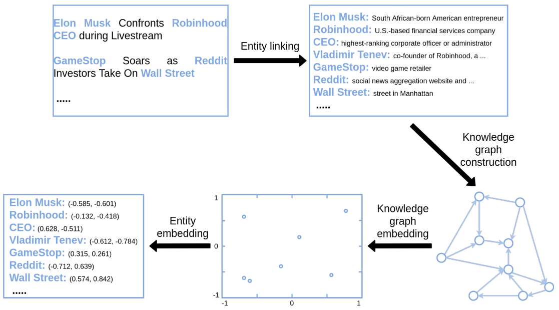
Elon Musk Confronts Robinhood CEO during Livestream
GameStop Soars as Reddit Investors Take On Wall Street
.....
Entity linking
Elon Musk: South African-born American entrepreneur
Robinhood: U.S.-based financial services company
CEO: highest-ranking corporate officer or administrator
Vladimir Tenev: co-founder of Robinhood, a ...
GameStop: video game retailer
Reddit: social news aggregation website and ...
Wall Street: street in Manhattan
.....
Knowledge graph construction
Knowledge graph embedding
1
0
-1
-1
0
1
Entity embedding
Elon Musk: (-0.585, -0.601)
Robinhood: (-0.132, -0.418)
CEO: (0.628, -0.511)
Vladimir Tenev: (-0.612, -0.784)
GameStop: (0.315, 0.261)
Reddit: (-0.712, 0.639)
Wall Street: (0.574, 0.842)
.....

Fig. 3. Illustration of the knowledge distillation process used by KGE-based recommendation models (reproduced from [102]).

Definition 4. Given a dimensionality $k << n$, the goal of **knowledge graph embedding** (KGE) is to project a knowledge graph $\mathcal{G} = (V, E)$ into a low-dimensional space, by learning k-dimensional representations for all entities and relations in $\mathcal{G}$, which preserve the structural information of the original graph [13, 102].

5.4.1. Overall framework

Frameworks classified in this category generally use a knowledge distillation process to incorporate side information in their recommendations. Firstly, named entities are extracted from news articles using a named entity recognizer. Secondly, these are connected to their corresponding nodes in a knowledge graph using an entity linking mechanism. Thirdly, one or multiple subgraphs are constructed using the linked entities, their relations, and neighbours from the knowledge graph. Afterwards, the obtained graphs are projected into a continuous, lower-dimensional space to compute a representation for their nodes and edges. Thus, these models utilize both the structural and semantic information encoded in knowledge graphs to represent news. Figure 3 exemplifies this process.

In contrast to models from the previous categories, KGE-based recommenders predict the probability that a user will click on a target article, namely the click-through rate. Several factors underlying these recommendation models should be considered:

- **Recommendation model input.** Usually the input to the recommendation model consists of an unread news article and the user's reading history. However, various elements, including textual information and knowledge entities can be combined to represent users and items.

- **Knowledge graph embedding model.** Several models can be utilized to compute node embeddings for the knowledge graph entities.
- **Components of recommender system.** The systems' architecture consists of multiple deep-learning models, each aiming to capture different aspects characterizing the news items, user's preferences, and interactions among users and news.
- **Aggregation of knowledge-level and text-level components.** Another distinguishing factor is constituted by the way in which the outputs of different components of the recommendation model are aggregated to predict the click-trough probability for a candidate article.

5.4.2. Representative models

The architectures of 10 KGE-based news recommendation frameworks are investigated in this section.

The Collaborative Entity Topic Ranking (**CETR**) [117] model combines matrix factorization, topic modelling and knowledge graph embeddings in a collaborative fashion to alleviate the data sparsity problem and the limitations of word-level topic models on very infrequent words appearing in news articles. The model joins together three modules, the first modelling the user's reading behaviour, the second performing entity-level topic analysis of news, and the last computing representations of the knowledge graph entities.

The user behaviour component takes as input the user-news interaction matrix **Y**, defined as follows:

Definition 5. The **user-item interaction matrix** $\mathbf{Y} = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\} \in \mathbb{R}^{M \times N}$ is defined according to the user's implicit feedback y_{uv}, where:

$$y_{uv} = \begin{cases} 1 & \text{if user } u \text{ interacted with item } v \\ 0 & \text{otherwise} \end{cases} \quad (47)$$

The user-item interaction matrix is is factorized into a matrix **U** of user features and a matrix **V** of news latent features. The factorization method, a Bayesian Personalized Ranking model [87], uses a sigmoid function to characterize the probability of observing a triplet (u, v, v') given the user and news matrices. Such a triplet denotes the scenario in which a user u has read article v, but not v'. The two feature matrices are learned with a maximum likelihood function applied over all triplets in the user's profile.

In the following step, topic analysis is conducted at the entity level, where entities belonging to the same topic are sampled from a Gaussian distribution [117]. The third module learns knowledge graph embeddings with the TransR model [69]. The probability of observing a quadruple (h, r, t, t'), denoting the head entity h being connected to tail entity t, but not to t', by relation r, is defined similarly to BPR. The three components are jointly trained by calculating the log likelihood of seeing all triplets, entities, and quadruplets, given the user and news feature matrices, the distribution of topics over entities, and the embeddings of entities and relations from the knowledge graph.

DKN, the deep knowledge-aware network proposed by Wang et al. [102], was the first architecture to fuse neural network-based text-level and knowledge-level representations of news using an attention module. The input to the recommendation model is constituted by the user's click history and one candidate news article. Each article t is represented by its title. In turn, the article's title is composed of a sequence of words, $t = [w_1, w_2, ..., w_N]$, and every word w might correspond to a entity e in a knowledge graph [102]. The enrichment of textual information with external knowledge follows the knowledge distillation process from Figure 3. Wang et al. [102] use not only direct knowledge graph correspondents of identified named entities to construct the subgraph, but also their one-hop neighbours to reduce sparsity and increase diversity among the extracted entities. This knowledge-level representation of news is further enhanced by taking into account the context of an entity $context(e)$, to increase the identifiability of entities after computing their embeddings.

Definition 6. The **context of an entity** e is defined as the set of its immediate neighbours in the knowledge graph [102]:

$$context(e) = \{e_i | (e, r, e_i) \in \mathcal{G} \vee (e_i, r, e) \in \mathcal{G}\} \quad (48)$$

The inner circle in Figure 4 exemplifies this concept. DKN takes as input the embedding of *GameStop short squeeze* to represent the entity, as well as its context, denoted by neighbours and associated relations, such as *USA* (country), or *Elon Musk*, *Robinhood*, *r/WallStreetBets* (participant).

One of the input elements to the recommendation model is constituted by the embedding of an entity's context, defined in the following manner:

Definition 7. The **context embedding** of entity e is defined as the average of the embeddings of its contextual entities [102]:

$$\bar{\mathbf{e}} = \frac{1}{|context(e)|} \sum_{e_i \in context(e)} \mathbf{e}_i \tag{49}$$

The first level in DKN's architecture is represented by a knowledge-aware convolutional neural network (KCNN), namely the CNN framework proposed by Kim [60] for sentence representation learning extended to incorporate symbolic knowledge in the text representations. Firstly, the entity embeddings $\mathbf{e}_i \in \mathbb{R}^{k \times 1}$ and the context embeddings $\bar{\mathbf{e}}_i \in \mathbb{R}^{k \times 1}$, obtained with TransD [56], are projected from the entity to the word vector space, according to Eqs.(50) and (51), using a hyperbolic tangent transformation function g.

$$g(\mathbf{e}_{1:n}) = [g(\mathbf{e}_1)g(\mathbf{e}_2)...g(\mathbf{e}_n)] \tag{50}$$

$$g(\bar{\mathbf{e}}_{1:n}) = [g(\bar{\mathbf{e}}_1)g(\mathbf{e}_{\bar{2}})...g(\bar{\mathbf{e}}_n)] \tag{51}$$

Secondly, the matrices containing word embeddings $\mathbf{w}_{1:n}$ (pre-trained or randomly initialized), transformed entity $g(\mathbf{e}_{1:n})$ and context $g(\bar{\mathbf{e}}_{1:n})$ embeddings are aligned and stacked to obtain a multi-channel input $\mathbf{W} = [[\mathbf{w}_1 g(\mathbf{e}_1)g(\bar{\mathbf{e}}_1)]...[\mathbf{w}_n g(\mathbf{e}_n)g(\bar{\mathbf{e}}_n)]] \in \mathbb{R}^{d \times n \times 3}$.

The word-aligned KCNN applies multiple filters of varying sizes to extract patterns from the titles of news, followed by max-over-time pooling and concatenation of features to obtain the final representation $\mathbf{e}(t)$ of an article. Hence, the KCNN component is able to discover latent knowledge-level connections among news using extracted entities and common sense knowledge embedded in knowledge graphs.

Additionally, DKN employs an attention network to capture the diverse interests of users in different news topics by dynamically aggregating a user's history according to the current candidate article [102]. The second level of the DKN framework concatenates the embeddings of a target news t_j and an article t_k read by the user, feeding the resulting vector into a Deep Neural Network (DNN) $\mathcal{H}$ which computes the impact of the candidate news on the read article. The output of the attention network $\mathcal{H}$ is normalized using a softmax function. This process is illustrated in Eq. (52):

$$\begin{aligned} a_{t_k,t_j} &= softmax(\mathcal{H}(\mathbf{e}(t_k), \mathbf{e}(t_j))) \\ &= \frac{exp(\mathcal{H}(\mathbf{e}(t_k),\mathbf{e}(t_j)))}{\sum_{k=1}^{N} exp(\mathcal{H}(\mathbf{e}(t_k),\mathbf{e}(t_j)))} \end{aligned} \tag{52}$$

Given the normalized attention weights, a user i's embedding with respect to the target article t_j is represented by the weighted sum of the N_i embeddings of article titles from his click history:

$$\mathbf{e}(i) = \sum_{k=1}^{N_i} a_{t^i_k,t_j} \mathbf{e}(t^i_k) \tag{53}$$

Lastly, DKN [102] predicts the click probability of user i for news article t_j with another DNN $\mathcal{G}$ that takes as input the final user embedding from Eq. (53) and the article's embedding, as $p_{i,t_j} = \mathcal{G}(\mathbf{e}(i), \mathbf{e}(t_j))$.

The recommendation model of Gao et al. [38], **fine-grained DKN with self-attention**, learns semantic-level and knowledge-level representations of news by adjusting the DKN architecture to use a fine-grained word-level description of news, obtained with a self-attention mechanism, instead of a topic-level representation given by the KCNN component. The user's click history and a candidate piece of news constitute the model's input. The framework consists of four-level self-attention modules [38]. Firstly, a word-level self-attention component computes the semantic-level and knowledge-level representation of articles using pre-trained embeddings of news tags, and transformed pre-trained embeddings of entities extracted from a knowledge graph and their context, similar to DKN. The attention weight measuring the impact of each word in the news representation is computed as follows:

$$a_1 = softmax(\mathbf{V}_1 tanh(\mathbf{W}_1 \mathbf{w}_i^T + \mathbf{W}'_1 \mathbf{q}_i^T + \mathbf{b}_1)) \tag{54}$$

where the subscripts of the trainable matrices denote the layer of the network and $\mathbf{q}_i$ are queries given by three keywords selected for each article. The word-level representation of news constitutes a weighted sum of its word embeddings $\mathbf{w}'_{1:n} = \sum_{i=1}^{n} a_i^t \mathbf{w}_i$, whereas the entity-level $\mathbf{e}'_{1:n}$ and context-level $\bar{\mathbf{e}}'_{1:n}$ representations are computed in a similar manner.

Secondly, the item-level attention model computes the final representation of news article t_k, according to Eq. (55), as a weighted sum of the different-level embeddings, where the weights are given by corresponding attention coefficients.

$$\mathbf{e}(t_k) = a_{word}\mathbf{h}_{word} + a_{entity}\mathbf{h}_{entity} + a_{context}\mathbf{h}_{context} \tag{55}$$

The attention weights of words are calculated as shown in Eq. (56), while those of entities and context can be computed analogously.

$$a_{word} = softmax(\mathbf{V}_2 tanh(\mathbf{W}_2 \mathbf{h}_{word} + \mathbf{b}_2)) \tag{56}$$

Thirdly, the user-level self-attention module computes the final representation of the user i's history $\mathbf{e}(i)$ with respect to the candidate news t_j as in Eq. (53). However, in contrast to DKN, here the attention weight is computed as follows:

$$a_{t_k,t_j} = softmax(\mathbf{V}_3 tanh(\mathbf{W}_3 \mathbf{e}(t_k)^T + \mathbf{W}'_3 \mathbf{e}(t_j)^T + \mathbf{b}_3)) \tag{57}$$

Fourthly, the vector representation of the user and the target news article are combined using a multi-head attention module [100] with 10 parallel attention layers. Lastly, the output of the fourth module is passed through a fully-connected layer to calculate the user's probability of clicking the candidate article.

A different approach is constituted by **RippleNet** [103], and end-to-end framework that propagates user preferences along the edges of a knowledge graph. RippleNet takes as input a candidate news article and the user's historical set of interests $\mathcal{V}_u$, which act as seeds in the knowledge graph. The main idea underlying the model is that of ripple sets $\mathcal{S}_u^k$, namely sets of knowledge triples situated k-hops away from the seed set $\mathcal{V}_u$. The concepts of relevant entities and ripple sets are defined below.

Definition 8. Giving the knowledge graph $\mathcal{G}$ and the interaction matrix $\mathbf{Y}$, the set of k-hop **relevant entities** for user u is defined as:

$$\mathcal{E}_u^k = \{e_t | (e_h, r, e_t) \in \mathcal{G} \text{ and } e_h \in \mathcal{E}_u^{k-1}\} \tag{58}$$

where $k = 1, 2, ..., H$ and $\mathcal{E}_u^0 = \mathcal{V}_u = \{v | y_{uv} = 1\}$ is the set of user u's past interacted items.

Definition 9. The k-hop **ripple set** of a user u is the set of knowledge triples whose head entities are $(k-1)$-hop relevant entities $\mathcal{E}_u^{k-1}$:

$$\mathcal{S}_u^k = \{(e_h, r, e_t) | (e_h, r, e_t) \in \mathcal{G} \text{ and } e_h \in \mathcal{E}_u^{k-1}\} \tag{59}$$

where $k = 1, 2, ..., H$.

The user's interests in certain entities are extended from the initial set along the edges of the knowledge graph, as shown in Figure 4. The further the hop, the weaker the user's potential preference in the corresponding ripple set becomes, since entities which are too distant from the user's initial interests might introduce noise in the recommendations. This behaviour is exemplified in Figure 4 by the fading colour of the concentric circles denoting ripple sets. The closer a neighbouring entity is to the center seed, the more related the two are assumed to be. In practice, this is controlled by the number H of hops considered [102].

In the first step, RippleNet calculates the probability p_i that a news article is similar, in the space of relation r_i, to a head entity h_i from the user's 1-hop ripple set $\mathcal{S}_u^1$. The relation type accounts for contextual similarities of entities, such as *Elon Musk* and *Vladimir Tenev* being similar when considering that they are both entrepreneurs, but having fewer similarities if only analysing their place of birth. Mathematically, the relevance probability for each triple (h_i, r_i, t_i) in $\mathcal{S}_u^1$ of user u is computed according to Eq. (60) using the embeddings of the item $\mathbf{v} \in \mathbb{R}^d$, the relations $\mathbf{R}_i \in \mathbb{R}^{d \times d}$, and the entity $\mathbf{h}_i \in \mathbb{R}^d$:

$$p_i = softmax(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i) = \frac{exp(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i)}{\sum_{(h,r,t) \in \mathcal{S}_u^1} exp(\mathbf{v}^T \mathbf{R} \mathbf{h})} \tag{60}$$

The 1-order response $\mathbf{o}_u^1$ of user u's history to candidate news v is defined as the sum of the embeddings $\mathbf{t}_i \in \mathbb{R}^d$ of tail entities from $\mathcal{S}_u^1$ weighted by their corresponding relevance probabilities, as follows:

$$\mathbf{o}_u^1 = \sum_{(h_i, r_i, t_i) \in \mathcal{S}_u^1} p_i \mathbf{t}_i \tag{61}$$

Eqs. (60) and (61) theoretically illustrate the preference propagation mechanism of RippleNet, through which the user's interests are spread from the initial set $\mathcal{V}_u$, along the links of $\mathcal{S}_u^1$, to the set of 1-hop relevant entities $\mathcal{E}_u^1$. The preference propagation can be extended H hops away from the initial seed set, by iteratively applying Eq. (61) on the user u's H ripple sets $\mathcal{S}_u^i$. The final user preference distribution with regards to candidate article v is computed by combining the responses of all H orders: $\mathbf{u} = \sum_{i=1}^{H} \mathbf{o}_u^i$. The click-through probability is then calculated using a sigmoid

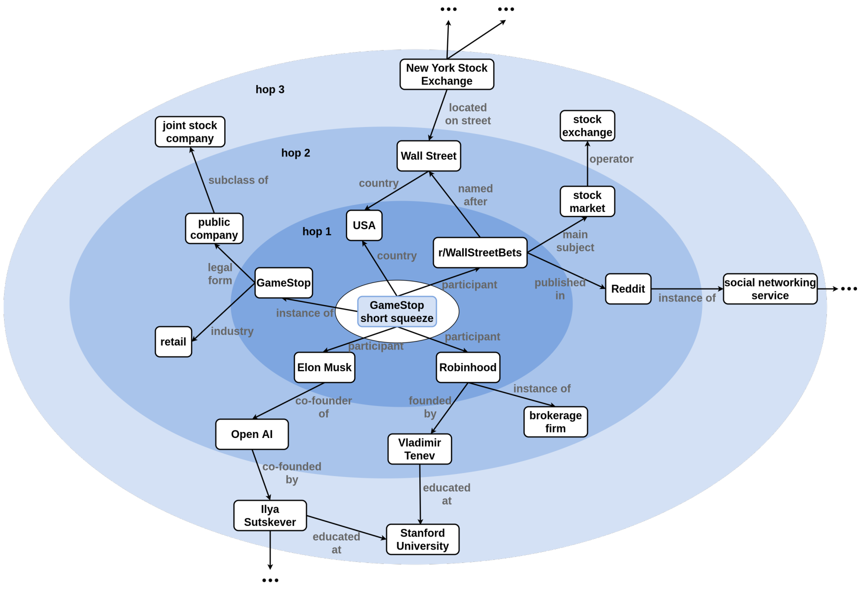

Fig. 4. Illustration of ripple sets of *GameStop short squeeze* in Wikidata. The concentric circles indicate ripple sets with different hops. The fading blue signifies decreasing relatedness between the center and the neighbouring entities (reproduced from [103]).

function applied to the embeddings of the user and the target news. In comparison to the previous methods, RippleNet not only incorporates external knowledge in its recommendations, but also automatically identifies possible explanatory paths connecting news from the user's click history to the candidate article.

An inward aggregation version of this model, denoted **RippleNet-agg**, was later proposed by Wang et al. [104] to extract high-order structural proximity information among entities in a knowledge graph. In comparison to the outward propagation model, this variant uses biases to aggregate and inject ripple sets information in an entity's representation. More specifically, the importance of a relation r_i to a user u is measured using a scoring function $\pi^u_{r_i} = g(\mathbf{u}, \mathbf{r_i})$ applied to the user and relation embeddings. This weight aims to capture the relation-dependent user preferences, such as a reader being interested in technology news that contain the same *entrepreneur* as previously clicked articles, while another being attracted by articles related to the same *significant event*.

In RippleNet-agg, higher-order proximity information is captured by encoding the ripple sets in the final prediction function at the item-end, compared to the user-end, as it was the case in the original RippleNet model. To this end, the topological proximity structure of a news article v is defined as the linear combination of its one-hop samples ripple set $\mathbf{v}^u_{\mathcal{S}(v)} = \sum_{e \in \mathcal{S}(v)} \tilde{\pi}^u_{r_{v,e}} \mathbf{e}$, where $\tilde{\pi}^u_{r_{v,e}} \mathbf{e}$ represents the normalized user-relation score over all neighbouring entities in v's ripple set.

Lastly, the representations of the entity $\mathbf{v}$ and its neighbourhood $\mathbf{v}^u_{\mathcal{S}(v)}$ are aggregated using an aggregation function *agg*, defined as:

$$agg = \sigma(\mathbf{W} \cdot (\mathbf{v} + \mathbf{v}^u_{\mathcal{S}(v)}) + \mathbf{b}) \qquad (62)$$

Although the aggregation function in Eq. (62) is represented by the sum operation followed by a nonlinear transformation σ, this could be replaced by a *concat* aggregation, which would concatenate instead of add the two representations, or a $neighbor - only$ aggregation function, which would only consider the neighbourhood representation.

In contrast to previous models, the Multi-task feature learning approach for Knowledge graph Recommendation (**MKR**) [106] uses the knowledge embedding task to assist the recommendation one. The model is trained in an end-to-end fashion by optimizing the two components alternately, with different frequencies. The two components are connected by cross&compress units to learn high-order interactions between entities in the knowledge graph and items from the recommender systems sharing features in non-task-specific latent spaces. MKR aims to improve generalization of predictions by using a multi-task learning environment.

MKR is comprised of three modules. The recommendation component uses as input two raw feature vectors **u** and **v** of the user and article. The latent features of the user are extracted using an L-layer multilayer perceptron (MLP) as shown in Eq. (63), where $\mathcal{M}$ is a fully-connected neural network layer:

$$\mathbf{u}_L = \mathcal{M}^L(\mathbf{u}) \tag{63}$$

The features of news article v are computed using L cross&compress units, as follows:

$$\mathbf{v}_L = \mathbb{E}_{e\ \mathcal{S}(v)}\left[\mathbf{C}^L(\mathbf{v},\mathbf{e})[\mathbf{v}]\right] \tag{64}$$

where $\mathcal{S}(v)$ denotes the set of entities corresponding to v, $[\mathbf{C}(\mathbf{v},\mathbf{e})]$ is a cross&compress unit, and suffix $[\mathbf{v}]$ indicates the unit's output.

The module outputs the probability of user u clicking on candidate news v, computed using a nonlinear function which takes as input latent features of the user $\mathbf{u}_L$ and item $\mathbf{v}_L$, combined with a predicting function f_{RS}, such as another MLP or inner product.

The goal of the KGE module is to learn the vector representation of the tail entity of triples in the knowledge graph. For a triple (h,r,t), it firstly uses L nonlinear layers to process the raw features of relation r, using a variant of Eq. (63), and cross&compress units to extract the latent feature vector of the head entity h, with a modified Eq. (64). The tail $\hat{\mathbf{t}}$ is predicted by feeding the concatenation of the feature vectors of the head entity $\mathbf{h}_L$ and relation $\mathbf{r}_L$ into a K-layer MLP [106]. Lastly, the triple's score is calculated using the normalized inner product of the feature vectors of the real and the predicted tail representations.

The two task-specific modules are connected using cross&compress units which adaptively control the weights of knowledge transfer between the two tasks. The unit takes as input an article v and a corresponding entity e from the knowledge graph. The cross operation constructs a cross feature matrix $\mathbf{C}_l \in \mathbb{R}^{d\times d}$, by considering every possible pairwise feature interaction of their latent vector representations, $\mathbf{v}_L \in \mathbb{R}^d$ and $\mathbf{e}_L \in \mathbb{R}^d$, as follows:

$$\mathbf{C}_l = \mathbf{v}_L\mathbf{e}_L^T = \begin{bmatrix} \mathbf{v}_L^{(1)}\mathbf{e}_L^{(1)} & ... & \mathbf{v}_L^{(1)}\mathbf{e}_L^{(d)} \\ ... & & ... \\ \mathbf{v}_L^{(d)}\mathbf{e}_L^{(1)} & ... & \mathbf{v}_L^{(d)}\mathbf{e}_L^{(d)} \end{bmatrix} \tag{65}$$

Afterwards, the compress operation projects the cross features matrix back into the latent feature spaces $\mathbb{R}^d$ of items and entities in order to derive their vector representations for the following layer, as follows:

$$\begin{aligned} \mathbf{v}_{l+1} &= \mathbf{C}_l\mathbf{w}_l^{VV} + \mathbf{C}_l^T\mathbf{w}_l^{EV} + \mathbf{b}_l^V \\ &= \mathbf{v}_l\mathbf{e}_l^T\mathbf{w}_l^{VV} + \mathbf{e}_l\mathbf{v}_l^T\mathbf{w}_l^{EV} + \mathbf{b}_l^V \end{aligned} \tag{66}$$

$$\begin{aligned} \mathbf{e}_{l+1} &= \mathbf{C}_l\mathbf{w}_l^{VE} + \mathbf{C}_l^T\mathbf{w}_l^{EE} + \mathbf{b}_l^E \\ &= \mathbf{v}_l\mathbf{e}_l^T\mathbf{w}_l^{VE} + \mathbf{e}_l\mathbf{v}_l^T\mathbf{w}_l^{EE} + \mathbf{b}_l^E \end{aligned} \tag{67}$$

Although such units are able to extract high-order interactions between items and entities from the two distinct tasks, Wang et al. [106] only employ them in the model's lower layers for two main reasons. On the one hand, transferability of features decreases as tasks become more distinct in higher layers. On the other hand, both item and user features, as well and entity and relation features blend together in deeper layers of the framework, which deems them unsuitable for sharing as they lose explicit association.

The Interaction Graph Neural Network (**IGNN**) [83] aims to improve previous KGE-based recommenders by enhancing the learning process of news and user representations with collaborative signals extracted from user-item interactions. This is achieved using two graphs: a knowledge graph for modelling news-news connections, and a user-item interaction graph.

The knowledge-based component jointly learns knowledge-level and semantic-level representations of news, similar to KCNN. More specifically, the embedding matrices of words, entities, and contextual entities are stacked before applying multiple filters and a max pooling layer to compute the representation of news. In contrast to DKN, in IGNN the embeddings of entities and context, obtained with TransE [8], are not projected into the word vector space before stacking. However, as observed by Wang et al. [102], this simpler approach disregards the fact that the word and entity embeddings are learned using distinct models, and hence, are situated in different feature spaces. In turn, this means that all three types of embeddings need to have the same dimensionality in order to be fed through the convolutional layer. Nonetheless, this might be detrimental in practice, if the ideal vector sizes for the word and entity representations differs.

Higher-order latent collaborative information from the user-item interactions is extracted using embedding propagation layers that integrate the message passing mechanism of GNNs [83] using the IDs of the user and candidate news as input. This strategy is based on the assumption that if several users read the same two news, this is an indication of collaborative similarity between the pair of news, which can then be exploited to propagate information between users and news. The propagation layers inherit the two main components of GNNs, namely message passing and message aggregation. The former passes the information from news t_j to user *i*, as follows:

$$m_{i\leftarrow j} = \frac{1}{\sqrt{|N(i)||N(j)|}}(\mathbf{W}_1\mathbf{e}(t_j) + \mathbf{W}_2(\mathbf{e}(t_j) \odot \mathbf{e}(i))) \tag{68}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathcal{R}^{d' \times d}$, and $\frac{1}{\sqrt{|N(i)||N(j)|}}$ is the Laplacian norm, defined using the 1-hop neighbourhoods of user *i* and article t_j, and representing the decay factor on the propagation edge between *i* and t_j.

The latter component aggregates the information propagated from the user's neighbourhood with the current representation of the user, before passing it through a LeakyReLU transformation function, namely $\mathbf{e}(i)^{(1)} = \psi(m_{i\leftarrow i} + \sum_{j\in N(i)} m_{i\leftarrow j})$. High-order interactions are obtained by stacking multiple propagation layers, in order to expand the size of the neighbourhood considered in the message passing step.

The KCNN results in a content-based representation of news and of users, where the latter is the result of a mean pooling function applied to the embeddings of the user's previously read articles. Similarly, the *k* propagation layers result in another *k* representations of user and news. Lastly, the inner product between the final user and news representations, obtained by concatenating the two kinds of embeddings, is used to determine the user's preference for the candidate news.

In addition to using side information to extract latent interactions among news, the Self-Attention Sequential Knowledge-aware Recommendation system (**Saskr**) [25] also considers the order in which users interact with the news. The sequence of interactions of a user with a group of news articles can reveal additional preferences, as it is generally assumed that users will read news deemed more relevant in the beginning of a session, and those in which they are less interested towards the end. Saskr combines sequential-aware with knowledge-aware modelling, both built as an encoder-decoder framework, to predict the article most likely to be clicked next by a user. The model's input is constituted by a chronologically ordered sequence of *L* items read by the user, $S_t = (S_{t-L}, S_{t-L+1}, ..., S + t - 1)$, where *t* denotes the time step.

The encoder of the sequential-aware component of Saskr is composed of an embedding layer, followed by multi-head self-attention and a feed forward network. The embedding layer projects an article's body in a *d*-dimensional latent space, by combining, for each piece of news *i*, its article embedding $\mathbf{Q}_i \in \mathbb{R}^d$ and positional embedding $\mathbf{P} \in \mathbb{R}^{L\times d}$. Eq. (69) shows the resulting embedding matrix $\mathbf{E} \in \mathbb{R}^{L\times d}$.

$$\mathbf{E} = \begin{bmatrix} \mathbf{Q}_{S_{t-1}} + \mathbf{P}_1 \\ \mathbf{Q}_{S_{t-2}} + \mathbf{P}_2 \\ \cdots \\ \mathbf{Q}_{S_{t-L}} + \mathbf{P}_L \end{bmatrix} \tag{69}$$

The article's embedding can be obtained in two ways. On the one hand, it can be computed as the sum of the pre-trained embeddings of its words, weighted by the corresponding TF-IDF weights, as $\mathbf{Q}_i = \sum_w tf - idf_{w_j,i} \cdot \mathbf{w}_j$. On the other hand, it can be derived by stacking the embeddings of entities extracted from the text, namely the set *entity*(i), as $\mathbf{Q}_i = \frac{1}{|entity(i)|}\sum_{e_i \in entity(i)} \mathbf{e}_i$.

These representations are then fed into a multi-head self-attention module [100], to obtain the intermediate vector $\mathbf{M} = MultiheadAtt^s_{encoder}(E, E, E)$. In turn, this intermediate representation functions as input for the fully-connected layers which compute the final sequential-aware encoding of the user's interaction history: $\mathbf{C^s} = FFM^s_{encoder}(\mathbf{M}) = ReLU(\mathbf{MW}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$, where FFM denotes the feed forward module. The attention and feed forward modules are stacked into *B* blocks to capture deeper interactions.

Given the embedding $\mathbf{C}^s$ of the user's interaction history, and the embedding $\mathbf{Q}_{cdt}$ of candidate article i_{cdt}, the decoder predicts the sequence-aware recommendatio score using Eq. (70):

$$g^s = FFM^s_{decoder}(MultiheadAtt^s_{decoder}(\mathbf{Q}_{cdt}, \mathbf{C}^s, \mathbf{C}^s)) \tag{70}$$

The knowledge-aware module uses external knowledge from a knowledge graph to detect connections between news. The knowledge-searching encoder extracts entities from the body of articles and links them to predefined entities in a knowledge graph for disambiguation purposes. The set of identified entities is additionally expanded with 1-hop neighbouring entities. The contextual entities are embedded using word embeddings pre-trained with a directional skip-gram model [95]. The resulting contextual-entity embedding matrix $\mathbf{C}^k$ is used as input by the preference-interpreting decoder, which predicts the knowledge-aware recommendation score for candidate news i_{cdt}:

$$g^k = FFM^k_{decoder}(MultiheadAtt^k_{decoder}(\mathbf{Q}_{cdt}, \mathbf{C}^k, \mathbf{C}^k)) \tag{71}$$

The final recommendation score for candidate news article i_{cdt} is determined by aggregating the scores predicted by the two components, weighted by factor ω which adjusts the contribution of each module, as $g_{cdt} = \omega \cdot g^s + (1-\omega) \cdot g^k$.

Liu et al. [70] propose a Knowledge-aware Representation Enhancement model for news Documents (**KRED**) - a new method for creating knowledge-enhanced representations of news for multiple downstream tasks, such as news recommendation, news popularity prediction or local news detection, trained using a multi-task learning strategy. A document vector $\mathbf{v}_d$, outputted by any natural-language understanding model and encoding a news article, constitutes the input to the KRED model. The framework encompasses three layers. As in previous models, entities extracted from the news articles are linked to their correspondents in a knowledge graph, and are, in this case, embedded using TransE [8]. To take into account the contextual information of an entity, the authors employ the approach of Knowledge Graph Attention Network (KGAT) [108] to compute the representation of an entity h using the TransE embeddings of itself $\mathbf{e}_h$ and its 1-hop neighbours, as follows:

$$\mathbf{e}_{\mathcal{N}(h)} = ReLU(\mathbf{W}_0(\mathbf{e}_h \oplus \sum_{(h,r,t)\in N(h)} \pi(h,r,t)\mathbf{e}_t)) \tag{72}$$

In Eq. (72), $\pi(h,r,t)$ represents the softmax normalized attention weights that adjust the amount of information propagated from a neighbour node to a given entity. The unnormalized attention coefficients $\pi_0(h,r,t)$ are determined using a two-layer fully connected neural network:

$$\pi_0(h,r,t) = \mathbf{w}_2 ReLU(\mathbf{W}_1(\mathbf{e}_h \oplus \mathbf{e}_r \oplus \mathbf{e}_t) + \mathbf{b}_1) + b_2 \tag{73}$$

The next, context embedding layer encodes the dynamic context of entities from a news article, determined by their position, frequency and category. The entity's position in the article (i.e. in the title or body) is encoded using a bias vector $\mathbf{C}^{(1)}_{p_h}$, $p_h \in \{1,2\}$. While entities appearing in both the article's body and title are considered more important, so are those occurring more often. The frequency of an entity is encoded by the vector $\mathbf{C}^{(2)}_{f_i}$. Lastly, a category encoding vector $\mathbf{C}^{(3)}_{t_i}$ indicates the entity type t_i. The embedding of entity h is thus enhanced in the following way:

$$\mathbf{e}_{\mathcal{I}_h} = \mathbf{e}_{N(h)} + \mathbf{C}^{(1)}_{p_h} + \mathbf{C}^{(2)}_{f_i} + \mathbf{C}^{(3)}_{t_i} \tag{74}$$

The entities' representations are aggregated into a single vector in the information distillation layer, by means of an attention mechanism which takes into account both the context-enhanced entity vectors and the original embedding of an article to compute its final representation. More specifically, the attention weights $\pi_0(h,v)$, computed according to Eq. (75), and then normalized using a softmax function, are used to weight the sum of entities from the same article to obtain its embedding $\mathbf{e}_{\mathcal{O}_h}$, as per Eq. (76).

$$\pi_0(h,v) = \mathbf{w}_2 ReLU(\mathbf{W}_1(\mathbf{e}_{\mathcal{I}_h} \oplus \mathbf{v}_d + \mathbf{b}_1) + b_2 \tag{75}$$

$$\mathbf{e}_{\mathcal{O}_h} = \textstyle\sum_{h\in\mathcal{N}(v)} \pi(h,v)\mathbf{e}_{\mathcal{I}_h} \tag{76}$$

The knowledge-aware document vector $\mathbf{v}_k$ is afterwards obtained by concatenating the entity and original document vectors and passing them through a fully-connected feed-forward network. In contrast to DKN, KRED is not constrained by the type of document embedding model. Hence, it allows any state-of-the-art, pre-trained or fine-tuned representation to be incorporated in the framework. Additionally, it is not restricted to short sequences of text, such as titles, but it can handle different types of data, including news bodies and metadata [70].

In addition to injecting external knowledge into the recommendation model, the Topic-Enriched Knowledge Graph Recommendation System (**TEKGR**) [62] improves items' representation by exploiting the topical relations among news. This is based on the assumption that even if two news share knowledge entities in which the user might be interested, they may belong to different topics, which are not all relevant for the reader. TEKGR, constructed of three layers, takes as input a user's click history and a candidate article. News are represented by their titles.

Firstly, the KG-based news modelling layer is composed of three encoders and outputs a vector representation for each given article. The word-level news encoder learns news representations using their titles without considering latent knowledge features. The first layer of the encoder projects the titles' sequence of words into a lower dimensional space, while the bidirectional GRU (Bi-GRU) layer encodes the contextual information of a news title. Bi-GRU obtains the hidden state of an article by concatenating the outputs of the forward and backward GRUs [62]. This is followed by an attention layer which extracts more informative features from the vector representations by giving a higher importance to more relevant words. Hence, the final representations of news article $\mathbf{e}(t_k)$ is given by the weighted sum of the contextual words representations, where the weights are attention coefficients.

The knowledge encoder extracts topic information from the news titles through three layers [62]. The concept extraction layer links each news title with corresponding concepts in a knowledge graph using a "is-a" relation. Afterwards, the concept embedding layer maps the extracted concepts to a high-dimensional vector space, while the self-attention network computes a weight for each word in the news title according to the associated concept and topic. For example, in the news title from Figure 1, *Elon Musk* will have a higher attention weight in relation with the *entrepreneur*, than with the *programmer*, concept. The layer's output is then concatenated with the news embedding vectors obtained from the word-level encoder.

The third, KG-level news encoder firstly performs a knowledge distillation process. The resulting subgraph is enriched with 2-hop neighbours of the extracted entities, as well as with topical information distilled by the knowledge encoder [62]. Therefore, not only are knowledge entities from the text disambiguated and their contextual information taken into account, but also adding topical relations among entities decreases data sparsity by connecting nodes not previously related in the knowledge graph. The topic and knowledge-aware news representation vector is computed with a graph neural network [105]. The final news embeddings are obtained by concatenating the word-level and KG-level representations.

Secondly, the attention layer computes the final user embedding by dynamically aggregating each clicked news with respect to the candidate news. This step is accomplished as in DKN, by feeding the concatenated embedding vectors of the user's click history and the candidate news into a DNN. Lastly, the user's probability of clicking on the target article is computed in the scoring layer using the dot product of the user's and article's feature vectors.

5.4.3. Summary

KGE-based news recommendation systems are summarized by focusing on four distinguishing aspects:

- **Recommendation model input.** These methods use the user's news interaction history and a candidate article as input. The user's interaction history is most often represented by previously clicked items. In such cases, the user profile is created by aggregating the representations of the individual articles from the click history. In contrast, CETR uses a user-item interaction matrix to represent the connection between users and news, and to generate collaborative recommendations. Similarly, RippleNet computes recommendations using the matrix of implicit feedback and a knowledge graph. Furthermore, the majority of models uses a combination of word-level and entity-level representations of articles, based usually on their titles. The entities directly extracted from the news articles are further enriched with contextual information from the knowledge graph, in the form of k-hop neighbours, where the maximum number of hops considered represents one of the model's hyperparameters. Furthermore, Saskr is the only model to take into account the order in which a user interacts with a sequence of news articles.
- **Knowledge graph embedding model.** Several approaches for embedding knowledge graph entities have been identified in the surveyed frameworks. Recommenders such as CETR, DKN, IGNN or KRED use TransE [8], TransH [109], TransR [69], or TransD [56] to compute knowledge graph embeddings. MKR uses a combination of MLP and cross&compress units, while Saskr embeds knowledge entities with pre-trained word embeddings. More recently, TEKGR adopts a GNN for deriving entity embeddings.

- **Components of recommender system.** With the exception of CETR, which uses a combination of matrix factorization, topic analysis and KGE models, the other systems are based on various combinations of neural networks. MKR uses a combination of MLPs and cross&compress units to train two components for the tasks of recommendation and knowledge graph embedding, while IGNN fueses KCNN for content-based representation of news with a message-passing GNN that captures collaborative signals among news. All the remaining models utilize a type of attention mechanism. For example, DKN combines KCNN used for news representation with a DNN-based attention layer. The fine-grained DKN with self-attention incorporates only self-attention modules at all three - word, item and user - levels, and employs another multi-head attention layer followed by a fully-connected layer for the final prediction. Similarly, Saskr is composed only of multi-head self-attention and fully connected layers. TEKGR and KRED combine attention modules with different types of GNNs. KRED uses a KGAT to aggregate the embeddings of an entity with those of its neighbours, followed by the attention mechanism of the Transformer [100] used for assigning different weights for each entity and for computing the article's final embedding. TEKGR combines attention with bi-GRU in the word-level encoder, and with KGE in the knowledge encoder. Additionally, it incorporates a GNN in the KG-level news encoder.
- **Aggregation of knowledge-level and text-level components.** As previously observed, the attention mechanism is widely used in models such as DKN, KRED, or TEKGR, to dynamically aggregate the outputs of different model components or the representations of individual modules at intermediate steps in the framework. A simpler strategy is adopted in IGNN, where the content-based and collaborative representations of news and users are concatenated before computing the final prediction. In comparison, MKR uses cross&compress units at the lower levels of its model to transfer similar latent features between the two task-specific components.

6. Evaluation Approaches

This section analyses approaches used for evaluating the surveyed knowledge-aware news recommender systems, as well as potential limitations concerning the reproducibility and comparability of experiments.

6.1. Evaluation methodologies

The type of evaluation methodology depends on the output of the recommendation models and the used data. In this context, the surveyed recommender systems were typically evaluated either through offline experiments based on historical data, through online studies on real-world websites, or in laboratory studies. Frameworks based on an item-ranking output usually use an online setting or laboratory experiment. In these scenarios, participants are asked to annotate news articles recommended to them by the model based on their relevance to the user's profile. In turn, the user profile is either created during the experiment or predefined and assigned to the participants by the evaluators. Once the annotations are obtained, the performance of the model is evaluated by comparing the predicted recommendations against the truth values provided by the annotators. In contrast, systems which output the click-through rate perform experiments in an offline settings, using data comprising of logs representing users' historical interactions with sets of news.

Table 3 provides an overview of evaluation settings in terms of datasets and metrics used. As it can be observed there, all models use different types of information retrieval accuracy measures, such as precision, recall, F1-score, or specificity. Some of the more recent, KGE-based systems also evaluate the model's performance in terms of rank-based measures, such as Normalized Discounted Cumulative Gain, Hit Rate, or Mean Reciprocal Rank. Generally, these metrics are computed at different positions in the recommendation list to observe the recommender's performance based on the length of the results list. Moreover, methods based on the vector space model use statistical hypothesis tests, such as the Student's *t*-test, to measure the significance of the experimental results.

6.2. Evaluation datasets

In comparison to the relatively uniform usage of evaluation metrics, the type of datasets used for evaluation varies significantly among recommender systems. Nearly all of the models based on the vector space model and on semantic similarities can be clustered into two groups, depending on the dataset used. As shown in Table 3, semantic aware context recommenders use the News@hand architecture described

Table 3

Overview of evaluation settings. The first two columns list the category and recommendation system. The third and fourth columns enumerate the dataset(s) and metrics used for evaluation. The last column shows the evaluation setup information provided for reproducibility purposes (i.e. processing steps, data split, parameters, code, all available). The abbreviations used in the table are the following: Eval. = Evaluation, Acc = Accuracy, P = Precision, R = Recall, F1 = F1-score, NDGC = Normalized discounted cumulative gain, RMSE = Root mean square error, MAE = Mean absolute error, ROC = Receiver operating characteristic, PR curves = Precision-recall curves, AUC = Area under the curve, NDPM = Normalized distance-based performance measure, HR = Hit rate, MRR = Mean reciprocal rank, Kappa = Kappa statistics, Student's *t*-test=One-tailed two-sample paired Student *t*-test, PROCS = processing steps, DS = data split, PARAMS = parameters.

Category	Model	Dataset(s)	Metric(s)	Eval. setup information
VSMM	CF-IDF [44]	Hermes News Portal	ROC, PR curves, Kappa	DS
	SF-IDF [20]	Hermes News Portal	Acc, P, R, F1, Spec, *t*-test	DS
	SF-IDF+ [75]	Hermes News Portal	Acc, P, R, F1, Spec, *t*-test	DS, PARAMS
	Bing-SF-IDF+ [22]	Hermes News Portal	Acc, P, R, F1, Spec, Kappa	DS, PARAMS
	OF-IDF [86]	Hermes News Portal	P, R, F1, runtime	DS
SSM	Semantic aware context recommendation [14, 16]	News@hand	P@K	PROCS, DS
	Social tags enriched recommendations [18]	News@hand	Relevance	PROCS, DS
	Semantic relatedness [41]	Unknown source	P, R, F1	-
	RSR [54]	Hermes News Portal	Acc, P, R, Spec	-
	Hybrid context-aware recommendation [17]	News@hand	P@K	PROCS, DS
	RSR 2 [37]	Hermes News Portal	Acc, P, R, Spec	DS
	SS [20]	Hermes News Portal	Acc, P, R, F1, Spec, *t*-test	DS
	BingSS [21]	Hermes News Portal	Acc, P, R, F1, Spec	DS, PARAMS
DM	ePaper [73, 94]	The Jerusalem Post	NDPM, MAE	PROCS, DS, PARAMS
	Magellan [31]	Unknown source	Acc, R	-
	OBSM [85]	New York Times, Sina News	P, R, F1	DS
	SED [58]	CNREC	P, R, F1	PROCS, DS, PARAMS
KGEM	CETR [117]	Hupu News	R@K	PROCS, DS, PARAMS
	DKN [102]	Bing News	F1, AUC	All available
	Fine-grained DKN with self-attention [38]	Unknown source	AUC, NDCG@K	DS, PARAMS
	RippleNet [103]	Bing News	P@K, R@K, F1@K, AUC, Acc	All available
	RippleNet-agg [104]	Bing News	P@K, R@K, F1@K, AUC, Acc	All available
	MKR [106]	Bing News	Acc, AUC, P@K, R@K	All available
	IGNN [83]	DC, Adressa	R@K, NDCG@K	PROCS, DS, PARAMS
	Saskr [25]	Eastday Toutiao	HR@10, MRR	PROCS, DS
	KRED [70]	Microsoft News	AUC, NDCG@K, HR@K, ACC, F1-macro	All available
	TEKGR [62]	Bing News, Adressa	F1, AUC	PROCS, DS, PARAMS

in [15]. The remaining models are incorporated in the Hermes News Portal [36]. Recommender systems based on distance construct their own datasets using news articles collected from websites such as the New York Times, or Sina News[7]. Joseph and Jiang [58] developed CNREC[8] for evaluating SED. CNREC is a dataset providing articles similarity and annotations for pairs of items showing the extent to which they are considered a good recommendation.

The datasets used by KGE-based frameworks consist of user interaction logs gathered from websites such as Bing News[9], Microsoft News[10], Hupu[11], or Eastday Toutiao[12]. An exception is constituted by IGNN and TEKGR, evaluated on the Adressa dataset. Adressa [45] is an event-based dataset comprising of click log data collected from a Norwegian news portal. Although the Adressa dataset is often utilized in evaluating deep learning-based news recommender systems [49, 50, 79, 121], it is not used by any other of the surveyed knowledge-aware models.

As it can be further observed in Table 4, which summarizes the statistics of the used evaluation datasets, the number of users and items contained in these

[7] http://news.sina.com.cn

[8] https://github.com/kevinj22/CNRec/blob/master/CNRec.zip

[9] https://www.bing.com/news

[10] https://news.microsoft.com

[11] https://www.hupu.com

[12] http://mini.eastday.com/

Table 4
Overview of evaluation datasets. The first two columns list the category and recommendation system. The third and fourth columns indicate the data source and the language of the dataset. The fifth column shows the time frame used for the collection of the dataset, whereas the remaining columns illustrate different statistics of the data. An entry annotated with "*" denotes that the statistics were approximated by us based on the data provided by the authors. The abbreviations used in the table are the following: # = number of, N/A = not applicable.

Category	Model	Data source	Language	Time frame	#Users	#Items	# Logs/Interactions
VSMM	CF-IDF [44]	Unknown	Unknown	Unknown	19	100	1900 *
	SF-IDF [20]	Unknown	Unknown	Unknown	19	100	1900
	SF-IDF+ [75]	Reuters	English	Unknown	N/A	100	N/A
	Bing-SF-IDF+ [22]	Reuters	English	Unknown	N/A	100	N/A
	OF-IDF [86]	Unknown	Unknown	2010	33	1823	3600
SSM	Semantic aware context recommendation [14, 16]	BBC, CNN, New York Times, Washington Post	English	01/01/2008-01/03/2008	16	9,698	N/A
	Social tags enriched recommendations [18]	BBC, CNN, New York Times, Washington Post	English	01/01/2008-01/03/2008	20	9,698	N/A
	Semantic relatedness [41]	CNN, BBC, USA Today, L.A. Times, Reuters	English	Unknown	N/A	158	N/A
		Synthetic data	Unknown	N/A	N/A	100	N/A
	RSR [54]	Unknown	Unknown	Unknown	5	Unknown	1500 *
	Hybrid context-aware recommendation [17]	BBC, CNN, New York Times, Washington Post	English	01/01/2008-01/03/2008	20	9,698	N/A
	SS [20]	Unknown	Unknown	Unknown	19	100	1900
	BingSS [21]	Reuters	English	Unknown	N/A	100	N/A
DM	ePaper [73, 94]	The Jerusalem Post	English	4 days	57	Unknown	4,731 *
	Magellan [31]	Unknown	Unknown	09/2010-01/2011	N/A	Unknown	N/A
	OBSM [85]	New York Times	English	2006-2007	581	6,000	232,400 *
		Sina News	Chinese	Unknown	581	Unknown	232,400 *
	SED [58]	Unknown	English	25/08/2014-28/08/2014	N/A	300	N/A
KGEM	CETR [117]	Hupu News	Chinese	Unknown	3,118	9,684	132,713
	DKN [102]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
	Fine-grained DKN with self-attention [38]	Unknown	Unknown	19/03/2018-31/03/2018	26,224	13,285	1,498,862
	RippleNet [103]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
	RippleNet-agg [104]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
	MKR [106]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
	IGNN [83]	DC	Unknown	Unknown	10,000	6,385	116,225
		Adressa	Norwegian	2017	640,503	20,428	3,101,991
	Saskr [25]	Eastday Toutiao	Chinese	30/10/2018-13/11/2018	6960	108,684	861,996
	KRED [70]	Microsoft News	Unknown	15/01/2019-28/01/2019	665,034	24,542	1,590,092
	TEKGR [62]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
		Adressa	Norwegian	2017	561,733	11,207	2,286,835

datasets varies widely. Models from the first three categories are evaluated on small datasets, usually with less than 1000 articles, with the exception of the semantic contextualization systems, tested with nearly 10,000 items. In contrast, KGE-based methods are mostly evaluated on over 1 million click logs from more than 100,000 users and items.

Another critical finding is that in many cases, datasets are not described clearly enough. In nearly a fourth of the cases, the data source is not specified. Moreover, the language of the dataset is rarely mentioned explicitly. While the language can easily be deduced from monolingual news websites, this does not hold true for international news platforms, leading to an unknown language in half of the cases.

6.3. Reproducibility and comparability of experiments

Table 3 also lists the type of information provided by each model with regards to the evaluation setup. Replicating experiments requires not only access to the data used, but also knowledge of how the data was split and processed for training and evaluation, and which values were used for the different parameters and hyperparameters of the model. Moreover, differences in the models' implementation, especially of models based on deep learning, can further influence the results obtained when reproducing experiments. Hence, access to the original implementation constitutes an important factor for the comparability and reproducibility of results.

However, as it can be observed in the last column of Table 3, only 5 out of the 27 surveyed papers provide all this information. For the first three categories of models, generally only the data split and some of the parameters are specified. Even when some processing steps are explained in the paper, not enough details are provided regarding how procedures such as named entity recognition or entity linking were performed. Moreover, systems in these categories each propose their own evaluation setups, without following a uniform procedure.

In contrast, most papers describing a KGE-based framework offer extensive details regarding their evaluation settings and model architecture. This phenomena could be explained in two ways. On the one hand, since all KGE-based models are deep learning architectures, hyperparameters play a central role in their performance. On the other hand, in recent years it has become increasingly important in the academic community to make implementation details available when publishing a research paper. Nonetheless, important aspects which would increase the comparability of experiments are still neglected in some works. Often, the entity extraction and linking processes are not thoroughly explained, meaning that if no implementation details are available, it would be impossible to reproduce the exact steps of the original experiments. In Saskr, for example, the authors offer few details on the construction of the news-specific knowledge graph used, and no specification of the news data source, or knowledge graph construction process.

Another significant factor to be considered in the evaluation and comparison of recommender systems is how different model features and components affect its performance. All of the surveyed papers compare their knowledge-aware news recommendation models against baselines which do not incorporate side information in order to illustrate the gains of a knowledge-enhanced system. In addition to evaluating a model against baselines and state-of-the-art systems, it is also necessary to understand the effect of different features and modules on the recommender's performance. To this end, the choice of knowledge resource is critical for a knowledge-aware model. However, none of the papers compare their model's performance using different knowledge bases to determine the extent to which the resource itself influences results.

As it can be seen in Table 5, only three papers from the first two categories evaluate their model's parameters. In these cases, the threshold values determining which articles are suggested to the user are empirically tested. In the case of recommenders based on distance, only ePaper and SED evaluate the influence of different parameters or user profile initialization on the model's performance. In comparison, the evaluation of KGE-based systems involves parameters sensitivity analysis, as well as experiments with different initialization, training or embedding strategies. Such extensive experiments could also be influenced by the type of models, since neural network architectures comprise of several components, and are more sensitive to hyperparameters and design choices than models from the first two categories.

An additional finding is that few works perform an ablation study to determine the contribution of each component to the overall system. In the case of semantic aware context recommenders, the authors analyse variants of the model obtained by removing either the contextualization of user preferences, the extension of user and news profiles, or both. DKN removes

Table 5
Overview of evaluated features and components. The first two columns list the category and recommendation system. The third column enumerates the features whose influence on the model's performance was examined. The fourth column indicates whether an ablation study was conducted and, if so, which components were investigated. The abbreviations used in the table are the following: dim. = dimension, emb. = embedding, init. = initialization, # = number of.

Category	Model	Eval. feats.	Ablation study
VSMM	CF-IDF [44]	threshold value	-
	SF-IDF [20]	-	-
	SF-IDF+ [75]	-	-
	Bing-SF-IDF+ [22]	-	-
	OF-IDF [86]	-	-
SSM	Semantic aware context recommendation [14, 16]	-	model components
	Social tags enriched recommendations [18]	-	-
	Semantic relatedness [41]	-	-
	RSR [54]	threshold value	-
	Hybrid context-aware recommendation [17]	-	model components
	RSR 2 [37]	-	-
	SS [20]	-	-
	BingSS [21]	threshold value, BingSS parameters	-
DM	ePaper [73, 94]	matching parameters, #user ratings, user profile init., concept weights	-
	Magellan [31]	-	-
	OBSM [85]	-	-
-	SED [58]	length expansion radius, entity screening, context words, edge weighting schema, distance measure, disconnected nodes penalty	-
KGEM	CETR [117]	-	-
	DKN [102]	word & entity emb. dim., #filters, window size	knowledge & attention component, KGE model, transformation function
	Fine-grained DKN with self-attention [38]	user profile length, #keywords	-
	RippleNet [103]	ripple set size, #hops, emb. dim. regularization weight	-
	RippleNet-agg [104]	aggregator, ripple set depth, neighbourhood sampling size, emb. dim.	-
	MKR [106]	KG size, RS training frequency, emb. dim.	cross&compress units, multi-task learning
	IGNN [83]	emb. dim., #emb. propagation layers	emb. propagation layers
	Saskr [25]	emb. layer init., article emb. strategy, sequence length, #targets, weight factor	-
	KRED [70]	base document vector, training strategy	layers (incl. knowledge component)
	TEKGR [62]	#hops	encoder types

not only the knowledge component during the ablation study, but also experiments with different types of knowledge graph embedding models. Additionally, DKN's performance was tested using different transformation functions, as well as with and without the attention module. MKR evaluates the contribution of its cross&compress units by replacing them with different modules, while IGNN examines the effectiveness of the embedding propagation layers by comparing different model variants which use them either to enhance the news, the user, or both representations. TEKGR evaluates the improvements of using side information by analysing the effect of its KG-level and knowledge encoders. Similarly, KRED conducts an ablation study in which it removes each of its entity representation, context embedding and information distillation layers.

6.4. Summary

Overall, this investigation of evaluation approaches shows that there is no unified evaluation methodology to produce comparable experiments. Moreover, the datasets utilized for evaluation are freely chosen by the authors, and there is no clear benchmark set of datasets used by all the systems. Although an effort has been made in recent years to provide more details on the evaluation setup, model architectures and choice of parameters, there is often still too little information specified for critical processing steps. In conclusion, it can be argued that only some of the most recent, KGE-based approaches could be replicated given the available data, whereas older methods based on the vector space model, on semantic similarities or on distance, cannot be reproduced in accordance to the original implementations. In this context, knowledge-aware news recommendation models can be said to lack reproducibility and comparability, standards which have been strongly encouraged in other fields of machine learning.

7. Future Directions and Open Issues

Existing works have already established a strong foundation for knowledge-aware news recommender systems. In this section, we identify and elaborate on several open issues in the field, and propose a number of promising research directions.

7.1. Comparability of evaluations

Zhang et al. [118] have observed that the entire field of recommender system lacks a unified evaluation methodology or benchmark datasets, which are common in the domains of computer vision or natural language processing to ensure a fair comparison of models. The findings of Section 6.3 have shown that currently, knowledge-aware news recommender systems also hardly produce comparable experiments. While KGE-based methods have a higher degree of reproducibility, the other models do not provide enough details on their evaluation methodology in order to be accurately replicated. Another important observation is that none of the deep learning models have been compared against recommenders from the other three categories. Nonetheless, comparability of evaluations is essential for benchmarking different models, which in turn, drives advancements in the field. Therefore, we argue that the field of knowledge-aware news recommender systems needs a stricter and more unified evaluation approach, including common benchmark datasets, clear processing steps, standardized metrics, unified and transparent hyperparameters in downstream fine-tuning, and ablation studies.

As it is common in other fields of machine learning, we believe that a set of benchmark datasets is needed to compare and contrast news recommenders. Such datasets should address all downstream tasks in the field of news recommendation, such as click-through rate or popularity prediction. Moreover, benchmark datasets should cover a wide range of sizes. Since scalability constitutes a key factor for a good news recommender, evaluating models on datasets of various sizes would prove to what extent a system could be used in real-world scenarios. Furthermore, benchmark datasets should have clearly defined splits for training, testing, and validation. This requirement is necessary to prevent each author from creating randomized test splits, which cannot be replicated. Wu et al.[111] have recently constructed MIND, a large-scale dataset for news recommendation containing click logs of 1 million users on English articles from Microsoft News. Similar efforts have already been conducted in other fields. Datasets such as MNIST or ImageNet in computer vision, or SQuAD in natural language processing, are already widely used for comparing models in their respective domains. With the creation of Open Graph Benchmark, the Graph Neural Network community has recently undertaken a similar effort in creating a set of benchmark datasets from varying domains and sizes [51].

In addition to evaluating on the same datasets, with the same data splits, it is necessary to establish a stricter criteria for describing the evaluation methodology in order to ensure replicability of experiments. This means that detailed information of all processing steps, from named entity recognition to the creation of news-specific knowledge graphs, should be provided to ensure that the experimental setup can be accurately reproduced at all steps.

The majority of papers already use the same information retrieval and rank-based metrics to evaluate their models. Nevertheless, every model should be evaluated using the same set of measures, which requires standardizing a set of evaluation metrics for each downstream application. Additionally, if the metrics consider the position of a recommendation in the results list, the same set of ranks should be applied throughout all modes being benchmarked.

Moreover, when comparing models against each other, authors should use unified and transparent sets of parameters in downstream fine tuning, in as far as possible given the recommendation framework. For example, the same knowledge graph or word embedding models, same neighbourhood sizes or embedding dimensions, should be used by all analysed methods.

Furthermore, ablation studies should be performed for each newly proposed model to investigate the contribution of each component to the whole system. While this holds true for any recommender system, for knowledge-aware techniques it is essential to test the influence of the knowledge component, as done, for example, in DKN's evaluation. Another interesting experiment would be to investigate the effect of the knowledge resource itself on the recommender's performance, by injecting external knowledge, for example, from different knowledge graphs.

Overall, all these steps would ensure that not only are models fairly and transparently compared against each other without great variations in parameter settings, but would also indicate whether the improvement of a new model over the state-of-the-art results is determined by the system's architecture, or simply, by a better tuned set of hyperparameters. Similar studies that can serve as an example for the field of news recommendation have been conducted for graph neural networks [33] or knowledge graph embeddings [91].

Lastly, we believe that a comparison of older and newer knowledge-aware news recommender systems is needed to compare and understand the strengths and weaknesses of all existing approaches for incorporating external knowledge into news recommendations.

7.2. Scalability of news recommenders

The continuously increasing amount of news published daily, as well as the growing number of online news readers constitute a constant challenge for any news recommender system, which require scalability in order to be applied in real-world scenarios. Several techniques, ranging from fast clustering to dimensionality reduction, have been proposed to address the scalability issue. For example, Li et al. [65] proposed a scalable news recommender system which firstly clusters news articles based on their content in order to reduce the amount of similarity computations required for personalized recommendation. A combination of three approaches has been adopted by Das et al. [27] to improve the scalability of a recommender system dealing with millions of users and articles from Google News. A MinHash-based user clustering algorithm and Probabilistic Latent Semantic Indexing (PLSI) [48], both adapted for large-scale dataset scalability using the MapReduce framework [28], were employed by Das et al. [27] to cluster dynamic news datasets. These methods were combined with an item covisitation technique for extracting user-item relations to generate personalized news recommendations.

However, the injection of external information in the recommender systems further enlarged the scale of the datasets that need to be processed by the model, particularly in the case of frameworks using knowledge graphs as side information. As shown in Sections 5.3 and 5.4, such models obtain scalability using subgraphs, constructed by sampling fixed-sized neighbourhoods. While this approach ensures that the recommendation model scales arbitrarily regardless of the size of the full graph, by not considering the entire graph at once, it is possible to ignore relevant neighbours of a node when gathering its contextual information. Hence, the sampling strategy used for defining a node's neighbourhood during subgraph construction influences the efficiency of the model. Overall, it can be concluded that knowledge-aware news recommender systems ensure scalability by sacrificing knowledge graph completeness. In this context, a promising research direction would be to investigate how to balance scalability and knowledge graph completeness in each downstream application scenario. To this end, we believe that an analysis of the effect of sampling strategy and neighbourhood size on the robustness of the system and quality of the recommendations, as performed in [104], should be conducted for a larger variety of recommenders.

7.3. Explainability of recommendations

Providing explanations for the results generated by a recommender system helps users to understand why a certain item has been recommended to them by the model. In turn, this can increase the users' trust in the system. For example, LISTEN, a model designed to explain rankings generated by a news recommendation model [98], explains the ranking of recommendations by identifying the most important features contributing to the current ranking and providing them to the user in a human interpretable form. The importance of features is determined by disrupting their values, one at a time, and observing how the change affects the ranking. In this case, a significant feature value will substantially change the ranking [98].

Although the workings and outputs of deep learning-based recommender systems are intricate and often not easily interpretable by non-expert users, attention mechanisms have recently alleviated the lack of interpretability of neural models. Attention weights not only provide insights into the inner functioning of a system, but also serve as explanations for which features in a user's or item's profiles have contributed to the model's recommendation. In this context, the Dynamic Explainable Recommender was designed by Chen et al. [24] to increase the accuracy of user modelling by taking into account the dynamic nature of user's preferences, while providing recommendation explanations. More specifically, the model utilizes time-aware gated recurrent units to encode the user's dynamic preferences and sentence-level convolutional neural networks to represent items based on the information captured in their reviews. The review information of different items is combined using a personalized attention mechanism, which learns the relevant pieces of information from a review according to the user's current preferences, thus being able to explain the generated recommendations tailored to the user's current state [24]. A different approach for balancing the accuracy and explainability of recommendations was adopted by Gao et al. [39], who built a rating prediction model using an attentive multi-view learning framework based on an explainable deep hierarchy. An attention mechanism connects adjacent views denoting different levels of features representing a user's profile. Personalized explanations are generated from these multi-level features using a constrained tree node selection solved with dynammic programming [39].

Incorporating knowledge graph information into recommender systems has been used not only to improve recommendation accuracy, but also to increase the explainability of results, as paths capturing user-items interactions in the knowledge graph could illustrate which semantic relations and entities contribute to a particular recommendation given the input user profile [52, 107, 114]. As such, reasoning over the knowledge graph can reveal possible user interests and provide explanations for why a certain article has been recommended to the reader. Another means of using a knowledge graph to provide users with human-readable explanations for a recommender's prediction was proposed by Ma et al. [72]. Their method learns inductive rules from an item-centric knowledge graph, which encode items associations in the form of multi-hop relational patterns. The induced rules are incorporated in the recommendation module to address the cold start problem and provide explainability.

A growing number of recent news recommeders employ graph neural networks as components in the framework. However, these deep learning models are often seen as black-box models, whose interpretability is concealed to regular users. The GNNExplainer proposed by Ying et al. [116] is a model-agnostic approach for explaining predictions of any GNN-based model. The method takes as input a trained GNN and a prediction and generates an explanation in the form of a compact subgraph of the input graph and a small subset of node features with the highest impact on the given prediction. Computing explanations requires optimizing the subgraph structure such that its mutual information with the GNN's prediction is maximised. Given the increasing usage of graph neural networks in news recommender systems, the GNNExplainer could be utilized to provide explanations for knowledge-aware news recommendations.

Hitherto, to the best of our knowledge, an explainable knowledge-aware news recommender system has not yet been designed. Providing explanations for online news readers remains thus an open problem. Therefore, we believe this is a noteworthy avenue which should be explored in future research.

7.4. *Fairness of recommendations*

Nowadays, news recommender systems have an increasing influence over people's lives, by controlling which articles a reader is exposed to. This has raised concerns about biases that might be amplified by such systems. Yao and Huang [115] identified two types of biases inherent in recommender systems, namely observation bias and population imbalance bias.

Observation bias is determined by feedback loops which prevent the model from learning how to predict items which are dissimilar to the previously recommended or consumed ones [35]. Content-based recommenders generate suggestions that are similar to the ones in the user's history, while collaborative filtering systems recommend items liked by similar users. In both cases, the model learns to make predictions based on its past actions, since users cannot provide feedback for items which are not recommended to them, thus reinforcing the recommender's algorithmic behaviour [35]. In the context of news recommendation, observation bias has given rise to the hypothesis that readers become trapped inside filter bubbles - states in which they are exposed only to news that support or amplify their opinions [80]. In turn, this might lead, in the long-run, to opinion polarization and self-radicalization of individuals through online media [78].

Bias stemming from imbalanced data is a systematic bias caused by societal or historical discriminations, which occurs when different categories of users are represented in unequal proportions in the data used for training a recommender system [115]. For example, population imbalance bias would occur if a recommender would suggest technology news mainly to men, and cooking articles to women.

Several techniques have been designed for fair recommender systems in general. For example, Beutel et al. [5] proposed using pairwise comparisons as a metric for measuring the ranking fairness of a recommender system. Moreover, they introduce a pairwise regularization method to improve the model's fairness property during training. Burke et al. [12] identify multiple stakeholders of a recommender system and distinguish between different types of fairness depending on the corresponding stakeholder group, namely consumer-centered, provider-centered, or both. The authors propose using the concept of balanced neighbourhoods combined with a sparse linear model to obtain a desirable trade-off between fairness of results and personalization of recommendations [12].

Wu et al. [110] proposed using decomposed adversarial learning and orthogonality regularization to diminish unfairness caused by the biases of sensitive user attributes, such as gender, in news recommendation. More specifically, during training, the model learns two types of user embeddings: bias-aware ones that capture biases encoded in sensitive attributes describing the user's behaviours, and bias-free ones that capture attribute-independent information related to the user's interests. Adversarial learning is utilized to ensure that the bias-free embeddings do not contain information from the sensitive user attributes, while orthogonality regularization ensures that the two types of representations are orthogonal to each other. Lastly, fairness-aware news recommendations are computed using only the bias-free user embeddings [110].

However, these methods have been developed for traditional recommender systems and do not consider biases that might stem from the knowledge resource used as side information. Hence, investigating how fairness can be incorporated into knowledge-aware news recommender systems represents a promising direction for future works in this field.

7.5. Multi-task learning for recommendation

Multi-task learning [23] is a transfer learning-based paradigm which aims to exploit similarities across different tasks in order to improve the generalization performance of a model. The model is trained for multiple related tasks in parallel and domain-specific information is transferred between tasks to prevent overfitting on a single downstream application [119]. This approach has proven successful in numerous applications, ranging from computer vision to speech recognition and natural language processing [90].

Multi-task learning has also been employed by recommender systems from different domains [77]. In the case of recommender systems using knowledge graphs as side information, the quality of recommendation might be negatively affected by missing facts in the knowledge graph as the user's preferences may be ignored if they are not captured by existing entities and relations. Recent works have shown that jointly learning a model for both recommendation and knowledge graph completion can result in improved recommendations [19, 67]. Similarly, in the field of knowledge-aware news recommendation, Wang et al. [106] have utilized this paradigm to jointly train a model for the tasks of news recommendation and knowledge graph embedding, while Liu et al. [70] jointly trained a knowledge-aware representation enhancement model for news documents on a variety of tasks, ranging from item recommendation to local news prediction.

Taking into account the advantages of the multi-task learning paradigm, we believe that utilizing transfer knowledge from tasks such as entity classification or link prediction for knowledge-aware news recommendation is a promising direction to pursue in the future.

8. Conclusion

This survey paper extensively reviews knowledge-aware news recommender systems. We propose a new taxonomy for classifying and clustering existing recommenders, based on how external knowledge is injected in the recommendation model to improve the results. According to the classification scheme, we categorize knowledge-aware news recommender systems into frameworks based on the vector space model, on semantic similarities, on distance, and on knowledge graph embeddings. Representative models from each category are summarized, and thoroughly analysed. Moreover, we discuss and compare evaluation approaches used by existing publications and identify limitations in terms of comparability and reproducibility of experiments. Lastly, we identify and examine open issues in the field and propose future research directions that could drive progress in this domain. We hope this survey can serve as a comprehensive overview of knowledge-aware news recommender systems, clarifying key aspects of the field and uncovering open problems and corresponding promising directions to pursue in future studies.

9. Acknowledgements

The work presented in this paper has been conducted in the ReNewRS project, which is funded by the Baden-Württemberg Stiftung in the Responsible Artificial Intelligence program.

References

[1] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE transactions on knowledge and data engineering* **17**(6) (2005), 734–749.

[2] M. An, F. Wu, C. Wu, K. Zhang, Z. Liu and X. Xie, Neural news recommendation with long-and short-term user representations, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 336–345.

[3] J. Beel, B. Gipp, S. Langer and C. Breitinger, Research-paper recommender systems: a literature survey, *International Journal on Digital Libraries* **17**(4) (2016), 305–338.

[4] F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault and J. Morissette, Bio2RDF: towards a mashup to build bioinformatics knowledge systems, *Journal of biomedical informatics* **41**(5) (2008), 706–716, Publisher: Elsevier.

[5] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E.H. Chi et al., Fairness in recommendation ranking through pairwise comparisons, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2212–2220.

[6] T. Bogers and A. Van den Bosch, Comparing and evaluating information retrieval algorithms for news recommendation, in: *Proceedings of the 2007 ACM conference on Recommender systems*, 2007, pp. 141–144.

[7] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: *Neural Information Processing Systems (NIPS)*, 2013, pp. 1–9.

[9] H.L. Borges and A.C. Lorena, A survey on recommender systems for news data, in: *Smart Information and Knowledge Management*, Springer, 2010, pp. 129–151.

[10] J. Borràs, A. Moreno and A. Valls, Intelligent tourism recommender systems: A survey, *Expert Systems with Applications* **41**(16) (2014), 7370–7389, Publisher: Elsevier.

[11] R. Burke, Hybrid recommender systems: Survey and experiments, *User modeling and user-adapted interaction* **12**(4) (2002), 331–370, Publisher: Springer.

[12] R. Burke, N. Sonboli and A. Ordonez-Gauger, Balanced neighborhoods for multi-sided fairness in recommendation, in: *Conference on Fairness, Accountability and Transparency*, PMLR, 2018, pp. 202–214.

[13] H. Cai, V.W. Zheng and K.C.-C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, *IEEE Transactions on Knowledge and Data Engineering* **30**(9) (2018), 1616–1637, Publisher: IEEE.

[14] I. Cantador and P. Castells, Semantic contextualisation in a news recommender system (2009).

[15] I. Cantador, A. Bellogín and P. Castells, News@ hand: A semantic web approach to recommending news, in: *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer, 2008a, pp. 279–283.

[16] I. Cantador, A. Bellogín and P. Castells, Ontology-based personalised and context-aware recommendations of news items, in: *2008 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology*, Vol. 1, IEEE, 2008b, pp. 562–565.

[17] I. Cantador, P. Castells and A. Bellogín, An enhanced semantic layer for hybrid recommender systems: Application to news recommendation, *International Journal on Semantic Web and Information Systems (IJSWIS)* **7**(1) (2011), 44–78.

[18] I. Cantador, M. Szomszor, H. Alani, M. Fernández and P. Castells, Enriching ontological user profiles with tagging history for multi-domain recommendations (2008).

[19] Y. Cao, X. Wang, X. He, Z. Hu and T.-S. Chua, Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences, in: *The world wide web conference*, 2019, pp. 151–161.

[20] M. Capelle, F. Frasincar, M. Moerland and F. Hogenboom, Semantics-based news recommendation, in: *Proceedings of the 2nd international conference on web intelligence, mining and semantics*, 2012, pp. 1–9.

[21] M. Capelle, F. Hogenboom, A. Hogenboom and F. Frasincar, Semantic news recommendation using Wordnet and Bing similarities, in: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 296–302.

[22] M. Capelle, M. Moerland, F. Hogenboom, F. Frasincar and D. Vandic, Bing-SF-IDF+ a hybrid semantics-driven news recommender, in: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015, pp. 732–739.

[23] R. Caruana, Multitask learning, *Machine learning* **28**(1) (1997), 41–75, Publisher: Springer.

[24] X. Chen, Y. Zhang and Z. Qin, Dynamic explainable recommendation based on neural attentive models, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 53–60, Issue: 01.

[25] Q. Chu, G. Liu, H. Sun and C. Zhou, Next News Recommendation via Knowledge-Aware Sequential Model, in: *Chinese Computational Linguistics*, M. Sun, X. Huang, H. Ji, Z. Liu and Y. Liu, eds, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2019, pp. 221–232. ISBN 978-3-030-32381-3. doi:10.1007/978-3-030-32381-3$_1$8.

[26] R.G. Crespo, O.S. Martínez, J.M.C. Lovelle, B.C.P. García-Bustelo, J.E.L. Gayo and P.O. De Pablos, Recommendation system based on user interaction data applied to intelligent electronic books, *Computers in human behavior* **27**(4) (2011), 1445–1449, Publisher: Elsevier.

[27] A.S. Das, M. Datar, A. Garg and S. Rajaram, Google news personalization: scalable online collaborative filtering, in: *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 271–280.

[28] J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters, *Communications of the ACM* **51**(1) (2008), 107–113, Publisher: ACM New York, NY, USA.

[29] D. Doychev, A. Lawlor, R. Rafter and B. Smyth, An analysis of recommender algorithms for online news, in: *CLEF 2014 Conference and Labs of the Evaluation Forum: Information Access Evaluation Meets Multilinguality, Multimodality and Interaction, 15-18 September 2014, Sheffield, United Kingdom*, 2014, pp. 177–184.

[30] D. Doychev, R. Rafter, A. Lawlor and B. Smyth, News recommenders: Real-time, real-life experiences, in: *International*

Conference on User Modeling, Adaptation, and Personalization, Springer, 2015, pp. 337–342.

[31] B. Drury, J. Almeida and M. Morais, Magellan: An adaptive ontology driven "breaking financial news" recommender, in: *6th Iberian Conference on Information Systems and Technologies (CISTI 2011)*, IEEE, 2011, pp. 1–6.

[32] S.K. Dwivedi and C. Arya, A survey of news recommendation approaches, in: *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, IEEE, 2016, pp. 1–6.

[33] V.P. Dwivedi, C.K. Joshi, T. Laurent, Y. Bengio and X. Bresson, Benchmarking graph neural networks, *arXiv preprint arXiv:2003.00982* (2020).

[34] L. Ehrlinger and W. Wöß, Towards a Definition of Knowledge Graphs, *SEMANTiCS (Posters, Demos, SuCCESS)* **48** (2016), 1–4, Publisher: Citeseer.

[35] G. Farnadi, P. Kouki, S.K. Thompson, S. Srinivasan and L. Getoor, A fairness-aware hybrid recommender system, *arXiv preprint arXiv:1809.09030* (2018).

[36] F. Frasincar, J. Borsje and L. Levering, A semantic web-based approach for building personalized news services, *International Journal of E-Business Research (IJEBR)* **5**(3) (2009), 35–53.

[37] F. Frasincar, W. IJntema, F. Goossen and F. Hogenboom, A semantic approach for news recommendation, in: *Business Intelligence Applications and the Web: Models, Systems and Technologies*, IGI Global, 2012, pp. 102–121.

[38] J. Gao, X. Xin, J. Liu, R. Wang, J. Lu, B. Li, X. Fan and P. Guo, Fine-grained deep knowledge-aware network for news recommendation with self-attention, in: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, IEEE, 2018, pp. 81–88.

[39] J. Gao, X. Wang, Y. Wang and X. Xie, Explainable recommendation through attentive multi-view learning, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 3622–3629, Issue: 01.

[40] Y. Gao, Y.-F. Li, Y. Lin, H. Gao and L. Khan, Deep Learning on Knowledge Graph for Recommender System: A Survey, *arXiv preprint arXiv:2004.00387* (2020).

[41] F. Getahun, J. Tekli, R. Chbeir, M. Viviani and K. Yetongnon, Relating RSS news/items, in: *International Conference on Web Engineering*, Springer, 2009, pp. 442–452.

[42] S. Givon and V. Lavrenko, Predicting social-tags for cold start book recommendations, in: *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 333–336.

[43] Google, *Freebase Data Dumps*, February 1, 2021 edn, 2021. https://developers.google.com/freebase/data.

[44] F. Goossen, W. IJntema, F. Frasincar, F. Hogenboom and U. Kaymak, News personalization using the CF-IDF semantic recommender, in: *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, 2011, pp. 1–12.

[45] J.A. Gulla, L. Zhang, P. Liu, Özgöbek and X. Su, The adressa dataset for news recommendation, in: *Proceedings of the international conference on web intelligence*, 2017, pp. 1042–1048.

[46] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong and Q. He, A survey on knowledge graph-based recommender systems, *IEEE Transactions on Knowledge and Data Engineering* (2020), Publisher: IEEE.

[47] M. Harandi and J.A. Gulla, Survey of User Profiling in News Recommender Systems, in: *INRA@ RecSys*, 2015, pp. 20–26.

[48] T. Hofmann, Latent semantic models for collaborative filtering, *ACM Transactions on Information Systems (TOIS)* **22**(1) (2004), 89–115, Publisher: ACM New York, NY, USA.

[49] L. Hu, C. Li, C. Shi, C. Yang and C. Shao, Graph neural news recommendation with long-term and short-term interest modeling, *Information Processing & Management* **57**(2) (2020a), 102142, Publisher: Elsevier.

[50] L. Hu, S. Xu, C. Li, C. Yang, C. Shi, N. Duan, X. Xie and M. Zhou, Graph neural news recommendation with unsupervised preference disentanglement, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020b, pp. 4255–4264.

[51] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta and J. Leskovec, Open graph benchmark: Datasets for machine learning on graphs, *arXiv preprint arXiv:2005.00687* (2020c).

[52] X. Huang, Q. Fang, S. Qian, J. Sang, Y. Li and C. Xu, Explainable interaction-driven user modeling over knowledge graph for sequential recommendation, in: *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 548–556.

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

[54] W. IJntema, F. Goossen, F. Frasincar and F. Hogenboom, Ontology-based news recommendation, in: *Proceedings of the 2010 EDBT/ICDT Workshops*, 2010, pp. 1–6.

[55] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, *Recommender systems: an introduction*, Cambridge University Press, 2010.

[56] G. Ji, S. He, L. Xu, K. Liu and J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 687–696.

[57] J.J. Jiang and D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, *arXiv preprint cmp-lg/9709008* (1997).

[58] K. Joseph and H. Jiang, Content based news recommendation via shortest entity distance over knowledge graphs, in: *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 690–699.

[59] M. Karimi, D. Jannach and M. Jugovac, News recommender systems–Survey and roads ahead, *Information Processing & Management* **54**(6) (2018), 1203–1227.

[60] Y. Kim, Convolutional Neural Networks for Sentence Classification, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751. doi:10.3115/v1/D14-1181.

[61] C. Leacock and M. Chodorow, Combining local context and WordNet similarity for word sense identification, *WordNet: An electronic lexical database* **49**(2) (1998), 265–283.

[62] D. Lee, B. Oh, S. Seo and K.-H. Lee, News Recommendation with Topic-Enriched Knowledge Graphs, in: *Proceedings of the 29th ACM International Conference on Information*

& Knowledge Management, Association for Computing Machinery, New York, NY, USA, 2020, pp. 695–704. ISBN 978-1-4503-6859-9. https://doi.org/10.1145/3340531.3411932.

[63] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer et al., Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia, *Semantic web* **6**(2) (2015), 167–195, Publisher: IOS Press.

[64] L. Li, D.-D. Wang, S.-Z. Zhu and T. Li, Personalized news recommendation: a review and an experimental investigation, *Journal of computer science and technology* **26**(5) (2011a), 754.

[65] L. Li, D. Wang, T. Li, D. Knox and B. Padmanabhan, Scene: a scalable two-stage personalized news recommendation system, in: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011b, pp. 125–134.

[66] M. Li and L. Wang, A Survey on Personalized News Recommendation Technology, *IEEE Access* **7** (2019), 145861–145879.

[67] Q. Li, X. Tang, T. Wang, H. Yang and H. Song, Unifying task-oriented knowledge graph learning and recommendation, *IEEE Access* **7** (2019), 115816–115828, Publisher: IEEE.

[68] D. Lin et al., An information-theoretic definition of similarity., in: *Icml*, Vol. 98, 1998, pp. 296–304, Issue: 1998.

[69] Y. Lin, Z. Liu, M. Sun, Y. Liu and X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29, 2015, Issue: 1.

[70] D. Liu, J. Lian, S. Wang, Y. Qiao, J.-H. Chen, G. Sun and X. Xie, KRED: Knowledge-Aware Document Representation for News Recommendations, in: *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 200–209.

[71] J. Liu, P. Dolan and E.R. Pedersen, Personalized news recommendation based on click behavior, in: *Proceedings of the 15th international conference on Intelligent user interfaces*, 2010, pp. 31–40.

[72] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma and X. Ren, Jointly learning explainable rules for recommendation with knowledge graph, in: *The World Wide Web Conference*, 2019, pp. 1210–1221.

[73] V. Maidel, P. Shoval, B. Shapira and M. Taieb-Maimon, Evaluation of an ontology-content based filtering method for a personalized newspaper, in: *Proceedings of the 2008 ACM conference on Recommender systems*, 2008, pp. 91–98.

[74] G.A. Miller, WordNet: a lexical database for English, *Communications of the ACM* **38**(11) (1995), 39–41, Publisher: ACM New York, NY, USA.

[75] M. Moerland, F. Hogenboom, M. Capelle and F. Frasincar, Semantics-based news recommendation with SF-IDF+, in: *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, 2013, pp. 1–8.

[76] J. Möller, D. Trilling, N. Helberger and B. van Es, Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity, *Information, Communication & Society* **21**(7) (2018), 959–977.

[77] X. Ning and G. Karypis, Multi-task learning for recommender system, in: *Proceedings of 2nd Asian Conference on Machine Learning*, JMLR Workshop and Conference Proceedings, 2010, pp. 269–284.

[78] K. O'Hara and D. Stevens, Echo chambers and online radicalism: Assessing the Internet's complicity in violent extremism, *Policy & Internet* **7**(4) (2015), 401–422, Publisher: Wiley Online Library.

[79] Y. Pang, J. Tong, Y. Zhang and Z. Wei, DACNN: Dynamic Attentive Convolution Neural Network for News Recommendation, in: *Proceedings of the 2020 5th International Conference on Mathematics and Artificial Intelligence*, 2020, pp. 161–166.

[80] E. Pariser, *The filter bubble: What the Internet is hiding from you*, Penguin UK, 2011.

[81] D.H. Park, H.K. Kim, I.Y. Choi and J.K. Kim, A literature review and classification of recommender systems research, *Expert systems with applications* **39**(11) (2012), 10059–10072, Publisher: Elsevier.

[82] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semantic web* **8**(3) (2017), 489–508, Publisher: IOS Press.

[83] Y. Qian, P. Zhao, Z. Li, J. Fang, L. Zhao, V.S. Sheng and Z. Cui, Interaction Graph Neural Network for News Recommendation, in: *International Conference on Web Information Systems Engineering*, Springer, 2019, pp. 599–614.

[84] J. Qin and P. Lu, Application of News Features in News Recommendation Methods: A Survey, in: *International Conference of Pioneering Computer Scientists, Engineers and Educators*, Springer, 2020, pp. 113–125.

[85] J. Rao, A. Jia, Y. Feng and D. Zhao, Personalized news recommendation using ontologies harvested from the web, in: *International conference on web-age information management*, Springer, 2013, pp. 781–787.

[86] R. Ren, L. Zhang, L. Cui, B. Deng and Y. Shi, Personalized financial news recommendation algorithm based on ontology, *Procedia Computer Science* **55** (2015), 843–851.

[87] S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv:1205.2618* (2012).

[88] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, *arXiv preprint cmp-lg/9511007* (1995).

[89] F. Ricci, L. Rokach and B. Shapira, Recommender Systems: Introduction and Challenges, in: *Recommender Systems Handbook*, F. Ricci, L. Rokach and B. Shapira, eds, Springer US, Boston, MA, 2015, pp. 1–34. doi:10.1007/978-1-4899-7637-6$_1$.

[90] S. Ruder, An overview of multi-task learning in deep neural networks, *arXiv preprint arXiv:1706.05098* (2017).

[91] D. Ruffinelli, S. Broscheit and R. Gemulla, You can teach an old dog new tricks! on training knowledge graph embeddings, in: *International Conference on Learning Representations*, 2019.

[92] G. Salton and C. Buckley, Term-weighting approaches in automatic text retrieval, *Information processing & management* **24**(5) (1988), 513–523, Publisher: Elsevier.

[93] G. Salton, A. Wong and C.-S. Yang, A vector space model for automatic indexing, *Communications of the ACM* **18**(11) (1975), 613–620, Publisher: ACM New York, NY, USA.

[94] P. Shoval, V. Maidel and B. Shapira, An ontology-content-based filtering method (2008), Publisher: Institute of Information Theories and Applications FOI ITHEA.

[95] Y. Song, S. Shi, J. Li and H. Zhang, Directional skip-gram: Explicitly distinguishing left and right context for word embeddings, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 175–180.

[96] F.M. Suchanek, G. Kasneci and G. Weikum, Yago: a core of semantic knowledge, in: *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

[97] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo, J. Zhang and R. Burke, Research commentary on recommendations with side information: A survey and research directions, *Electronic Commerce Research and Applications* **37** (2019), 100879, Publisher: Elsevier.

[98] M. ter Hoeve, A. Schuth, D. Odijk and M. de Rijke, Faithfully explaining rankings in a news recommender system, *arXiv preprint arXiv:1805.05447* (2018).

[99] R. Troncy, Bringing the IPTC news architecture into the semantic web, in: *International Semantic Web Conference*, Springer, 2008, pp. 483–498.

[100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, in: *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[101] D. Vrandečić and M. Krötzsch, Wikidata: a free collaborative knowledgebase, *Communications of the ACM* **57**(10) (2014), 78–85, Publisher: ACM New York, NY, USA.

[102] H. Wang, F. Zhang, X. Xie and M. Guo, DKN: Deep knowledge-aware network for news recommendation, in: *Proceedings of the 2018 world wide web conference*, 2018a, pp. 1835–1844.

[103] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie and M. Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018b, pp. 417–426.

[104] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie and M. Guo, Exploring High-Order User Preference on the Knowledge Graph for Recommender Systems, *ACM Transactions on Information Systems* **37**(3) (2019a), 1–26. doi:10.1145/3312738.

[105] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li and Z. Wang, Knowledge-aware graph neural networks with label smoothness regularization for recommender systems, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019b, pp. 968–977.

[106] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie and M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: *The World Wide Web Conference*, 2019c, pp. 2000–2010.

[107] X. Wang, D. Wang, C. Xu, X. He, Y. Cao and T.-S. Chua, Explainable reasoning over knowledge graphs for recommendation, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019d, pp. 5329–5336, Issue: 01.

[108] X. Wang, X. He, Y. Cao, M. Liu and T.-S. Chua, Kgat: Knowledge graph attention network for recommendation, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019e, pp. 950–958.

[109] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28, 2014, Issue: 1.

[110] C. Wu, F. Wu, X. Wang, Y. Huang and X. Xie, Fairness-aware News Recommendation with Decomposed Adversarial Learning, *arXiv preprint arXiv:2006.16742* (2020a).

[111] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu et al., Mind: A large-scale dataset for news recommendation, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020b, pp. 3597–3606.

[112] S. Wu, W. Zhang, F. Sun and B. Cui, Graph Neural Networks in Recommender Systems: A Survey, *arXiv preprint arXiv:2011.02260* (2020c).

[113] Z. Wu and M. Palmer, Verb Semantics and Lexical Selection, in: *32nd Annual Meeting of the Association for Computational Linguistics*, 1994, pp. 133–138.

[114] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo and Y. Zhang, Reinforcement knowledge graph reasoning for explainable recommendation, in: *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 285–294.

[115] S. Yao and B. Huang, Beyond parity: fairness objectives for collaborative filtering, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 2925–2934.

[116] R. Ying, D. Bourgeois, J. You, M. Zitnik and J. Leskovec, Gnnexplainer: Generating explanations for graph neural networks, *Advances in neural information processing systems* **32** (2019), 9240, Publisher: NIH Public Access.

[117] K. Zhang, X. Xin, P. Luo and P. Guot, Fine-grained news recommendation by fusing matrix factorization, topic analysis and knowledge graph representation, in: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2017, pp. 918–923.

[118] S. Zhang, L. Yao, A. Sun and Y. Tay, Deep learning based recommender system: A survey and new perspectives, *ACM Computing Surveys (CSUR)* **52**(1) (2019), 1–38, Publisher: ACM New York, NY, USA.

[119] Y. Zhang and Q. Yang, A survey on multi-task learning, *arXiv preprint arXiv:1707.08114* (2017).

[120] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li and M. Sun, Graph neural networks: A review of methods and applications, *arXiv preprint arXiv:1812.08434* (2018).

[121] Q. Zhu, X. Zhou, Z. Song, J. Tan and L. Guo, Dan: Deep attention neural network for news recommendation, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 5973–5980, Issue: 01.

[122] Understand Your World with Bing, 2013. https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/.

[123] Özgöbek, J.A. Gulla and R.C. Erdur, A Survey on Challenges and Methods in News Recommendation., in: *WEBIST (2)*, 2014, pp. 278–285.