

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 behavior. 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. We propose a taxonomy that divides the models into three categories: neural methods, non-neural entity-centric methods, and non-neural path-based methods. Moreover, the underlying recommendation algorithms, as well as their evaluations are analyzed. Lastly, open issues in the domain of knowledge-aware news recommendations are identified and potential research directions are 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 use 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 behavior. Recommender systems have proven successful over time in numerous domains [1], ranging from music (e.g. Spotify), movies (e.g. Netflix, MovieLens),

or books recommendation [2–4], to e-commerce (e.g. Amazon, eBay), travel and tourism [5], or research paper recommendation [6].

In comparison to these domains, news recommendation poses additional challenges which hinder a direct transfer of traditional recommendation techniques [7]. Firstly, the relevance of news changes quickly within short periods of time and is highly dependent on the time sensitiveness and popularity of articles [8, 9]. 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 [8]. 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 [10], the fact that usually user profiles are constrained to a single session [11],

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

and that their feedback is typically collected implicitly, from their reading behavior rather than explicitly provided during a session [12]. Additionally, news articles contain a large number of knowledge entities and common sense knowledge, which are not incorporated in conventional news recommendation methods [13].

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 [14]. 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 [15]. 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 [15] or [16], 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 analyze the news recommendation problem from various angles, including challenges and algorithmic approaches [7–9, 17–21], performance comparison in online news recommendation challenges [11, 22], user profiling [23], news features-based methods [24], or impact on content diversity [25]. However, the focus of these studies is not on the use of external knowledge resources. 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 used 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 non-neural and neural-based methods, where the former category further distinguishes between entity-centric and path-based methods.
2. This survey aims to provide a **comprehensive review** of recommender systems for the news domain which use 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 eight potential **future research directions** are proposed in terms of comparability of evaluations, scalability of systems, explainability and fairness of results, multilingual and multi-modal news recommenders, multi-task learning for recommendation, sequential and timely recommendations, and changing user preferences.

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 outlines the methodology used in this survey, including the search strategy, the sources, the inclusion and exclusion criteria, as well as the study execution process. Section 4 covers related work in news and knowledge-aware recommender systems. Section 5 introduces and defines commonly used notations and concepts, and analyses different aspects of knowledge-aware news recommenders. Section 6 classifies and discusses knowledge-aware news recommender systems, whereas Section 7 investigates various evaluation approaches adopted by the different models. Section 8 discusses open issues identified in the field. We close with a short summary in section 9.

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 behavior, in order to help individuals overcome information overload [26].

User's preferences are learned using either explicit (e.g. ratings) or implicit (e.g. browsing history) feedback [27]. 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 [28]. 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 [28]. 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) [29].

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 [20]. In addition, the news is characterized by **short shelf lives** and **high item churn**, as their relevance highly depends on the **recency** of articles, since users prefer reading about the latest events that took place [10, 20]. 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" [9]. For example, while readers might be concerned with news about the elections in a country for multiple days or even weeks, they 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 [23, 30]. On the other hand, highly popular news might affect a user's short-term interest, which changes more rapidly, within a short time span [30].

For example, a user might read several news articles related to GameStop's short squeeze after browsing the "latest news" section of a website that announced that Robinhood has limited the buying and trading of GameStop stocks.

In addition to the previously described challenges, in the news domain, users are usually not required to sign in and create profiles in order to read articles. Another related challenge is that the users rarely provide explicit feedback in terms of likes and ratings. In turn, this **consumption behavior** means that their profiles are either limited to a single session or tracked through browser cookies, and that **feedback is gathered implicitly** by analyzing the clicks stored in logs [11, 22]. 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 [8, 9].

Furthermore, users often read multiple news stories in a sequence [7]. Although **sequential consumption** is also characteristic of music items, the major difference lies in the fact that in the news domain readers do not want items to be repeated multiple times in a row, as may be the case with songs. Instead, they prefer being recommended either updates on ongoing stories, or completely different articles [7].

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 [20].

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 [8, 18]. In turn, over-specialized news suggestions can reduce the **diversity** and **novelty** of the content being shown to the readers, by decreasing both their exposure to diversified information from various sources, as well as their likelihood of discovering new content that is not highly similar or related to the previously consumed articles [18]. Over time, over-personalization of news recommenders might trap users into **filter bubbles**, namely states in which the recommended articles are concentrated only on a certain standpoint, thus narrowing the readers' perspective and reducing their free-

dom of choice regarding the news content they consume [31].

Another significant challenge for the news domain is the existence of **fake news**, namely fabricated articles that mimic true news media content, but whose information lacks credibility and accuracy, and aims to propagate misinformation [32]. In the context of news recommendation, this can be represented as a second dimension, namely trustworthiness, which is orthogonal to the actual recommendation fit [33], and which does not exist in other fields, such as movie or music recommendation. By working with large volumes of data whose credibility has not been checked, news recommendation algorithms can contribute to the spread of fake content from unreliable sources [34].

3. Methodology

As aforementioned, this survey aims to provide a comprehensive review of knowledge-aware news recommender systems. The following subsections will describe the methodology used for conducting the study. More specifically, we firstly present our search strategy, including the platforms and queries used to retrieve relevant publications. Afterwards, we discuss the criteria for including and excluding papers from our study, followed by the description of the selection process.

3.1. Search Strategy

The search strategy of our survey consists in defining a set of queries for retrieving relevant publications from a list of sources. The results are then de-duplicated, as explained in the following paragraphs.

3.1.1. Search Queries

We defined two queries, targeting the task of (*Q1*) *news recommendation* and the usage of (*Q2*) *external knowledge*, in order to collect relevant literature. Table 1 illustrates the search strings used for each of the two queries. Keywords meant to capture (*Q2*) *external knowledge* include multiple terms referring to widely used sources of knowledge, such as knowledge graphs or ontologies. As such, the results of query (*Q2*) are given by the union of the results of the corresponding search strings. Since we are interested only in news recommender systems that use a form of external knowledge, the final query used in the publications' search process represents the intersection of queries (*Q1*) and (*Q2*).

3.1.2. Sources

The following bibliographic databases and archives constitute the sources used for the literature search:

- DBLP¹
- ACM Digital Library²
- IEEE Xplore³
- Science Direct⁴
- Springer Link⁵
- Web of Science⁶

3.1.3. De-duplication

The results collected from the previously specified sources are then merged and de-duplicated in a three-fold process. Firstly, for all the publications retrieved during the keyword-based search, we gather the associated bibtex files produced by each of the digital libraries and store them using the Zotero⁷ bibliographic tool, which also performs automatic detection of duplicates based on the papers' metadata. Secondly, we serialize as a spreadsheet the retrieved publications and their metadata, including title, DOI, abstract, authors, publication venue, and date. Lastly, we use the spreadsheet to perform manual de-duplication of the papers which could not be detected by the bibliographic tool due to large differences in their metadata, such as the publication venue which can be reported by some digital libraries as the conference venue and referred to by others as Lecture Notes in Computer Science (LNCS). The manually de-duplicated results of the keyword-based search constitute the final list of papers used for selecting the relevant literature in the next step.

3.2. Selection Strategy

The publications retrieved during the keyword-based search need to be further filtered in order to eliminate false positives, which are irrelevant for the current survey. Consequently, a pre-defined set of inclusion criteria (Section 3.2.1) are applied to the retrieved papers in two stages, as described in Section 3.2.2.

3.2.1. Selection Criteria

The list of inclusion criteria displayed in Table 2 was developed based on the goals of the survey in order

¹<https://dblp.dagstuhl.de/>

²<https://dl.acm.org/>

³<https://ieeexplore.ieee.org/>

⁴<https://www.sciencedirect.com/>

⁵<https://link.springer.com/>

⁶<https://www.webofscience.com/>

⁷<https://www.zotero.org/>

Table 1
Search strings used in the search process

Query	Search Strings
(Q1) news recommendation	news recommend*
(Q2) external knowledge	knowledge base*, knowledge graph*, ontolog*, linked data*, semantic*

Table 2
Selection inclusion and exclusion criteria

Criteria	Inclusion Criteria (IC)	Exclusion Criteria (EC)
C1 Publication Date	The paper is published between 2008 and 2020.	The paper is published before 2008 or after 2020.
C2 Language	The paper is written in English.	The paper is written in a language different than English.
C3 Publication Type	The paper has been peer reviewed (e.g. conference or workshop proceedings, journal paper, book chapter). The paper is a primary study.	The paper has not been peer reviewed (e.g. theses, books, technical reports, (extended) abstracts, talks, presentations, tutorials, guidelines). The paper is a secondary study (e.g. systematic literature review, survey).
C4 Accessibility	The paper's content can be accessed from a technical university (e.g. University of Mannheim) without additional payment.	The paper's content cannot be accessed from a technical university (e.g. University of Mannheim) without additional payment.
C5 Duplicate	If the same system and results are presented in multiple publications of the same study (e.g. conference and journal paper), the most recent version of the study will be included. An exception is made in case the two versions of the study differ significantly (e.g. short paper and long paper), in which case both will be discussed.	Studies for which a newer or more complete version exists.
C6 Recommender System Scope	The paper presents a recommendation system applied only in the news domain, or in multiple domains, including the news one. The paper presents a system or algorithm mainly aimed at item recommendation.	The paper presents a recommendation system which is not applied in the news domain. The paper presents a system or algorithm which is not mainly aimed at item recommendation (e.g. an algorithm for improving the diversity of recommendations generated by another system).
C7 Use of External Knowledge	The paper presents a system that uses a form of external knowledge from a knowledge base.	The paper presents a system that does not use a form of external knowledge from a knowledge base.
C8 Interaction of Recommendation Model and External Knowledge Source	The external knowledge source is used to enhance the performance and accuracy of the news recommender system.	The external knowledge source and recommender system do not jointly target news recommendation (e.g. knowledge graph construction, where the new graph is evaluated on the downstream task of recommendation using an existing recommender system).
C9 Recommender System Evaluation	The paper presents a recommender system whose performance is evaluated on the task of news recommendation.	The paper only describes the theoretical idea of a recommender system or its implementation, but lacks an evaluation of the proposed system on the task of news recommendation.

to filter out irrelevant publications. Each criterion is composed of both an *inclusion criterion (IC)* and an *exclusion criterion (EC)*. A paper needs to fulfill all inclusion criteria to be selected for the study.

3.2.2. Selection Process

The study selection process is composed of two phases⁸. Firstly, relevant papers are pre-filtered based on their metadata. More specifically, the validity of criteria C1-C4 is assessed by examining the publications' language, publishing year, venue, type, keywords, title and abstract. The validity of the remaining criteria is also checked if the information contained in the metadata allows it. Papers not fulfilling all of these criteria are excluded from the rest of the study. Papers whose relevance cannot be determined solely from the metadata will be kept until the next stage of the selection process. In the second phase, the fulfillment of criteria C5-C9 is checked using the complete content of the pre-filtered publications. The papers meeting all the requirements of this phase will be included in the final set of publications for the survey. Table 3 shows the number of papers remaining after different stages of the selection process.

Table 3

Number of papers in different phases of the selection process

Phase	Number of Remaining Papers
Keyword-based search	716
De-duplication	680
Pre-filtering on metadata	62
Filtering on content	38

4. Related Work

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

4.1. News Recommender Systems

Several surveys on news recommender systems and corresponding issues have been conducted. A comparison and evaluation of content-based news recom-

menders are performed in [21]. Borges and Lorena [19] 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 [17].

Özgöbek et al. [8] 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. [9] 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. [20] 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 [18] analyzed 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 [18].

In comparison to the previous general studies, Harandi and Gulla [23] investigate and categorize approaches used for user profiling in news recommendation according to the problems addressed and the types of features used. Additionally, Qin and Lu [24] 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.

⁸The corresponding spreadsheets for both phases are available at: https://docs.google.com/spreadsheets/d/11NxjctjgQ5sAbbA2g2vKgqyUigg4t6S_H98-VsHJDWQ/edit?usp=sharing

Lastly, Feng et al. [35] conduct a systematic literature review of research published in the area of news recommendation in the past two decades. They firstly classify and discuss challenges from this domain according to the three main types of recommendation techniques. Various recommendation frameworks are then categorized according to application domain, such as social media-based, semantics-based, and mobile-based systems. Even though Feng et al. [35] briefly review a small number of semantic-based recommenders, their analysis is limited to older models, and does not include any of the newer approaches of the past five years. Furthermore, the authors briefly examine evaluation approaches and datasets used, before discussing which of the numerous challenges of news recommendation have been addressed by the surveyed recommenders [35].

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 that incorporate external knowledge to enhance the recommendations and to overcome the limitations of conventional recommendation techniques.

4.2. 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 [36], DBpedia [37], YAGO [38], Wikidata [39], Microsoft Satori [40]) or focusing on a particular field (e.g. Bio2RDF [41]) [14, 42]. In addition, such graphs can capture higher-order relations connecting entities with several related attributes [15].

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 [15]. 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 [16, 43]. 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 [43].

Guo et al. [15] 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 analyze how knowledge graphs are used 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 [15].

Recent advancements in deep learning techniques for graph data, in the form of Graph Neural Networks (GNN) [43, 44], have given rise to new knowledge-aware, deep recommender systems. Gao et al. [16] are the first to provide a comprehensive overview of GNN-based knowledge-aware deep recommender (GNN-KADR) systems, in which they analyze 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. [16] 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, that combines the feature information of a node's neighborhood 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. [16], 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 modeled in the framework [16]. The first subcategory creates multiple subgraphs for each relation type found in a node's neighborhood 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 [16]. 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 [16].

GNN-based recommender systems are investigated also by Wu et al. [43], 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. [43], 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. [45] consists of an extensive, systematic survey of recent advancements in recommender systems that use side information. The models, mostly hybrid techniques, are analyzed from two perspectives. On the one hand, Sun et al. [45] 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 [45].

In the surveys discussed above, knowledge-aware news recommender systems are rarely analyzed. 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.

5. 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.

5.1. 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 4, unless specified otherwise.

Table 4
Commonly used notations

Notations	Descriptions
$ \cdot $	Set size
\odot	Element-wise product
\oplus	Vector concatenation
\tanh	Hyperbolic tangent function
$\sigma(\cdot)$	Nonlinear transformation function
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
$\mathcal{U} = \{u_1, u_2, \dots, u_M\}$	Set of users
$\mathcal{V} = \{v_1, v_2, \dots, v_N\}$	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 1. An **ontology** is defined as a set of k concepts [46]:

$$\mathcal{O} = \{c_1, c_2, \dots, c_k\} \quad (1)$$

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

Definition 2. The **semantic neighborhood** of a concept c_i is defined as the set of concepts that are directly related to concept c_i , including itself [46]:

$$N(c_i) = \{c_1^i, c_2^i, \dots, c_k^i\} \quad (2)$$

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 [47]. Often, entities are assigned one or more types, defined by type statements of the form $\langle e, t \rangle$, where $e \in E$ and $t \in \mathcal{C}$.

5.2. 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 target function types or addressed challenges.

5.2.1. 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 [9].

5.2.2. 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 [37], Hudong encyclopedia [48]) or directly from news articles (i.e. financial domain ontology using information from Yahoo! Finance

[46]) – and controlled vocabularies used in the news domain, such as the IPTC News Codes⁹ [49].

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 [36] was initially launched by Metaweb in 2007, and later acquired by Google in 2010, before being shut down in 2015 [14]. The latest version of Freebase, available at Google's Data Dumps¹⁰ has been estimated to contain 1.9 billion triples [50]. Wikidata [39], a collaboratively edited knowledge graph, containing several language editions of Wikipedia, as well as data previously contained in Freebase [14], consists of 92 million items¹¹ and over 1174 million statements¹². DBpedia [37] 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 triples¹³.

WordNet [51], 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, troponymy, or entailment, is often used in knowledge-aware news recommender systems for word sense disambiguation. More specifically, each word 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 3.0¹⁴ contains 117,659 synsets and 206,941 word-sense pairs.

In the subgroup of commercial knowledge bases, Satori [40], the knowledge graph proposed by Microsoft, is the most often used 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 [14].

⁹<https://iptc.org/standards/newscodes/>

¹⁰<https://developers.google.com/freebase>

¹¹<https://www.wikidata.org/wiki/Wikidata:Statistics>

¹²<https://wikidata-todo.toolforge.org/stats.php>

¹³<https://www.dbpedia.org/resources/knowledge-graphs/>

¹⁴<https://wordnet.princeton.edu/documentation/wnstats7wn>

5.2.3. Structure of Knowledge Base

News recommendation models use knowledge bases by exploiting their different structures in order to extract either semantic, structural, or both types of information. A few knowledge-aware news recommender systems exploit only 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. A larger share of models however enriches the basic set of knowledge entities by expanding it with the neighborhoods of extracted entities in the knowledge graph and by considering the paths and relationships between entities (denoted as *entities+paths*). Another method for enhancing the set of concepts or entities extracted from a knowledge base is by taking into account its structure, namely the different types of relations between nodes in an ontology, such as synonymy or hyponymy relationships in semantic lexicons, or the distances between concepts, entities or classes (denoted as *concepts+KB structure* or *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*).

5.2.4. Target Function

Two main target functions 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.

5.2.5. 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 represen-

tations. Knowledge-aware recommender systems propose 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 se-

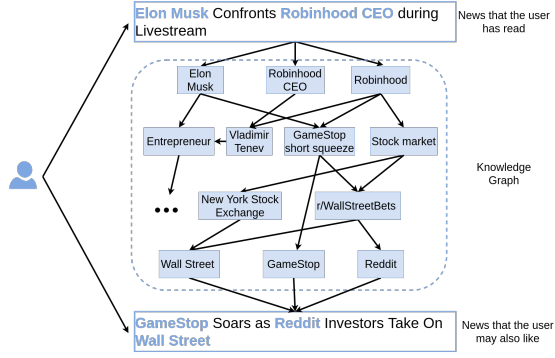


Fig. 1. Illustration of a knowledge graph-enhanced news recommender system (reproduced from [13]).

mantically similar, even if they are published in different sources and have different writing styles.

6. Knowledge-Aware News Recommendation Models

Knowledge-aware news recommendation systems can be classified into different categories based on how external knowledge is injected in the recommendation model, on the used structures of the knowledge base, as well on how latent representations of users and articles are computed. Our proposed taxonomy, illustrated in Figure 2, distinguishes between the methods based on how the latent representations are generated from entities and/or concepts in a knowledge base, i.e., Non-Neural Methods and methods based on neural networks (Section 6.3). We further split Non-Neural Methods into Entity-Centric (Section 6.1) and Path-Based (Section 6.2), depending on whether the recommendation approach defines the similarity between users and news articles based on distances between the concepts or entities from the knowledge base. To support readers reviewing the literature, the surveyed models are listed in Table 5 according to the aforementioned criteria.

Factorization models constitute some of the state-of-the-art recommendation techniques in various fields [88–90], and have already been adopted in the area of news recommendation [91, 92]. Moreover, latent factor models have also been adapted to support knowledge graphs in hybrid knowledge-aware recommendation engines [93–96]. Nonetheless, as it can be observed from Table 5, factorization models are rarely used by knowledge-aware news recommender systems. More specifically, only one model from the 39

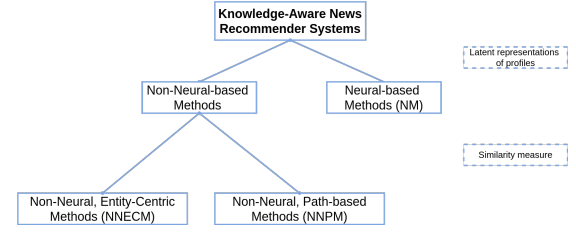


Fig. 2. The categorization of knowledge-aware news recommender systems. We divide existing frameworks into three categories, based firstly on how latent representations of user and news article profiles are generated, and secondly, on the type of similarity measure used.

surveyed ones uses matrix factorization (see Section 6.3). Hence, we have not added a dedicated subcategory for such methods to our taxonomy.

Recommenders based on factorization models are collaborative filtering-based approaches. However, this recommendation technique is the least adopted one in the domain of news recommendation [7]. Raza et al. [7] have shown that content-based methods are the most widely used recommendation techniques in this field, followed by hybrid approaches. This phenomenon can be explained by the challenges faced by recommender systems in the news domain, explained in Section 2, such as the lack of explicit feedback (i.e. ratings), or limited amount of data available for user profiling. In turn, this affects collaborative-filtering approaches such as factorization models, which rely on a large amount of information regarding the user-item interactions in order to generate accurate recommendations.

Comparison with existing taxonomies Several taxonomies have already been proposed for general knowledge-aware recommender systems. Gao et al. [16] categorize GNN-based knowledge-aware deep recommender systems based on the different types of the two basic components of the graph embedding module, namely the aggregator and the updater. In comparison to the work of Gao et al. [16], in the current survey, we neither limit our analysis to knowledge-aware deep news recommenders, nor to those based necessarily on GNNs. Hence, there is no overlap between the two taxonomies.

Another categorization of knowledge graph-based recommender systems divides models into different categories based on how they utilize the knowledge graph information, namely into embedding-based methods, path-based methods, and unified methods [15]. Embedding-based methods encode the knowledge graph information by means of knowledge graph embeddings and directly use it to enhance the repre-

Table 5

Overview of knowledge-aware news recommendation approaches. We present the model's category, abbreviated name, publishing year, type of recommendation technique, external knowledge resource and its used structures, target function, and 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 following abbreviations are used: NNECM=Non-Neural, Entity-Centric Methods, NNPM=Non-Neural, Path-Based Methods, NM=Neural Methods, RS=Recommender System, CB=Content-Based, CF=Collaborative Filtering, H=Hybrid, DO=Domain Ontology

Category	Model	Year	RS Type	KB	KB Structure	Target function	Addressed Challenge
NNECM	Semantic context-aware recommendation [52, 53]	2008	H	DO (IPTC + Wikipedia)	entities+paths	Item ranking	Capture semantic relatedness of concepts
	Social tags enriched recommendations [54]	2008	CB	Wordnet, Wikipedia	entities+paths	Item ranking	Data sparsity, cold-start problem
	Semantic relatedness [55]	2009	CB	DO (IPTC + Wikipedia)	entities+paths	Item ranking	Capture semantic relatedness of concepts
	RSR [46]	2010	CB	DO	entities+paths	Item ranking	Capture semantic meaning, faster computations, eliminate ambiguity
	CF-IDF [56]	2011	CB	DO	concepts/entities only	Item ranking	-
	Hybrid context-aware recommendation [53]	2011	H	DO (IPTC + Wikipedia)	entities+paths	Item ranking	Data sparsity, cold-start problem
	RSR 2 [57]	2012	CB	DO	entities+paths	Item ranking	Capture semantic meaning, faster computations, eliminate ambiguity
	SF-IDF [58]	2012	CB	WordNet	concepts/entities only	Item ranking	Capture semantic meaning
	SF-IDF+ [59]	2013	CB	WordNet	entities+paths	Item ranking	Capture semantic relationships between synsets
	Bing-SF-IDF [60]	2013	CB	WordNet	entities+paths	Item ranking	Handle knowledge entities
NNPM	Bing-SF-IDF+ [61]	2015	CB	WordNet	entities+paths	Item ranking	Capture semantic meaning
	Agarwal and Singhal [62]	2014	CB	DO (IPTC)	conceptys/entities only	Item ranking	-
	OF-IDF [63]	2015	CB	DO	entities+paths	Item ranking	Eliminate ambiguity
	CF-IDF+ [64]	2018	CB	DO	entities+paths	Item ranking	Handle relationship types between concepts
	Bing-CF-IDF+ [65]	2019	CB	DO	entities+paths	Item ranking	Handle knowledge entities
	Bing-CSF-IDF+ [66]	2020	CB	DO, WordNet	entities+paths	Item ranking	Capture semantic meaning, handle knowledge entities
	ePaper [67]	2008	CB	IPTC News Codes	concepts+KB structure	Item ranking	-
	Magellan [68]	2011	CB	DO	concepts+KB structure	Item ranking	-
	SS [58]	2012	CB	WordNet	concepts+KB structure	Item ranking	Capture semantic meaning
	BingSS [69]	2013	CB	WordNet	concepts+KB structure	Item ranking	Handle knowledge entities
NM	OBSM [70]	2013	CB	Ontologies based on DBpedia and Hadoop	concepts+KB structure	Item ranking	Eliminate ambiguity
	Kumar and Kulkarni [71]	2013	CB	Wikipedia	concepts+KB structure	Item ranking	-
	Werner and Cruz [72]	2013	CB	DO	concepts+KB structure	Item ranking	-
	BKSport [73]	2016	H	DO	concepts+KB structure	Item ranking	Capture semantic meaning
	SED [74]	2019	CB	Freebase	entities+KB structure	Item ranking	Cold-start problem
	CEIR [75]	2017	CF	Wikidata	entities+KB structure	CTR	Extend text-level information
	Colombo-Mendoza et al. [76]	2018	H	DO	entities+KB structure	Item ranking	-
	DKN [13]	2018	CB	Microsoft Satori	entities+KB structure	CTR	Extract knowledge-level news connections, extend text-level information with common sense
	Gao et al. [77]	2018	CB	KG	entities+KB structure	CTR	Extract knowledge-level news connections
	RippleNet [78]	2018	CB	Microsoft Satori	entities+KB structure	CTR	Data sparsity, cold-start problem
NM	RippleNet+agg [79]	2019	CB	Microsoft Satori	entities+KB structure	CTR	Data sparsity, cold-start problem
	MKR [80]	2019	CB	Microsoft Satori	entities+KB structure	CTR	Data sparsity, cold-start problem
	IGNN [81]	2019	CB	Wikipedia	entities+KB structure	CTR	Mine high-order connectivity of users and news
	Saskr [82]	2019	CB	News-specific KG	entities+KB structure	CTR	Extract knowledge-level news connections, diversity, cold-start problem, handle knowledge entities
	KRED [83]	2020	CB	Microsoft Satori	entities+KB structure	CTR	Handle knowledge entities
	TEKGR [84]	2020	CB	Microsoft Satori	entities+KB structure	CTR	Handle knowledge entities, capture topical relatedness between news
	KCNr [85]	2020	CB	Douban	entities+KB structure	CTR	Handle knowledge and content factors
	KG-RWSNM [86]	2020	H	Wikidata	entities+KB structure	CTR	Extract deep semantic features from news
	MUKG [87]	2020	CB	Microsoft Satori	entities+KB structure	CTR	Data sparsity, cold-start problem
							Mine high-order interactions between item and entities

sentations of users or items. This category is further split into models that construct knowledge graphs of items and their relations, extracted from a dataset or external knowledge base, and those that build user-item graphs, in which the users, items, and their attributes form the graph's nodes, while user-related and attribute-related relations constitute the edges [15]. In our survey, this category overlaps with the neural-based models, which use a form of knowledge graph embedding, as it will be explained in Section 6.3. However, we do not further differentiate between neural-based recommenders depending on how the underlying knowledge graph is created.

Guo et al.'s [15] second category of path-based methods includes those recommenders that leverage connectivity patterns of entities in the user-item graph. In this context, this category has similarities with our proposed non-neural, path-based methods. However, in our case, the connectivity patterns are exploited from any source of structured knowledge base and are not restricted to user-item graphs.

The third category proposed by Guo et al. [15], namely unified methods, incorporates models that combine the first two types of techniques by leveraging both the connectivity information, as well as the semantic representation of entities and relations. This class, containing the RippleNet [78] and RippleNet-agg [79] models, overlaps again with our neural-based methods due to the neural nature of latent representations used to profile items and users.

Lastly, Sun et al. [45] classify recommender systems on two dimensions. The first dimension is concerned with the recommendation technique and differentiates between memory-based methods, latent factor models, representation learning models, and deep learning models. The second dimension focuses on the type of side information used, namely structural data (i.e. flat, network, hierarchical features, and knowledge graphs) and non-structural data, in the form of text, image, and video features. According to this categorization scheme, models classified by Sun et al. [45] under deep learning methods that use knowledge graphs as side information would correspond to neural-based methods in our taxonomy.

The following subsections analyze the knowledge-aware recommender systems presented in Table 5 according to the taxonomy introduced above. For each category of models, the overall framework, as well as representative models are investigated.

6.1. Entity-Centric Methods (Non-Neural)

Recommender systems classified in this category represent the profiles of users and news articles using latent representations generated from concepts and/or entities in a knowledge base using non-neural methods. Generally, such representations are computed using a Vector Space Model [97], most often variants of the Term Frequency-Inverse Document Frequency (TF-IDF) model [98], modified to take into consideration side information from a knowledge base. The similarity between articles and the preference of a user for a candidate article are determined using different semantic or non-semantic similarity metrics.

6.1.1. Overall Framework

Non-neural, entity-centric methods first create a vector representation of both the target article and the user profile, where the latter consists of the user's reading history. Afterwards, the models compute the similarity between the two profiles and recommend a list of the top N articles whose similarity scores exceed a predefined threshold. As such, the majority of techniques listed here adopt a content-based recommendation approach. We analyze these systems in terms of three 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.
- **Similarity metric.** 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.

6.1.2. Representative Models

In this subsection, we discuss 11 representative non-neural, entity-centric recommendation techniques.

Cantador et al. [52, 53] developed a **semantic context-aware recommendation** model which aims to contextualize 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 $\mathbf{u} = (w_1^u, w_2^u, \dots, w_q^u)$, and $\mathbf{v} = (w_1^v, w_2^v, \dots, w_p^v)$, respectively. The concepts in the user's vector representation are weighted with weights

$w_i'' \in [-1, 1]$, 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 $w_i'' \in [0, 1]$ place the article's representation in the same vector space as the user's preferences [53].

A personalized 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. [53] 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. (3), comprises a set of weighted concepts from a domain ontology, collected from the articles accessed by the user during a session.

$$\mathbf{C}_u^t[c_i] = \xi \cdot \mathbf{C}_u^{t-1}[c_i] + (1 - \xi) \cdot \text{Req}_u^t[c_i] \quad (3)$$

where $\text{Req}_u^t \in [0, 1]^{|\mathcal{O}|}$ 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 [53]. 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. [53] 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, $\mathbf{P}_u = \{c_i'' \in \mathcal{O} | w_k'' \neq 0\}$, is expanded as follows:

$$\mathbf{EP}_u[c_j] = \begin{cases} \mathbf{P}_u[c_j], & \text{if } \mathbf{P}_u[c_j] > 0 \\ R(\{\mathbf{EP}_u[c_i] \cdot \text{power}(c_i)\}_{c_i \in \mathcal{O}, r(c_i, c_j)}), & \text{otherwise} \end{cases} \quad (4)$$

where $\text{power}(c_i) \in [0, 1]$ represents a propagation power assigned to each concept and $R(\mathbf{X}) = \sum_{S \subseteq \mathbb{N}_n} \{(-1)^{|S|+1} \times \prod_{i \in S} x_i\}$, 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: $pref_c(v, u) = \lambda \cdot pref(v, \mathbf{EP}_u) + (1 - \lambda) \cdot pref(v, \mathbf{EC}_u)$. 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, whereas 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, Cantandor et al. [53] also 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 [53]. More specifically, each ontology concept c_i'' occurring in a user's reading history is represented as a vector of weights measuring its importance for the user, namely $\mathbf{c}_i'' = (w_{1,i}, w_{2,i}, \dots, w_{M,i}) \in [-1, 1]^M$. 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 $\text{sim}(u, C_q) = \frac{\sum_{c_i^u \in C_q} w_1^u}{|C_q|}$, where c_i is the concept associated with the w_i^u element in the user's preference vector.

Cantandor et al. [53] propose two recommendation models that use 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. (5):

$$\text{pref}(u, v) = \sum_q \text{nsim}(v, C_q) \sum_{y \neq u} \text{nsim}_q(u, y) \cdot \text{sim}_q(y, v) \quad (5)$$

Here $\text{sim}(v, C_q) = \frac{\sum_{c_i \in C_q} w_i^v}{\sqrt{|C_q|}}$ represents the similarity between item v and cluster C_q , and $\text{nsim}(v, C_q)$ denotes the normalized similarity over the set of all clusters. Moreover $\text{sim}_q(u, y)$ and $\text{nsim}_q(u, y)$ are the single and normalized 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:

$$\text{pref}(u, v) = \sum_{y \neq u} \text{nsim}_q(u, y) \cdot \text{sim}_q(y, v) \quad (6)$$

The recommendations corresponding to the cluster to which the user has the highest similarity will be suggested (q maximizes $\text{sim}(v, C_q)$ in Eq. (6)).

The same context-aware and multi-facet, group-oriented hybrid recommendations are also adopted by Cantandor et al. [54] 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.

Concept Frequency - Inverse Document Frequency (**CF-IDF**) [56] constitutes a variant of the TF-IDF weighting scheme that uses concepts instead of terms in order to represent news articles. In the framework proposed by Goossen et al. [56], the profile of a user u consists of a set of q concepts from an ontology, namely $u = \{c_1^u, c_2^u, \dots, c_q^u\}, c_i^u \in \mathcal{O}$. 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 $c^u = \{v_1, v_2, \dots, v_k\}$. An article is thus composed of a set of p concepts occurring in it, denoted as $v = \{c_1^n, c_2^n, \dots, c_p^n\}, c^n \in \mathcal{O}$.

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. (7). Analogously, the article's profile is computed according to Eq. (8).

$$\mathbf{u} = [\langle c_1^u, w_1^u \rangle, \dots, \langle c_q^u, w_q^u \rangle] \quad (7)$$

$$\mathbf{v} = [\langle c_1^n, w_1^n \rangle, \dots, \langle c_p^n, w_p^n \rangle] \quad (8)$$

The CF-IDF weights are computed similarly to TF-IDF weights. Firstly, the *Concept Frequency* $\text{cf}_{i,j}$ 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. (9).

$$\text{cf-idf}_{i,j} = \text{cf}_{i,j} \times \text{idf}_i = \frac{n_{i,j}}{\sum_k n_{k,j}} \times \log \frac{|D|}{|d : c_i \in d|} \quad (9)$$

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. [58] 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. [58] represent the user's and article's profile as sets of WordNet synsets of the terms appearing in the news article. Mathematically, the news item's profile is represented as:

$$v = \{s_1^n, s_2^n, \dots, s_p^n\} \quad (10)$$

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:

$$u = \bigcup_{q \in Q} v_q = \bigcup_{q \in Q} \{s_1^u, s_2^u, \dots, s_q^u\} \quad (11)$$

The synsets in both profiles are weighted using SF-IDF weights, obtained from TF-IDF by replacing terms with synsets s , i.e. $\text{sf-idf}_{s,d} = \text{sf}_{s,d} \times \text{idf}_{s,d}$. 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+** [59] 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. (12), 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 .

$$S(s) = s + \bigcup_{r \in R(s)} r(s) \quad (12)$$

Hence, the item's and user's profiles, v and u , are extended according to Eqs. (13) and 14, respectively.

$$v = \{S(s_1^n), S(s_2^n), \dots, S(s_p^n)\} \quad (13)$$

$$u = \{S(s_1^u), S(s_2^u), \dots, S(s_q^u)\} \quad (14)$$

Furthermore, SF-IDF+ not only uses extended synsets instead of synsets, as is the case for the SF-IDF model,

but also assigns different weights w_r to the relationship r connecting a synset with its semantically related synset, as per Eq. (15).

$$\text{sf-idf+}_{s,d,r} = \text{sf}_{s,d} \times \text{idf}_{s,d} \times w_r \quad (15)$$

A similar strategy to the one proposed in SF-IDF+ is also adopted by **CF-IDF+** [64], an extension of the original CF-IDF model in which the initial set of identified concepts is expanded with related neighbors. 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. [61] 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 co-occur on websites [61]. The Bing-SF-IDF+ similarity score combines two elements, namely the Bing component which measures the similarity between pairs of named entities¹⁵, and the SF-IDF+ component which computes the similarity between synsets.

The SF-IDF+ profiles and weights are built and calculated according to Eqs. (12)-(15). 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:

$$v = \{e_1^n, e_2^n, \dots, e_k^n\} \quad (16)$$

$$u = \{e_1^u, e_2^u, \dots, e_l^u\} \quad (17)$$

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. (18).

$$\mathbf{V} = (\langle e_1^n, e_1^u \rangle, \dots, \langle e_k^n, e_l^u \rangle), \forall e^n \in v, e^u \in u. \quad (18)$$

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

¹⁵The Bing similarity is comparable to the Normalized Google Distance [99].

by querying Bing contain either one or both of the entities in a pair. A page rank-based Point-Wise Mutual Information (PMI) [100] 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 [61]. PMI assumes independence between the two named entities and is based on their marginal probabilities, as illustrated in Eq. (19).

$$\text{sim}_{PMI}(e^n, e^u) = \log \frac{\frac{c(e^n, e^u)}{N}}{\frac{c(e^n)}{N} \times \frac{c(e^u)}{N}} \quad (19)$$

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:

$$\text{sim}_{Bing}(\mathbf{V}) = \frac{\sum_{(e^n, e^u) \in \mathbf{V}} \text{sim}_{PMI}(e^n, e^u)}{|\mathbf{V}|} \quad (20)$$

The SF-IDF+ similarity $\text{sim}_{sf-idf+}(u, v)$, and the Bing similarity $\text{sim}_{Bing}(u, v)$ 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. (22).

$$\text{sim}_{Bing-sf-idf+}(u, v) = \alpha \times \overline{\text{sim}}_{Bing}(u, v) + (1 - \alpha) \times \overline{\text{sim}}_{sf-idf+}(u, v) \quad (21)$$

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

This approach of enhancing semantics-driven recommender systems with named entity similarities using the Bing page counts are prototypical also for other models, such as **Bing-SF-IDF** [60], **Bing-CF-IDF+** [65], or **Bing-CSF-IDF+** [66].

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. [63]. 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 $\mathbf{v} = (w_1^n, w_2^n, \dots, w_p^n)$, where the weights are computed as follows:

$$w_{i,j} = \text{OF}_{i,j} \times \text{IDF}_i = \frac{n_{i,j}}{\max_i n_{i,j}} \times \log \frac{N}{n_i} \quad (22)$$

In Eq. (22), $n_{i,j}$ is the number of occurrences of concept c_i in article j , and $1 \leq i \leq p$, 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. [63], 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:

$$\mathbf{u} = \sum_m (\alpha + \mathbf{S}^\alpha) + \sum_n (\beta + \mathbf{S}^\beta) + \sum_l (\gamma + \mathbf{S}^\gamma) \quad (23)$$

In Eq. (23), 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.

The **semantic relatedness** model of Getahun et al. [55] 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:

$$\mathbf{v}_l = (\langle c_1^l, w_1^l \rangle, \dots, \langle c_p^l, w_p^l \rangle), l \in \{i, j\} \quad (24)$$

In comparison to item profiles of models such as CF-IDF (Eq. (8)), 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:

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

where

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

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

The Ranked Semantic Recommendation (RSR) [46] 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 $r(c_i) = \{c_1^i, c_2^i, \dots, c_k^i\}$, 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 $u_R = u \cup R$.

Another assumption underlying RSR [46] 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 uses 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 to capture the type of relationship between concepts:

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

The final rank of every concept in the user's extended profile, denoted $\text{Rank}(e_j) = \sum_{i=1}^q r_{ij}$, 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:

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

The Ranked Semantic Recommendation 2 (RSR 2) [57] 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 use related concept instances such as *Open AI*, *SpaceX*, *Tesla, Inc.*, or *Neuralink* to represent the ar-

ticle. Thus, the original article profile is extended by the set of related concepts, namely $v_E = v \cup E$, 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.

6.1.3. Summary

Non-neural, entity-centric news recommendation techniques are summarized in terms of three 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 uses concepts extracted from the news and contained in a domain ontology to represent articles. RSR and RSR2 also use ontological concepts, as well as concepts related to them in the ontology. 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 majority of 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. In contrast, semantic context-aware techniques use weights in the range $[-1, 1]$ to

denote the users' likes and dislikes. Moreover, the semantic relatedness model defines concept weights in terms of semantic enclosure which considers the global neighborhood 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.

- **Similarity metric.** Cosine similarity is often employed to determine the preference of a user for an unread news article in context-aware models, as well as in models using a variant of the TF-IDF model. The latter category extends the similarity measure with the Bing similarity metric when named entities are taken into account through Bing page counts. 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.

6.2. Path-based Methods (Non-Neural)

The profiles of users and news articles in non-neural, path-based recommendation methods are represented using concepts or entities from a knowledge base. Similar to the models in the previous section, some of the recommendations approaches classified here generate latent representations of these concepts or entities using non-neural methods. However, in contrast to non-neural, entity-centric recommenders, path-based ones define the user-item and item-item similarities using metrics that take into account the distance between concepts and/or entities from the knowledge base.

6.2.1. 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 $v = \{\langle c_1^n, w_1^n \rangle, \dots, \langle c_p^n, w_p^n \rangle\}$, where $c_i^n \in \mathcal{O}$, w_i^n is the weight of concept c_i^n ($1 \leq i \leq p$), 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 $u = \{\langle c_1^u, w_1^u \rangle, \dots, \langle c_q^u, w_q^u \rangle\}$, 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 nearly always computed in a content-based, item-ranking fashion. We distinguish non-neural, path-based frameworks in terms of three aspects:

- **Profile representation.** Different kinds of semantic information can be included in the users' and items' representations.
- **Weighting scheme.** The concepts comprising the user and news profiles are weighted using different strategies to measure their importance.
- **Similarity metric.** The similarity between two news articles is computed using several distance measures.

6.2.2. Representative Models

In the following, we investigate six representative recommendation techniques for this category.

ePaper [67, 101] 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 [67]. 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. [67] 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. (29), 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 [101].

$$IS = \frac{\sum_{i=1}^p N_i \cdot S_i}{\sum_{j=1}^q N_j} \quad (29)$$

A different approach is adopted in **Magellan** [68], 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 [102] 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 $w_{\text{centrality}} = \frac{1}{\text{hops}} \times \frac{w_r}{\text{hops}}$.

The centrality score ensures that concepts with shorter and stronger connections to the top-ranked concept will be assigned 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 that 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. (30)), weighted by a constant value α assigned to the top-ranked concept:

$$w_{\text{importance}} = \alpha \times \frac{w_{\text{centrality}}}{\text{rank}} \quad (30)$$

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, $w = w_{\text{importance}} \times n_i$. According to this weighting scheme, Magellan will score higher stories which frequently contain entities with 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.

Similar to SF-IDF, the Semantic Similarity (SS) recommendation model [58] represents a news item using the WordNet synsets of the terms it contains, as shown in Eqs. (10) and (11). 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:

$$V = (\langle s_1^n, s_1^u \rangle, \dots, \langle s_p^n, s_q^u \rangle) \forall s^n \in v, s^u \in u \quad (31)$$

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

$$W \subseteq V \forall (s^n, s^u) \in W : \text{POS}(s^n) = \text{POS}(s^u) \quad (32)$$

where $\text{POS}(s^n)$ and $\text{POS}(s^u)$ 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 $\text{sim}(s^n, s^u)$ relative to the total number of combinations $|W|$, illustrated as follows:

$$\text{sim}_{SS} = \frac{\sum_{(s^n, s^u) \in W} \text{sim}(s^n, s^u)}{|W|} \quad (33)$$

The WordNet taxonomy constitutes a hierarchy of "is-a" relationships between its nodes which, in turn, constitute synsets. As such, Capelle et al. [58] propose five semantic similarity measures to calculate the similarity rank $\text{sim}(s^n, s^u)$ for each combination of synsets

in W , namely the extent to which two synsets are semantically close. Three of the measures (Jiang and Conrath [103] $\text{sim}_{J\&C}$, Resnik [104] sim_R , Lin [105] sim_L) utilize the information content of a node, defined as $\text{IC}(s) = -\log \sum_{w \in S} p(w)$. 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 [104]. The three metrics are illustrated in Eqs. (34)-(36).

$$\text{sim}_{J\&C}(s^n, s^u) = \frac{1}{\text{dist}_{J\&C}(s^n, s^u)} \quad (34)$$

$$= \frac{1}{\text{IC}(s^n) + \text{IC}(s^u) - 2 \times \text{LCS}(s^n, s^u)} \quad (35)$$

$$\text{sim}_R(s^n, s^u) = \text{IC}(\text{LCS}(s^n, s^u)) \quad (36)$$

$$\text{sim}_L(s^n, s^u) = \frac{2 \times \log p(\text{LCS}(s^n, s^u))}{\log p(s^n) + \log p(s^u)}$$

The two remaining metrics, of Leacock and Chodorow [106] $\text{sim}_{L\&C}$, and of Wu and Palmer [107] $\text{sim}_{W\&P}$, shown in Eqs. (37)-(38), 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.

$$\text{sim}_{L\&C}(s^n, s^u) = -\log \frac{\text{length}(s^n, s^u)}{2D} \quad (37)$$

$$\text{sim}_{W\&P}(s^n, s^u) = \frac{2 \times \text{depth}(\text{LCS}(s^n, s^u))}{\text{length}(s^n, s^u) + 2 \times \text{depth}(\text{LCS}(s^n, s^u))} \quad (38)$$

Similar to Bing-SF-IDF+, **BingSS** [69] extends the semantic lexicon-driven SS recommendation model by taking into account named entities. The semantic similarity formula from Eq. (33) is modified to take into account only the set of synset pairs $\text{TOP}_W^{\beta_{SS}}$ with the highest similarity in W , as follows:

$$\text{sim}_{SS} = \frac{\sum_{(s^n, s^u) \in W} \text{sim}_{SS}(s^n, s^u) \in \text{TOP}_W^{\beta_{SS}}}{|\text{TOP}_W^{\beta_{SS}}|} \quad (39)$$

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. (20) is modified accordingly to take into account this assumption, as illustrated in Eq. (40).

$$\text{sim}_{\text{Bing}} = \frac{\sum_{(e^n, e^u) \in V} \text{sim}_{\text{PMI}}(e^n, e^u) \in \text{TOP}_V^{\beta_{\text{Bing}}}}{|\text{TOP}_V^{\beta_{\text{Bing}}}|} \quad (40)$$

where $\text{TOP}_V^{\beta_{\text{Bing}}}$ represents the set of top- β_{Bing} entity pairs with the highest similarity in V (see Eq. 18), 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 :

$$\text{sim}_{\text{BingSS}} = \alpha \times \text{sim}_{\text{Bing}} + (1 - \alpha) \times \text{sim}_{\text{SS}} \quad (41)$$

OBSM, the ontology-based similarity model proposed by Rao et al. [70], uses 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.

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

The similarity between the profile of a target news article and user is computed in the following way [70]:

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

where

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

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 to 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 models, **SED**, the entity shortest distance over knowledge graphs algorithm proposed by Joseph and Jiang [74], defines item-item similarity as the shortest distance between the subgraphs consisting of named entities extracted from news articles. The approach is threefold [74]. 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 neighborhood 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).

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

Lastly, the similarity between the two articles is computed as the pair-wise shortest distance over the union of their subgraphs[74], as shown in Eq. (46).

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

This method provides a symmetric average minimum row-wise distance which places 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 [74]. For the weighted SED algorithm, different weighting schemes could be used, including the relation weighting scheme, which assigns edge weights based on the number of shared neighbors of two entity nodes from an article.

6.2.3. Summary

Non-neural, path-based knowledge-aware news recommender systems are summarized from the following perspectives:

- **Profile representation.** Various types of representations are used to construct the profiles of users and news articles. On the one hand, ePaper, Magellan, and OBSM use ontological concepts, SS represents items and users in terms of WordNet synsets, while BingSS additionally considers named entities. On the other hand, SED represents profiles in terms of subgraphs consisting of named entities extracted from news articles.
- **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 uses 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. 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. 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.

- **Similarity metric.** The majority of models previously discussed use 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. The SS-based models use different functions that either take into account the information content of a node in the knowledge base, or the path lengths between nodes. 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.

6.3. Neural Network-based Methods

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 latent representations of news articles, generated using neural networks, with external information contained in knowledge graphs, encoded by means of knowledge graph embeddings, defined below.

Definition 4. Given a dimensionality $k \ll n$, 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, 47].

6.3.1. 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 neighbors 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 use both the structural and semantic information encoded in knowledge graphs to represent news. Figure 3 exemplifies this process.

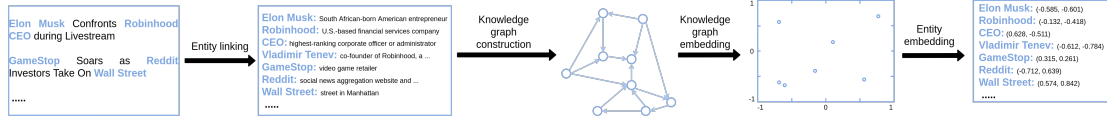


Fig. 3. Illustration of the knowledge distillation process used by neural-based recommendation models (reproduced from [13]).

In contrast to models from the previous categories, neural-based recommenders predict the probability that a user will click on a target article, namely the click-through rate. We consider several factors underlying these recommendation models:

- **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 used to compute node embeddings for the knowledge graph entities [108, 109].
- **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-through probability for a candidate article.

6.3.2. Representative Models

The architectures of 10 neural-based news recommendation frameworks are discussed in this section.

The Collaborative Entity Topic Ranking (CETR) [75] model combines matrix factorization, topic modeling, 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 modeling the user’s reading behavior, the second performing entity-level topic analysis of news, and the last computing representations of the knowledge graph entities.

The user behavior component takes as input the user-news interaction matrix \mathbf{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 factorized into a matrix \mathbf{U} of user features and a matrix \mathbf{V} of news latent features. The factorization method, a Bayesian Personalized Ranking (BPR) model [110], 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 [75]. The third module learns knowledge graph embeddings with the TransR model [111]. 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. [13], 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, $t = [w_1, w_2, \dots, w_N]$, and every word w might correspond to an entity e in a knowledge graph [13]. The enrichment of textual information with external knowledge follows the knowledge distillation process from Figure 3. Wang et al. [13] use not only direct knowledge graph correspondents of identified named entities to construct the subgraph, but also their one-hop neighbors 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 $\text{context}(e)$, 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 [13]:

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

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 neighbors 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 [13]:

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

The first level in DKN's architecture is represented by a knowledge-aware convolutional neural network (KCNN), namely the CNN framework proposed by Kim [112] 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 [113], are projected from the entity to the word vector space, according to Eqs.(50) and (51), using a hyperbolic tangent transformation function g .

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

$$g(\bar{\mathbf{e}}_{1:n}) = [g(\bar{\mathbf{e}}_1)g(\bar{\mathbf{e}}_2)...g(\bar{\mathbf{e}}_n)] \quad (51)$$

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 $\mathbf{W} = [[\mathbf{w}_1g(\mathbf{e}_1)g(\bar{\mathbf{e}}_1)]...[\mathbf{w}_ng(\mathbf{e}_n)g(\bar{\mathbf{e}}_n)]] \in \mathbb{R}^{d \times n \times 3}$.

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 [13]. 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):

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

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:

$$\mathbf{e}(i) = \sum_{k=1}^{N_i} a_{t_k, t_j} \mathbf{e}(t_k) \quad (53)$$

Lastly, DKN [13] 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.** [77] 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 [77]. 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 im-

part of each word in the news representation is computed as follows:

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

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 $\mathbf{w}'_{1:n} = \sum_{i=1}^n a_i' \mathbf{w}_i$, 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.

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

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

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

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:

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

Fourthly, the vector representation of the user and the target news article are combined using a multi-head attention module [114] 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** [78], an 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:

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

where $k = 1, 2, \dots, H$ 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} :

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

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

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 that are too distant from the user's initial interests might introduce noise in the recommendations. This behavior is exemplified in Figure 4 by the fading color of the concentric circles denoting ripple sets. The closer a neighboring 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 [13].

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 analyzing 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 $\mathbf{h}_i \in \mathbb{R}^d$:

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

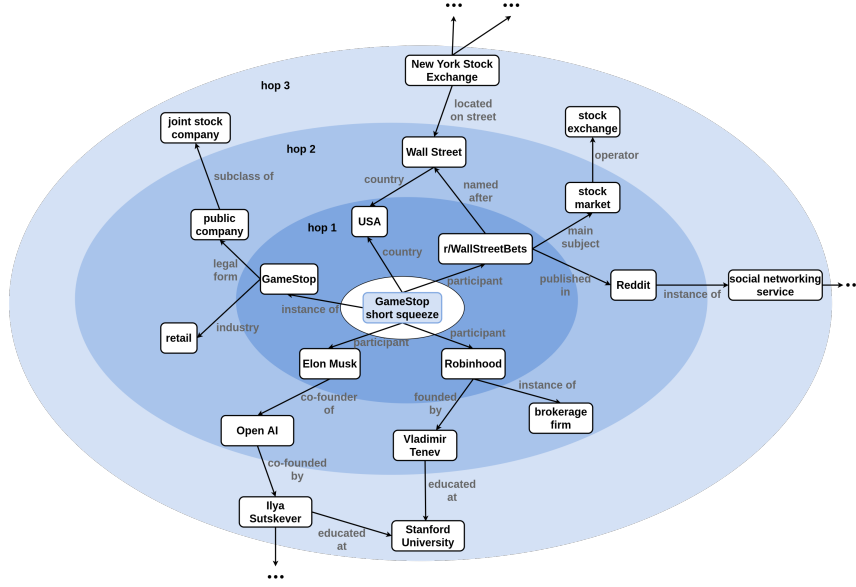


Fig. 4. Illustration of ripple sets of *GameStop short squeeze* in Wikidata. The concentric circles indicate ripple sets with different hops. The fading blue signifies decreasing relatedness between the center and the neighboring entities (reproduced from [78]).

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:

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

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 with 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 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. [79] to extract high-order structural proximity infor-

mation 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 $\pi_{r_i}^u = g(\mathbf{u}, \mathbf{r}_i)$ 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 $\mathbf{v}_{\mathcal{S}(v)}^u = \sum_{e \in \mathcal{S}(v)} \tilde{\pi}_{r_{v,e}}^u \mathbf{e}$, where $\tilde{\pi}_{r_{v,e}}^u \mathbf{e}$ represents the normalized user-relation score over all neighboring entities in v 's ripple set.

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

$$\text{agg} = \sigma(\mathbf{W} \cdot (\mathbf{v} + \mathbf{v}_{\mathcal{S}(v)}^u) + \mathbf{b}) \quad (62)$$

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

In contrast to previous models, the Multi-task feature learning approach for Knowledge graph Recommendation (MKR) [80] 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 the generalization of predictions by using a multi-task learning environment.

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

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

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

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

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 non-linear 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{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 [80]. Lastly, the score of the triple 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:

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

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:

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

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

Although such units are able to extract high-order interactions between items and entities from the two distinct tasks, Wang et al. [80] only employ them in the model's lower layers for two main reasons. On the one hand, the 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) [81] aims to improve previous neural-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 modeling 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 [115], are not projected into the word vector space before stacking. However, as observed by Wang et al. [13], 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 differ.

Higher-order latent collaborative information from the user-item interactions is extracted using embedding propagation layers that integrate the message passing mechanism of GNNs [81] 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 articles, 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:

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

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 neighborhoods 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 neighborhood 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 neighborhood 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**) [82] 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 modeling, 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, $S_t = (S_{t-L}, S_{t-L+1}, \dots, S_{t-1})$, 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 $\mathbf{E} \in \mathbb{R}^{L \times d}$.

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

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 $\mathbf{Q}_i = \sum_w tf - idf_{w,j} \cdot \mathbf{w}_j$. On the other hand, it can be derived by stacking the embeddings of entities extracted from the text, namely the set $\text{entity}(i)$, as $\mathbf{Q}_i = \frac{1}{|\text{entity}(i)|} \sum_{e_i \in \text{entity}(i)} \mathbf{e}_i$.

These representations are then fed into a multi-head self-attention module [114], to obtain the intermediate vector $\mathbf{M} = \text{MultiheadAtt}_{\text{encoder}}^s(E, E, E)$. 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:

$$\mathbf{C}^s = \text{FFM}_{\text{encoder}}^s(\mathbf{M}) = \text{ReLU}(\mathbf{M}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (70)$$

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 recommendation score using Eq. (71):

$$g^s = \text{FFM}_{\text{decoder}}^s(\text{MultiheadAtt}_{\text{decoder}}^s(\mathbf{Q}_{cdt}, \mathbf{C}^s, \mathbf{C}^s)) \quad (71)$$

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 neighboring entities. The contextual entities are embedded using word embeddings pre-trained with a directional skip-gram model [116]. 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} :

$$g^k = \text{FFM}_{\text{decoder}}^k(\text{MultiheadAtt}_{\text{decoder}}^k(\mathbf{Q}_{cdt}, \mathbf{C}^k, \mathbf{C}^k)) \quad (72)$$

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. [83] propose a Knowledge-aware Representation Enhancement model for news Documents (**KRED**) - a new method for creating knowledge-enhanced representations of news for multiple down-

stream 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 [115]. To take into account the contextual information of an entity, the authors employ the approach of Knowledge Graph Attention Network (KGAT) [117] to compute the representation of an entity h using the TransE embeddings of itself \mathbf{e}_h and its 1-hop neighbors, as follows:

$$\mathbf{e}_{N(h)} = \text{ReLU}(\mathbf{W}_0(\mathbf{e}_h \oplus \sum_{(h,r,t) \in N(h)} \pi(h,r,t)\mathbf{e}_t)) \quad (73)$$

In Eq. (73), $\pi(h,r,t)$ represents the softmax normalized attention weights that adjust the amount of information propagated from a neighbor node to a given entity. The unnormalized attention coefficients $\pi_0(h,r,t)$ are determined using a two-layer fully connected neural network:

$$\pi_0(h,r,t) = \mathbf{w}_2 \text{ReLU}(\mathbf{W}_1(\mathbf{e}_h \oplus \mathbf{e}_r \oplus \mathbf{e}_t) + \mathbf{b}_1) + b_2 \quad (74)$$

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 $\mathbf{C}_{p_h}^{(1)}$, $p_h \in \{1, 2\}$. 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 $\mathbf{C}_{f_i}^{(2)}$. Lastly, a category encoding vector $\mathbf{C}_{t_i}^{(3)}$ indicates the entity type t_i . The embedding of entity h is thus enhanced in the following way:

$$\mathbf{e}_{\mathcal{I}_h} = \mathbf{e}_{N(h)} + \mathbf{C}_{p_h}^{(1)} + \mathbf{C}_{f_i}^{(2)} + \mathbf{C}_{t_i}^{(3)} \quad (75)$$

The entities' representations are aggregated into a single vector in the information distillation layer, by means of an attention mechanism that 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. (76), 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. (77).

$$\pi_0(h, v) = \mathbf{w}_2 \text{ReLU}(\mathbf{W}_1(\mathbf{e}_{\mathcal{I}_h} \oplus \mathbf{v}_d + \mathbf{b}_1) + b_2) \quad (76)$$

$$\mathbf{e}_{\mathcal{O}_h} = \sum_{h \in \mathcal{N}(v)} \pi(h, v) \mathbf{e}_{\mathcal{I}_h} \quad (77)$$

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 into 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 [83].

In addition to injecting external knowledge into the recommendation model, the Topic-Enriched Knowledge Graph Recommendation System (TEKGR) [84] improves items' representation by exploiting the topical relations among the 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 articles are represented by their titles.

Firstly, the KG-based news modeling 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 bi-directional 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 [84]. This is followed by an attention layer which extracts more informative features from the vector representations by giving higher importance to more relevant words. Hence, the final representation of news article $\mathbf{e}(t_k)$ is given by the weighted sum of the contextual word representations, where the weights are attention coefficients.

The knowledge encoder extracts topic information from the news titles through three layers [84]. The con-

cept extraction layer links each news title with corresponding concepts in a knowledge graph using an "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 to 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 sub-graph is enriched with 2-hop neighbors of the extracted entities, as well as with topical information distilled by the knowledge encoder [84]. Therefore, not only are knowledge entities from the text disambiguated and their contextual information is 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 are computed with a graph neural network [118]. 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.

6.3.3. Summary

Neural-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 recom-

mendations using the matrix of implicit feedback and a knowledge graph. Furthermore, the majority of models use 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 neighbors, 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 [115], TransH [119], TransR [111], or TransD [113] 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 fuses KCNN for content-based representation of news with a message-passing GNN that captures collaborative signals among the news. All the remaining models use a type of attention mechanism. For example, DKN combines KCNN used for news representation with a DNN-based attention layer. Gao et al.'s model [77] 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 neighbors, followed by the attention mechanism of the Transformer [114] used for assigning different weights for each entity and for comput-

ing 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.

7. 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.

7.1. Evaluation Methodologies

The type of evaluation methodology depends on the target function of the recommendation models and the user 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 target function 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 that target the click-through rate perform experiments in an offline setting, using data comprising of logs representing users' historical interactions with sets of news.

Table 6

Overview of evaluation settings. We list the model's category and abbreviated name, datasets used, and reported evaluation metrics and setup information. 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
NNECM	Semantic aware context recommendation [52, 53]	News@hand	P@K	PROCS, DS
	Social tags enriched recommendations [54]	News@hand	Relevance	PROCS, DS
	Semantic relatedness [55]	Unknown source	P, R, F1	-
	RSR [46]	Hermes News Portal	Acc, P, R, Spec	-
	CF-IDF [56]	Hermes News Portal	ROC, PR curves, Kappa	DS
	Hybrid context-aware recommendation [53]	News@hand	P@K	PROCS, DS
	RSR 2 [57]	Hermes News Portal	Acc, P, R, Spec	DS
	SF-IDF [58]	Hermes News Portal	Acc, P, R, F1, Spec, t -test	DS
	SF-IDF+ [59]	Hermes News Portal	Acc, P, R, F1, Spec, t -test	DS, PARAMS
	Bing-SF-IDF [60]	Hermes News Portal	Acc, P, R, F1, Spec, Kappa	DS, PARAMS
	Bing-SF-IDF+ [61]	Hermes News Portal	Acc, P, R, F1, Spec, Kappa	DS, PARAMS
	Agarwal and Singhal [62]	BBC, CNBC, Times of India	P, R, F1	PROCS, PARAMS
	OF-IDF [63]	Hermes News Portal	P, R, F1, runtime	DS
	CF-IDF+ [64]	Hermes News Portal	P, R, F1, ROC, AUC, Kappa	PROCS, DS, PARAMS
NNPM	Bing-CF-IDF+ [65]	Hermes News Portal	P, R, F1, ROC, Kappa	PROCS, DS, PARAMS
	Bing-CSF-IDF+ [66]	Hermes News Portal	F1, Kappa, Student's t -test	PROCS, DS
	ePaper [67]	The Jerusalem Post	NDPM, MAE	PROCS, DS, PARAMS
	Magellan [68]	Unknown source	Acc, R	-
	SS [58]	Hermes News Portal	Acc, P, R, F1, Spec, t -test	DS
	BingSS [69]	Hermes News Portal	Acc, P, R, F1, Spec	DS, PARAMS
	OBSM [70]	New York Times, Sina News	P, R, F1	DS
	Kumar and Kulkarni [71]	CNN news	R	PROCS, DS, PARAMS
	Werner and Cruz [72]	Unknown source	P, R, F1	PROCS, PARAMS
	BKSport [73]	Sky Sports, ESPN, Yahoo Sports	P	PROCS, DS, PARAMS
NM	SED [74]	CNREC	P, R, F1	PROCS, DS, PARAMS
	CETR [75]	Hupu News	R@K	PROCS, DS, PARAMS
	Colombo-Mendoza et al. [76]	Unknown source	P, R, F1	PROCS
	DKN [13]	Bing News	F1, AUC	All available
	Gao et al. [77]	Unknown source	AUC, NDCG@K	DS, PARAMS
	RippleNet [78]	Bing News	P@K, R@K, F1@K, AUC, Acc	All available
	RippleNet-agg [79]	Bing News	P@K, R@K, F1@K, AUC, Acc	All available
	MKR [80]	Bing News	Acc, AUC, P@K, R@K	All available
	IGNN [81]	DC, Adressa	R@K, NDCG@K	PROCS, DS, PARAMS
	Saskr [82]	Eastday Toutiao	HR@10, MRR	PROCS, DS
	KRED [83]	Microsoft News	AUC, NDCG@K, HR@K, ACC, F1-macro	All available
	TEKGR [84]	Bing News, Adressa	F1, AUC	PROCS, DS, PARAMS
	KCNR [85]	Sogou News	R@K, NDCG@K	PROCS, DS, PARAMS
	KG-RWSNM [86]	Weibo	AUC, P, R, F1	PROCS, PARAMS
	MUKG [87]	Bing News	AUC, Acc, P@K, R@K	PROCS, DS, PARAMS

Table 6 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, neural-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, non-neural, entity-centric methods use statistical hypothesis tests, such as the Student's t -test, to measure the significance of the experimental results.

7.2. Evaluation Datasets

In comparison to the relatively uniform usage of evaluation metrics, the type of datasets used for evaluation varies significantly among recommender systems. The majority of non-neural models can be clustered into two groups, depending on the dataset used. As shown in Table 6, semantic-aware context recommenders use the News@hand architecture described in [120]. Most of the remaining models are incorporated in the Hermes News Portal [121]. A few of the path-based, non-neural recommender systems construct their own datasets using news articles collected from websites such as the New York Times, or Sina News¹⁶, CNN¹⁷, Sky Sports¹⁸, ESPN¹⁹, or Yahoo Sports²⁰. Joseph and Jiang [74] developed CNREC²¹ 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 neural-based frameworks consist of user interaction logs gathered from websites such as Bing News²², Microsoft News²³, Hupu²⁴, Eastday Toutiao²⁵, Sogou News²⁶, or the Weibo dataset [122]. An exception is constituted by IGNN and TEKGR, evaluated on the Adressa dataset. Adressa [123] is an event-based dataset comprising of click log data collected from a Norwegian news portal. Although the Adressa dataset is often used in evaluating deep learning-based news recommender systems [124–127], it is not used by any other of the surveyed knowledge-aware models.

As it can be further observed in Table 7, which summarizes the statistics of the used evaluation datasets, the number of users and items contained in these datasets varies widely. Non-neural models 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, neural-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 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 more than half of the cases.

7.3. Reproducibility and Comparability of Experiments

Table 6 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 neural-based models, 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 6, only 5 out of the 39 surveyed papers provide all this information. For both sub-categories of non-neural models, generally only the data split and some of the parameters or processing steps 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 neural-based framework offer extensive details regarding their evaluation settings and model architecture. This phenomenon could be explained in two ways. On the one hand, since all neural-based models are deep learning architectures, hyperparameters play a central role in their performance. On the other hand, most of the approaches in this family have been published in the recent past, and there has been a trend in the recent years in the academic community to make implementation details available when publishing a research paper in order to facility reproducibility.

¹⁶<http://news.sina.com.cn>

¹⁷<http://edition.cnn.com/services/rss>

¹⁸<http://www.skysports.com>

¹⁹<http://www.espnfcasia.com>

²⁰<http://sports.yahoo.com/>

²¹<https://github.com/kevinj22/CNRec/blob/master/CNRec.zip>

²²<https://www.bing.com/news>

²³<https://news.microsoft.com>

²⁴<https://www.hupu.com>

²⁵<http://mini.eastday.com/>

²⁶<https://www.news.sogou.com>

Table 7

Overview of evaluation datasets. We list the model's category and abbreviated name, the dataset's source, language, time frame, as well as the number of users, items, and interactions. 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
NNECM	Semantic aware context recommendation [52, 53]	BBC, CNN, New York Times, Washington Post	English	01/01/2008-01/03/2008	16	9,698	N/A
	Social tags enriched recommendations [54]	BBC, CNN, New York Times, Washington Post	English	01/01/2008-01/03/2008	20	9,698	N/A
	Semantic relatedness [55]	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 [46]	Unknown	Unknown	Unknown	5	Unknown	1500 *
	CF-IDF [56]	Unknown	Unknown	Unknown	19	100	1900 *
	Hybrid context-aware recommendation [53]	BBC, CNN, New York Times, Washington Post	English	01/01/2008-01/03/2008	20	9,698	N/A
	RSR 2 [57]	Unknown	Unknown	Unknown	5	300	1500 *
	SF-IDF [58]	Unknown	Unknown	Unknown	19	100	1900
	SF-IDF+ [59]	Reuters	English	Unknown	N/A	100	N/A
NNPM	Bing-SF-IDF [60]	Reuters	English	N/A	100	N/A	N/A
	Bing-SF-IDF+ [61]	Reuters	English	Unknown	Unknown	Unknown	Unknown
	Agarwal and Singhal [62]	BBC, CNBC, Times of India	Unknown	2010	33	1823	3600
	OF-IDF [63]	Unknown	Unknown	Unknown	N/A	100	N/A
	CF-IDF+ [64]	Reuters	English	Unknown	N/A	100	N/A
	Bing-CF-IDF+ [65]	Reuters	English	Unknown	N/A	100	N/A
	Bing-CSF-IDF+ [66]	Reuters	English	Unknown	N/A	100	N/A
	ePaper [67]	The Jerusalem Post	English	4 days	57	Unknown	4,731 *
	Magellan [68]	Unknown	Unknown	09/2010-01/2011	N/A	Unknown	N/A
	SS [58]	Unknown	Unknown	Unknown	19	100	1900
KGEM	BingSS [69]	Reuters	English	Unknown	N/A	100	N/A
	OBSM [70]	New York Times	English	2006-2007	581	6,000	232,400 *
	Kumar and Kulkarni [71]	Sina News	Chinese	Unknown	581	Unknown	232,400 *
	Werner and Cruz [72]	CNN	English	Unknown	Unknown	Unknown	Unknown
	BKSport [73]	Unknown	Unknown	Unknown	10	70	700 *
	SED [74]	Sky Sports, ESPN, Yahoo Sports	English	Unknown	Unknown	100	Unknown
	CETR [75]	Unknown	English	25/08/2014-28/08/2014	N/A	300	N/A
	Colombo-Mendoza et al. [76]	Hupu News	Chinese	Unknown	3,118	9,684	132,713
	DKN [13]	Bolsa de Madrid	Spanish	Unknown	10	20	200 *
	Gao et al. [77]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
KGEM	RippleNet [78]	Unknown	Unknown	19/03/2018-31/03/2018	26,224	13,285	1,498,862
	RippleNet-agg [79]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
	MKR [80]	Bing News	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
	IGNN [81]	DC	Unknown	16/10/2016-11/08/2017	141,487	535,145	1,025,192
	Saskr [82]	Adressa	Norwegian	Unknown	10,000	6,385	116,225
	KRED [83]	Eastday Toutiao	Chinese	2017	640,503	20,428	3,101,991
	TEKGR [84]	Microsoft News	Unknown	15/01/2019-28/01/2019	665,034	108,684	861,996
	KCNR [85]	Bing News	Unknown	16/10/2016-11/08/2017	24,542	24,542	1,590,092
	KG-RWSNM [86]	Adressa	Norwegian	2017	141,487	535,145	1,025,192
	MUKG [87]	Sogou News	Chinese	19/03/2019-26/03/2019	561,733	11,207	2,286,835

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

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. Nonetheless, all the previously discussed knowledge-aware news recommender systems use a form of named entity recognition and linking in order to identify entities and concepts in the news articles and to then map them to a knowledge base. However, almost none of the papers explicitly mention how these steps are performed and implemented. Other important steps required by any news recommender system, such as general text preprocessing, which can heavily influence the data representation, and ultimately, the generated recommendations, are also not discussed. In addition to the recommendation module itself, such steps constitute important dimensions that may differ between systems, and in turn, choices in their design and implementation may lead to great differences in performance. 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 8, only 6 out of 16 papers describing non-neural, entity-centric systems 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 non-neural, path-based recommenders, only ePaper [67], BingSS [69], Kumar and Kulkarni [71] and SED [74] evaluate the influence of different parameters or user profile initialization on the model's performance. In

comparison, the evaluation of neural-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 analyze variants of the model obtained by removing either the contextualization of user preferences, the extension of user and news profiles, or both. Werner and Cruz [72] investigate different similarity metrics and recommendation techniques, while BKSPORT examines the performance of the recommender when taking into account only semantic similarities, only content similarities, or both. Colombo-Mendoza et al. [76] similarly experiment with two network-based feature learning algorithms and different similarity metrics. DKN removes 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 analyzing 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. KCNR analyses the model's performance with and without the user preference prediction module, and the influence of the preference propagation in the knowledge graph by considering different k -hop neighbor information. KG-RWSNM performs a ablation study on the impact of the social network information, while MUKG examines the recommender's performance given varying degrees of data sparsity.

Table 8

Overview of evaluated features and components. We report the model's category and abbreviated name, the features evaluated and the components evaluated during an ablation study, if one was conducted. The abbreviations used in the table are the following: dim. = dimension, emb. = embedding, init. = initialization, # = number of

Category	Model	Eval. feats.	Ablation study
NNECM	Semantic aware context recommendation [52, 53]	-	model components
	Social tags enriched recommendations [54]	-	-
	Semantic relatedness [55]	-	-
	RSR [46]	threshold value	-
	CF-IDF [56]	threshold value	-
	Hybrid context-aware recommendation [53]	-	model components
	RSR 2 [57]	-	-
	SF-IDF [58]	-	-
	SF-IDF+ [59]	-	-
	Bing-SF-IDF [60]	threshold value	-
	Bing-SF-IDF+ [61]	threshold value	-
	Agarwal and Singhal [62]	-	-
	OF-IDF [63]	-	-
	CF-IDF+ [64]	-	-
	Bing-CF-IDF+ [65]	threshold value, Bing similarity parameters	-
	Bing-CSF-IDF+ [66]	threshold value, Bing similarity parameters	-
NNPM	ePaper [67, 101]	matching parameters, #user ratings, user profile init., concept weights	-
	Magellan [68]	-	-
	SS [58]	-	-
	BingSS [69]	threshold value, Bing similarity parameters	-
	OBSM [70]	-	-
	Kumar and Kulkarni [71]	vector dim.	-
	Werner and Cruz [72]	-	similarity metric, recommendation algorithm
	BKSport [73]	-	semantic similarity, content similarity
	SED [74]	length expansion radius, entity screening, context words, edge weighting schema, distance measure, disconnected nodes penalty	-
NM	CETR [75]	-	-
	Colombo-Mendoza et al. [76]	-	network-based feature learning algorithm, similarity metric
	DKN [13]	word & entity emb. dim., #filters, window size	knowledge & attention component, KGE model, transformation function
	Gao et al. [77]	user profile length, #keywords	-
	RippleNet [78]	ripple set size, #hops, emb. dim. regularization weight	-
	RippleNet-agg [79]	aggregator, ripple set depth, neighbourhood sampling size, emb. dim.	-
	MKR [80]	KG size, RS training frequency, emb. dim.	cross&compress units, multi-task learning
	IGNN [81]	emb. dim., #emb. propagation layers	emb. propagation layers
	Saskr [82]	emb. layer init., article emb. strategy, sequence length, #targets, weight factor	-
	KRED [83]	base document vector, training strategy	layers (incl. knowledge component)
	TEKGR [84]	#hops	encoder types
	KCNR [85]	user's click history length	preference prediction module, k-hop neighbour information
	KG-RWSNM [86]	-	social network component, entity information
	MUKG [87]	-	data sparsity, recommender system module

7.4. Summary

Overall, this investigation of evaluation approaches shows that there is no unified evaluation methodology to produce comparable experiments. Moreover, the datasets used 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, we argue that only some of the most recent, neural-based approaches could be replicated given the avail-

able data, whereas the remaining methods cannot be reproduced in accordance to the original implementations. In this context, knowledge-aware news recommendation models lack reproducibility and comparability, standards which have been strongly encouraged in other fields of machine learning.

8. Open Issues and Future Directions

Existing works have already established a strong foundation for knowledge-aware news recommender systems. In this section, we firstly discuss which fundamental challenges of news recommendation have already been addressed by knowledge-aware models, then identify and elaborate on several open issues in the field, and propose a number of promising research directions.

In addition to general problems, such as the cold-start, data sparsity, personalization, diversity, or privacy issues [128], news recommenders systems face additional domain-specific challenges, as explained in Section 2. Data sparsity and cold-start problems, or taking into account named entities and semantic relations between the news have clearly been addressed by the injection of external structured information from a knowledge base into the recommendation module, as it can be observed from Section 5.2.5 and Table 5. Similarly, the news consumption behavior and the lack of explicit feedback have also been tackled by enriching initial user profile information gathered from click logs of read news articles with related information about entities or concepts identified in the respective articles and contained in a knowledge base. Although recommendation diversity has been explicitly addressed only in one of the discussed models, using knowledge graphs to extract knowledge-level connections between news can improve diversity compared to solely text-based recommenders by taking into account relations between news that stem from second or third-order related neighbors of extracted entities.

Several challenges have only been partly addressed by the surveyed works. For example, recommenders that construct user and item profiles based on ontological concepts take into account a smaller amount of data than those that use the full-text of news articles. Therefore, computations are faster and the negative effect of the large volume of data characterizing news recommendations is diminished. However, as will be explained in Section 8.2, it remains unclear what is the impact of utilizing complete, large-scale knowl-

edge graphs as side information. Similarly, the issue of sequential news consumption has only been addressed by Saskr, although it constitutes an important characteristic of news recommendation in general which has already been addressed by non-knowledge-aware recommenders [124, 129, 130].

Nevertheless, a large number of news-specific challenges are not yet tackled by knowledge-aware news recommender systems. Issues related to the fairness of recommendation, such as over-specialization and bias or fake news, which constitute central challenges and avenues for research nowadays are not addressed in any of the surveyed papers. Similarly, neural knowledge-aware recommenders focus on the short-term interests of users and do not take into consideration their long-term preferences. Lastly, the influence of using knowledge bases, which often might not be frequently updated, on the timely recommendation of news with short shelf lives has so far not been investigated.

In the remainder of this section we elaborate on the identified open issues and propose future research directions.

8.1. Comparability of Evaluations

Zhang et al. [131] have observed that the entire field of recommender systems 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. A similarly troubling finding with regards to the reproducibility of research published in the area of recommender systems has been discussed by Dacrema et al. [132]. The authors compared numerous works published in recent years at prestigious conferences in the domain of neural, collaborative filtering-based recommendation approaches and found that less than a half could be reproduced. Moreover, the majority of the proposed methods were equally good or even outperformed by simpler methods, due to methodological issues such as the choice of baselines, propagation of weak baselines, or the poor tuning of these baselines [132].

The findings of Section 7.3 have shown that currently, knowledge-aware news recommender systems also hardly produce comparable experiments. While neural-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 and verified. Another important

observation is that none of the deep learning models have been compared against recommenders from the non-neural approaches. 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, unification of evaluation metrics, usage of comparable resources and hyperparameters of pre-trained models, and ablation studies.

Need for Benchmark Datasets 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. [133] 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 [134]. Lastly, we have shown in Section 7.3 that the surveyed recommender systems cannot be accurately reproduced and faithfully compared to each other, partially due to the wide range of different knowledge graphs and similar resources used, as well as to the lack of information regarding the construction, processing, and usage of such external resources. In this context, a standard benchmark dataset should also contain a range of knowledge resources, such as knowledge graphs and ontologies, in order to truly allow for a full and fair comparison of knowledge-aware news recommender techniques.

Ensure Reproducibility. In addition to evaluating on the same datasets, with the same data splits, it is necessary to establish a stricter criterion for describing the evaluation methodology in order to ensure replicability of experiments. This means that detailed information of all processing steps, from general text pre-processing, to named entity recognition and linking, or the creation of news-specific knowledge graphs, should be provided to ensure that the experimental setup can be accurately reproduced at all steps. Moreover, underlying assumptions should be clearly stated, and all design choices regarding the hyperparameter optimization strategy should be reported. In case benchmark datasets are not used, the datasets on which the experiments are performed should be made publicly available. Lastly, as suggested in [132], the source code of all proposed models, including baselines, should be published using persistent repositories and ensuring that the reproduction steps are easy to follow by other users (e.g. software requirements, scripts for executing all steps of the pipeline and experiments are readily available).

Unification of Evaluation Metrics. 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.

Usage of Comparable Resources and Hyperparameters of Pre-trained Models. When comparing models against each other, authors should use the same external resources in as far as possible given the recommendation framework. For example, the same knowledge graph or ontology should be used to ensure that the knowledge resource itself does not influence the results on the downstream task. Similarly, in case pre-trained models, such as word or knowledge graph embeddings, are used, the same parameters (e.g. dimension of embedding vector) should be used by all analyzed models.

Ablation Studies. 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 evalua-

tion. 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 [135].

As discussed above, while natural language processing methods, in particular entity recognition and linking, play a crucial role in processing new texts, their effect is rarely documented. In the realm of ablation studies, we would encourage to investigate those effects more thoroughly (e.g. by exploiting different entity linking methods), and to also investigate interaction effects between the natural language processing methods, recommendation methods, and the knowledge resource used.

Overall, all these steps would ensure that models are not only 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 [136] or knowledge graph embeddings [137].

Lastly, we believe that a comparison between non-neural and neural-based 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.

8.2. Scalability of News Recommenders

The continuously increasing amount of news published daily, as well as the growing number of online news readers and their desire to receive news content in a timely manner [7] constitute a constant challenge for any news recommender system, which requires 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. [138] 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. [10] 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) [139], both adapted for large-scale dataset scalability using the MapReduce framework [140], were employed by Das et al. [10] 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 6.2 and 6.3, such models obtain scalability using subgraphs, constructed by sampling fixed-sized neighborhoods. 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 neighbors of a node when gathering its contextual information. Hence, the sampling strategy used for defining a node's neighborhood 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 neighborhood size on the robustness of the system and quality of the recommendations, as performed in [79], should be conducted for a larger variety of recommenders.

8.3. 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 [141], 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 [141].

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. [142] to increase the accuracy of user modeling 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 [142]. A different approach for balancing the accuracy and explainability of recommendations was adopted by Gao et al. [143], 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 dynamic programming [143].

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 [144–146]. 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. [147]. Their method learns inductive rules from an item-centric knowledge graph, which encodes items associations in the form of multi-hop relational patterns. The induced rules are incor-

porated in the recommendation module to address the cold start problem and provide explainability.

A growing number of recent news recommenders 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. [148] 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 require optimizing the subgraph structure, such that its mutual information with the GNN's prediction is maximized. Given the increasing usage of graph neural networks in news recommender systems, the GNNExplainer could be used 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.

8.4. 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 [149] identified two types of biases inherent in recommender systems, namely observation bias, and population imbalance bias.

Observation bias is determined by feedback loops that prevent the model from learning how to predict items that are dissimilar to the previously recommended or consumed ones [150]. 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 that are not recommended to them, thus reinforcing the recommender's algorithmic behavior [150]. 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 the news that

supports or amplify their opinions [31]. In turn, this might lead, in the long run, to opinion polarization and self-radicalization of individuals through online media [151].

Bias stemming from imbalanced data is a systematic bias caused by societal or historical discrimination, which occurs when different categories of users are represented in unequal proportions in the data used for training a recommender system [149]. 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. [152] 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. [153] 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 neighborhoods combined with a sparse linear model to obtain a desirable trade-off between fairness of results and personalization of recommendations [153].

Wu et al. [154] 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 behaviors, and bias-free ones that capture attribute-independent information related to the user's interests. Adversarial learning is used 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 [154].

Symeonidis et al. [155] propose a popularity-based and a distance-based novelty-aware matrix factorization technique to address the problem of filter bubbles created by recommender systems. Novelty-aware matrix factorization introduces in the classic regularized matrix factorization model a soft constraint that controls how new items are being recommended. In the popularity-based recommendation setting, the novelty

of an item is defined as the inverse of its popularity, with items being more novel the fewer people are aware of them. In the case of distance-based recommendations, an item is considered novel if the topic category to which it belongs does not comprise of many other items with which the user has already interacted in the past [155]. However, this approach focuses on systems based on matrix factorization, which are not used by knowledge-aware news recommender systems, as discussed previously.

In the field of news recommendation, Gharahighi et al. [156] address the news recommendation task from a multi-stakeholder perspective and adopt a hypergraph learning method in order to take into account multiple stakeholders and counteract the negative effect of popularity bias on the recommendations. The stakeholders involved in the news recommendation scenario and their interactions are modeled by means of a hypergraph, thus enabling the direct computation of the relatedness between different stakeholders, represented as vertices of the hypergraph. Moreover, the authors introduce a temporal-aware learning approach which dynamically updates the weights given to the different stakeholders in order to increase recommendation fairness [156].

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. Moreover, none of the surveyed models investigates whether filter bubbles arise when using external knowledge resources for recommendations. Hence, given the importance of these topics, investigating how fairness can be incorporated into knowledge-aware news recommender systems, as well as examining if filter bubbles are created and how their effect can be diminished, represent promising directions for future works in this field.

Another related problem is that of fake news, which can be propagated by recommender systems using news data whose credibility has not been verified. In this context, numerous fake news detection algorithms have already been proposed [157–159]. Additionally, knowledge graphs can also be used to detect whether the news is fabricated [160–162]. Nonetheless, none of the surveyed works is concerned with the potential propagation of fake news by the recommender system or ways to mitigate it. Therefore, we conclude that incorporating a fake news detection module, potentially based on knowledge graphs, represents an important avenue for research that would contribute to re-

ducing the spread of fake news and misinformation by knowledge-aware news recommendation algorithms.

8.5. Multilingual and Multi-Modal News Recommendation

Today, online news comes in various shapes. Next to online newspapers, internet users increasingly consume their news in the form of podcasts or videos, most often using a mix of text, audio, and video modalities [163]. Given that observation, multi-modal recommendation methods are likely to gain more traction but are rarely observed so far [7]. Here, knowledge-based recommenders would be an interesting opportunity, since knowledge-based content representations and multi-modal knowledge graphs [164] could be used to form links between news present in different modalities. Moreover, given the strong trend of neural recommendation methods in the field, multi-modal embedding models [165, 166] could be an interesting pathway towards developing such recommendation techniques [167].

Multi-lingual news consumption is also quite frequent. According to a study from 2014, 36% of all internet users in the European Union “frequently” and even 81% “occasionally” consume news and information online in more than one language [168]. While many knowledge graphs are inherently multi-lingual, and the use of identifiers for concepts and entities can help to bridge the gap between documents in different languages, all of the approaches surveyed in this document are monolingual. Like multilingual neural language models can be applied to the task of cross-lingual news recommendation [169], we also foresee the development of knowledge-based multi-lingual news recommenders.

8.6. Multi-Task Learning for Recommendation

Multi-task learning [170] 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 [171]. This approach has proven successful in numerous applications, ranging from computer vision to speech recognition and natural language processing [172].

Multi-task learning has also been employed by recommender systems from different domains [173]. 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 [174, 175]. Similarly, in the field of knowledge-aware news recommendation, Wang et al. [80] have used this paradigm to jointly train a model for the tasks of news recommendation and knowledge graph embedding, while Liu et al. [83] 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.

8.7. Sequential and Timely Recommendations

Readers consume news in sequences and prefer updates about ongoing and developing stories, rather than repeated or highly similar articles. Taking into account sequential dependencies between articles has been addressed in news recommendation generally by means of recurrent neural networks (RNNs) [129]. More recent approaches combine RNNs with attention modules. For example, Zhu et al. [124] use an attention-based RNN as a sequential information extractor that can automatically model the dynamic history sequential features used to represent a user’s clicked articles, while Bai et al. [130] use a combination of RNNs and attention to build a sequence-aware, user-based collaborative filtering recommender system. However, knowledge-aware sequential news recommender systems have been rarely proposed so far. In this context, we believe that enhancing existing sequential news recommenders with side information from a knowledge base is an interesting research avenue towards tackling this problem.

In addition to consuming articles sequentially, readers prefer recent and up-to-date news. Not only does an article’s relevance diminish over time, but the news is constantly updated and superseded by more popular pieces of information. In turn, this means that recom-

mendations that are based not only on the text content of news, but also on knowledge entities and side information, need to ensure the timeliness of the information contained in external knowledge bases. However, many large knowledge graphs quickly become outdated and do not contain the latest information about world events [176]. In this context, we encourage a shift from static towards temporal knowledge graphs that capture temporal dynamics of entities and the relations [177] between them. This could help knowledge-aware news recommenders overcome the problem that the validity of any facts contained in static knowledge graphs is constrained to a specific time period.

Temporal knowledge graphs have gained traction in the latest years in the field of recommender systems. For example, Xiao et al. [178] proposed a temporal knowledge graph, which is incrementally constructed from user-item interactions and related auxiliary information, and used for recommendation. Similarly, Mezni [179] leverages a temporal knowledge graph to build a time-aware recommender system for service recommendation. Therefore, we believe that using temporal knowledge graphs is not only a potential solution to ensure the timeliness of external data injected in the recommendation module, but also a highly promising direction for future research in the field of knowledge-aware news recommendation.

8.8. Changing User Preferences

In addition to preferring timely news, readers also have preferences that evolve over time. On the one hand, short-term preferences are determined by current trends, popularity, and context of certain news and events, such as the local elections in a country. On the other hand, long-term preferences evolve more slowly and are motivated by socio-economic and personal factors, such as an interest in climate change [23]. However, as it can be observed from Section 6.3, neural knowledge-aware news recommender systems learn single representations of users that do not differentiate between the two types of user interest. In turn, this can be detrimental for the generated recommendations, as users might not only want to see news regarding the latest events, but also read articles related to their long-term interests.

The LSTUR model proposed by An et al. [30] takes into account both kinds of preferences when constructing user profiles. More specifically, long-term representations are given by the embeddings of user IDs, while short-term preferences are captured from the

users' recently browsed news using a gated recurrent network (GRU). The two representations can either be concatenated to obtain the final user representation, or the long-term user representation can be used to initialize the hidden state of the GRU network for the short-term representation module [30].

Another approach has been proposed by Hu et al. [126]. Their model, GNewsRec, uses a GNN on a heterogeneous user-news-topic graph to learn the user's long-term interest encoded as high-order relationships between users, items, and topics in the graph. To capture the short-term user preferences, the authors employ an attention-based long short-term memory [180] on the user's reading history.

Given the existing research already conducted to account for changing user interests in news recommendation, we believe that incorporating techniques that are able to differentiate between short-term and long-term user preferences in knowledge-aware news recommenders is a pathway worth pursuing to ensure diversity of recommendations and user satisfaction.

9. Conclusion

In this survey paper, we have extensively reviewed knowledge-aware news recommender systems. We propose a new taxonomy for classifying existing recommenders, based on how the latent representations are generated for the users' and articles' profiles using concepts and entities from a knowledge base, as well as on the type of similarity metric used. According to the classification scheme, we categorize knowledge-aware news recommender systems into non-neural and neural-based frameworks, with the former category further divided into entity-centric and path-based methods. Representative models from each category are summarized and thoroughly analyzed. 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.

10. Acknowledgement

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] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] J. Borràs, A. Moreno and A. Valls, Intelligent tourism recommender systems: A survey, *Expert Systems with Applications* **41**(16) (2014), 7370–7389.
- [6] J. Beel, B. Gipp, S. Langer and C. Breiteringer, Research-paper recommender systems: a literature survey, *International Journal on Digital Libraries* **17**(4) (2016), 305–338.
- [7] S. Raza and C. Ding, News recommender system: a review of recent progress, challenges, and opportunities, *Artificial Intelligence Review* (2021), 1–52, Publisher: Springer.
- [8] Özgöbek, J.A. Gulla and R.C. Erdur, A Survey on Challenges and Methods in News Recommendation, in: *WEBIST* (2), 2014, pp. 278–285.
- [9] M. Karimi, D. Jannach and M. Jugovac, News recommender systems—Survey and roads ahead, *Information Processing & Management* **54**(6) (2018), 1203–1227.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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*, 2018, pp. 1835–1844.
- [14] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semantic web* **8**(3) (2017), 489–508.
- [15] 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).
- [16] 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).
- [17] 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.
- [18] M. Li and L. Wang, A Survey on Personalized News Recommendation Technology, *IEEE Access* **7** (2019), 145861–145879.
- [19] 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.
- [20] 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) (2011), 754.
- [21] 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.
- [22] 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.
- [23] M. Harandi and J.A. Gulla, Survey of User Profiling in News Recommender Systems, in: *INRA@ RecSys*, 2015, pp. 20–26.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, *Recommender systems: an introduction*, Cambridge University Press, 2010.
- [28] 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.
- [29] R. Burke, Hybrid recommender systems: Survey and experiments, *User modeling and user-adapted interaction* **12**(4) (2002), 331–370.

- [30] 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.
- [31] E. Pariser, *The filter bubble: What the Internet is hiding from you*, Penguin UK, 2011.
- [32] D.M. Lazer, M.A. Baum, Y. Benkler, A.J. Berinsky, K.M. Greenhill, F. Menczer, M.J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild et al., The science of fake news, *Science* **359**(6380) (2018), 1094–1096, Publisher: American Association for the Advancement of Science.
- [33] T. Hassan, Trust and trustworthiness in social recommender systems, in: *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 529–532.
- [34] S. Mohseni, E. Ragan and X. Hu, Open issues in combating fake news: Interpretability as an opportunity, *arXiv preprint arXiv:1904.03016* (2019).
- [35] C. Feng, M. Khan, A.U. Rahman and A. Ahmad, News recommendation systems-accomplishments, challenges & future directions, *IEEE Access* **8** (2020), 16702–16725, Publisher: IEEE.
- [36] 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.
- [37] 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.
- [38] 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.
- [39] D. Vrandečić and M. Krötzsch, Wikidata: a free collaborative knowledgebase, *Communications of the ACM* **57**(10) (2014), 78–85.
- [40] Understand Your World with Bing, 2013, Publication Title: Bing search blog. <https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/>.
- [41] 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.
- [42] L. Ehrlinger and W. Wöb, Towards a Definition of Knowledge Graphs, *SEMANTiCS (Posters, Demos, SuCCESS)* **48** (2016), 1–4.
- [43] S. Wu, W. Zhang, F. Sun and B. Cui, Graph Neural Networks in Recommender Systems: A Survey, *arXiv preprint arXiv:2011.02260* (2020).
- [44] 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).
- [45] 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.
- [46] 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.
- [47] 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.
- [48] Z. Wang, Z. Wang, J. Li and J.Z. Pan, Building a large scale knowledge base from chinese wiki encyclopedia, in: *Joint International Semantic Technology Conference*, Springer, 2011, pp. 80–95.
- [49] R. Troncy, Bringing the IPTC news architecture into the semantic web, in: *International Semantic Web Conference*, Springer, 2008, pp. 483–498.
- [50] Google, *Freebase Data Dumps*, February 1, 2021 edn, 2021. <https://developers.google.com/freebase/data>.
- [51] G.A. Miller, WordNet: a lexical database for English, *Communications of the ACM* **38**(11) (1995), 39–41.
- [52] 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, 2008, pp. 562–565.
- [53] 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.
- [54] I. Cantador, M. Szomszor, H. Alani, M. Fernández and P. Castells, Enriching ontological user profiles with tagging history for multi-domain recommendations (2008).
- [55] 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.
- [56] 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.
- [57] 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.
- [58] 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.
- [59] 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.
- [60] F. Hogenboom, M. Capelle and M. Moerland, News Recommendation Using Semantics with the Bing-SF-IDF Approach, in: *Advances in Conceptual Modeling - ER 2013 Workshops, LSAWM, MoBiD, RIGiM, SeCoGIS, WISM, DaSeM, SCME, and PhD Symposium, Hong Kong, China, November 11-13, 2013, Revised Selected Papers*, J. Parsons and D.K.W. Chiu, eds, Lecture Notes in Computer Science, Vol. 8697, Springer, 2013, pp. 160–169.
- [61] 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.

- [62] S. Agarwal and A. Singhal, Handling skewed results in news recommendations by focused analysis of semantic user profiles, in: *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, 2014, pp. 74–79.
- [63] 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.
- [64] E. de Koning, F. Hogenboom and F. Frasin-car, News recommendation with CF-IDF+, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **10816 LNCS** (2018), 170–184.
- [65] E. Brocken, A. Hartveld, E. de Koning, T. van Noort, F. Hogenboom, F. Frasin-car and T. Robal, Bing-CF-IDF+: A Semantics-Driven News Recommender System, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **11483 LNCS** (2019), 32–47.
- [66] L.H. van Huijsdijnen, T. Hoogmoed, G. Keulers, E. Langendoen, S. Langendoen, T. Vos, F. Hogenboom, F. Frasin-car and T. Robal, Bing-CSF-IDF+: A Semantics-Driven Recommender System for News, *Communications in Computer and Information Science* **1259 CCIS** (2020), 143–153.
- [67] 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.
- [68] 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.
- [69] M. Capelle, F. Hogenboom, A. Hogenboom and F. Frasin-car, Semantic news recommendation using Wordnet and Bing similarities, in: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 296–302.
- [70] 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.
- [71] S. Kumar and M. Kulkarni, Graph Based Techniques for User Personalization of News Streams, in: *Proceedings of the 6th ACM India Computing Convention*, Compute '13, Association for Computing Machinery, New York, NY, USA, 2013. ISBN 978-1-4503-2545-5.
- [72] D. Werner and C. Cruz, A Method to Manage the Precision Difference between Items and Profiles: In a Context of Content-Based Recommender System and Vector Space Model, in: *2013 International Conference on Signal-Image Technology Internet-Based Systems*, 2013, pp. 337–344.
- [73] Q.M. Nguyen, T.T. Nguyen and T.D. Cao, Semantic-based recommendation method for sport news aggregation system, *Lecture Notes in Business Information Processing* **268** (2016), 32–47.
- [74] 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.
- [75] 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.
- [76] L.O. Colombo-Mendoza, J.A. García-Díaz, J.M. Gómez-Berbís and R. Valencia-García, A deep learning-based recommendation system to enable end user access to financial linked knowledge, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **10870 LNAI** (2018), 3–14.
- [77] 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.
- [78] 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*, 2018, pp. 417–426.
- [79] 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) (2019), 1–26. <https://dl.acm.org/doi/10.1145/3312738>.
- [80] 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*, 2019, pp. 2000–2010.
- [81] 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.
- [82] 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.
- [83] 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.
- [84] 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.
- [85] Z. Wang, W. Ma, M. Zhang, W. Chen, J. Xu, Y. Liu and S. Ma, Incorporating Knowledge and Content Information to Boost News Recommendation, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **12430 LNAI** (2020), 443–456.
- [86] J. Yang, J. Wan, Y. Wang and Y. Mao, Social network-based News Recommendation with Knowledge Graph, in: *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, Vol. 1, 2020, pp. 1255–1260.
- [87] J. Sun and M.D.M. BILLA SHAGAR, MUKG: Unifying Multi-Task and Knowledge Graph Method for Recommender System, in: *2020 2nd International Conference on Image Pro-*

- cessing and Machine Vision, IPMV 2020, Association for Computing Machinery, New York, NY, USA, 2020, pp. 14–21. ISBN 978-1-4503-8841-2.
- [88] Y. Shi, M. Larson and A. Hanjalic, Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges, *ACM Computing Surveys (CSUR)* **47**(1) (2014), 1–45, Publisher: ACM New York, NY, USA.
- [89] P. Bhargava, T. Phan, J. Zhou and J. Lee, Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data, in: *Proceedings of the 24th international conference on world wide web*, 2015, pp. 130–140.
- [90] Y. Koren, R. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer* **42**(8) (2009), 30–37, Publisher: IEEE.
- [91] Z. Xia, S. Xu, N. Liu and Z. Zhao, Hot news recommendation system from heterogeneous websites based on bayesian model, *The Scientific World Journal* **2014** (2014), Publisher: Hindawi.
- [92] S. Raza and C. Ding, News Recommender System Considering Temporal Dynamics and News Taxonomy, in: *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 920–929.
- [93] Y. Sun, J. Han, X. Yan, P.S. Yu and T. Wu, Paths: Meta path-based top-k similarity search in heterogeneous information networks, *Proceedings of the VLDB Endowment* **4**(11) (2011), 992–1003, Publisher: VLDB Endowment.
- [94] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick and J. Han, Recommendation in heterogeneous information networks with implicit user feedback, in: *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 347–350.
- [95] C. Shi, Z. Zhang, P. Luo, P.S. Yu, Y. Yue and B. Wu, Semantic path based personalized recommendation on weighted heterogeneous information networks, in: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 453–462.
- [96] R. Catherine and W. Cohen, Personalized recommendations using knowledge graphs: A probabilistic logic programming approach, in: *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 325–332.
- [97] G. Salton, A. Wong and C.-S. Yang, A vector space model for automatic indexing, *Communications of the ACM* **18**(11) (1975), 613–620.
- [98] G. Salton and C. Buckley, Term-weighting approaches in automatic text retrieval, *Information processing & management* **24**(5) (1988), 513–523.
- [99] R.L. Cilibrasi and P.M. Vitanyi, The google similarity distance, *IEEE Transactions on knowledge and data engineering* **19**(3) (2007), 370–383, Publisher: IEEE.
- [100] G. Bouma, Normalized (pointwise) mutual information in collocation extraction, *Proceedings of GSCL* **30** (2009), 31–40.
- [101] P. Shoval, V. Maidel and B. Shapira, An ontology-content-based filtering method, *Int'l Journal of Information Theories and Applications* **15** (2008).
- [102] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, Springer Science & Business Media, 2007.
- [103] J.J. Jiang and D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, *arXiv preprint cmp-lg/9709008* (1997).
- [104] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, *arXiv preprint cmp-lg/9511007* (1995).
- [105] D. Lin et al., An information-theoretic definition of similarity, in: *ICML*, Vol. 98, 1998, pp. 296–304.
- [106] 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.
- [107] Z. Wu and M. Palmer, Verb Semantics and Lexical Selection, in: *32nd Annual Meeting of the Association for Computational Linguistics*, 1994, pp. 133–138.
- [108] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge graph embedding: A survey of approaches and applications, *IEEE Transactions on Knowledge and Data Engineering* **29**(12) (2017), 2724–2743, Publisher: IEEE.
- [109] Y. Dai, S. Wang, N.N. Xiong and W. Guo, A survey on knowledge graph embedding: Approaches, applications and benchmarks, *Electronics* **9**(5) (2020), 750, Publisher: Multidisciplinary Digital Publishing Institute.
- [110] S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv:1205.2618* (2012).
- [111] 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.
- [112] 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.
- [113] 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.
- [114] 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.
- [115] 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.
- [116] 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.
- [117] 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*, 2019, pp. 950–958.
- [118] 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*, 2019, pp. 968–977.

- [119] 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.
- [120] 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, 2008, pp. 279–283.
- [121] F. Frasincar, J. Borsje and L. Levering, A semantic web-based approach for building personalized news services, *International Journal of E-Business Research (IJEER)* **5**(3) (2009), 35–53.
- [122] J. Zhang, B. Liu, J. Tang, T. Chen and J. Li, Social influence locality for modeling retweeting behaviors, in: *Twenty-third international joint conference on artificial intelligence*, 2013.
- [123] 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.
- [124] 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.
- [125] 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.
- [126] 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) (2020), 102142.
- [127] 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*, 2020, pp. 4255–4264.
- [128] B. Shao, X. Li and G. Bian, A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph, *Expert Systems with Applications* **165** (2021), 113764, Publisher: Elsevier.
- [129] K. Park, J. Lee and J. Choi, Deep neural networks for news recommendations, in: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2255–2258.
- [130] B. Bai, G. Zhang, Y. Lin, H. Li, K. Bai and B. Luo, CSRN: Collaborative Sequential Recommendation Networks for News Retrieval, *arXiv preprint arXiv:2004.04816* (2020).
- [131] 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.
- [132] M.F. Dacrema, S. Boglio, P. Cremonesi and D. Jannach, A troubling analysis of reproducibility and progress in recommender systems research, *ACM Transactions on Information Systems (TOIS)* **39**(2) (2021), 1–49, Publisher: ACM New York, NY, USA.
- [133] 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*, 2020, pp. 3597–3606.
- [134] 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* (2020).
- [135] M.M. Voit and H. Paulheim, Bias in Knowledge Graphs - An Empirical Study with Movie Recommendation and Different Language Editions of DBpedia, in: *3rd Conference on Language, Data and Knowledge (LDK 2021)*, D. Gromann, G. Sérasset, T. Declerck, J.P. McCrae, J. Gracia, J. Bosque-Gil, F. Bobillo and B. Heinisch, eds, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, pp. 14:1–14:13.
- [136] V.P. Dwivedi, C.K. Joshi, T. Laurent, Y. Bengio and X. Bresson, Benchmarking graph neural networks, *arXiv preprint arXiv:2003.00982* (2020).
- [137] 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.
- [138] 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*, 2011, pp. 125–134.
- [139] T. Hofmann, Latent semantic models for collaborative filtering, *ACM Transactions on Information Systems (TOIS)* **22**(1) (2004), 89–115.
- [140] J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters, *Communications of the ACM* **51**(1) (2008), 107–113.
- [141] 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).
- [142] 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.
- [143] 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.
- [144] 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, 2019, pp. 5329–5336.
- [145] 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.
- [146] 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.
- [147] 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.
- [148] R. Ying, D. Bourgeois, J. You, M. Zitnik and J. Leskovec, Gnnexplainer: Generating explanations for graph neural net-

- works, *Advances in neural information processing systems* **32** (2019), 9240.
- [149] 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.
- [150] G. Farnadi, P. Kouki, S.K. Thompson, S. Srinivasan and L. Getoor, A fairness-aware hybrid recommender system, *arXiv preprint arXiv:1809.09030* (2018).
- [151] 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.
- [152] 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.
- [153] 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.
- [154] C. Wu, F. Wu, X. Wang, Y. Huang and X. Xie, Fairness-aware News Recommendation with Decomposed Adversarial Learning, *arXiv preprint arXiv:2006.16742* (2020).
- [155] P. Symeonidis, L. Coda and M. Zanker, Counteracting the filter bubble in recommender systems: Novelty-aware matrix factorization, *Intelligenza Artificiale* **13**(1) (2019), 37–47, Publisher: IOS Press.
- [156] A. Gharahighehi, C. Vens and K. Pliakos, Fair multi-stakeholder news recommender system with hypergraph ranking, *Information Processing & Management* **58**(5) (2021), 102663.
- [157] N. Vo and K. Lee, The rise of guardians: Fact-checking url recommendation to combat fake news, in: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 275–284.
- [158] N. Vo and K. Lee, Standing on the shoulders of guardians: Novel methodologies to combat fake news, in: *Disinformation, Misinformation, and Fake News in Social Media*, Springer, 2020, pp. 183–210.
- [159] K.-C. Lo, S.-C. Dai, A. Xiong, J. Jiang and L.-W. Ku, All the Wiser: Fake News Intervention Using User Reading Preferences, in: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 1069–1072.
- [160] J.Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li and J. Liu, Content based fake news detection using knowledge graphs, in: *International semantic web conference*, Springer, 2018, pp. 669–683.
- [161] A. Tchechmedjiev, P. Fafalios, K. Boland, M. Gasquet, M. Zloch, B. Zapiiko, S. Dietze and K. Todorov, ClaimsKG: A knowledge graph of fact-checked claims, in: *International Semantic Web Conference*, Springer, 2019, pp. 309–324.
- [162] Y. Dun, K. Tu, C. Chen, C. Hou and X. Yuan, KAN: Knowledge-aware Attention Network for Fake News Detection, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, 2021, pp. 81–89, Issue: 1.
- [163] N. Newman, R. Fletcher, A. Schulz, S. Andi, C.T. Robertson and R.K. Nielsen, Reuters Institute Digital News Report 2021, *Reuters Institute for the Study of Journalism* (2021).
- [164] Y. Liu, H. Li, A. Garcia-Duran, M. Niepert, D. Onor-Rubio and D.S. Rosenblum, MMKG: multi-modal knowledge graphs, in: *European Semantic Web Conference*, Springer, 2019, pp. 459–474.
- [165] S. Thoma, A. Rettinger and F. Both, Towards holistic concept representations: Embedding relational knowledge, visual attributes, and distributional word semantics, in: *International Semantic Web Conference*, Springer, 2017, pp. 694–710.
- [166] H. Mousselly-Sergieh, T. Botschen, I. Gurevych and S. Roth, A multimodal translation-based approach for knowledge graph representation learning, in: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 2018, pp. 225–234.
- [167] R. Sun, X. Cao, Y. Zhao, J. Wan, K. Zhou, F. Zhang, Z. Wang and K. Zheng, Multi-modal knowledge graphs for recommender systems, in: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1405–1414.
- [168] The Gallup Organization, User language preferences online, *Analytical Report* (2014).
- [169] T.Q.Y.H. Chuhan Wu Fangzhao Wu, Empowering News Recommendation with Pre-trained Language Models, in: *SIGIR*, 2021.
- [170] R. Caruana, Multitask learning, *Machine learning* **28**(1) (1997), 41–75.
- [171] Y. Zhang and Q. Yang, A survey on multi-task learning, *arXiv preprint arXiv:1707.08114* (2017).
- [172] S. Ruder, An overview of multi-task learning in deep neural networks, *arXiv preprint arXiv:1706.05098* (2017).
- [173] 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.
- [174] 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.
- [175] 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.
- [176] M. Färber, F. Bartscherer, C. Menne and A. Rettinger, Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago, *Semantic Web* **9**(1) (2018), 77–129, Publisher: IOS Press.
- [177] S. Ji, S. Pan, E. Cambria, P. Marttinen and S.Y. Philip, A survey on knowledge graphs: Representation, acquisition, and applications, *IEEE Transactions on Neural Networks and Learning Systems* (2021), Publisher: IEEE.
- [178] C. Xiao, L. Sun and W. Ji, Temporal Knowledge Graph Incremental Construction Model for Recommendation, in: *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, Springer, 2020, pp. 352–359.
- [179] H. Mezni, Temporal Knowledge Graph Embedding for Effective Service Recommendation, *IEEE Transactions on Services Computing* (2021), Publisher: IEEE.
- [180] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural computation* **9**(8) (1997), 1735–1780, Publisher: MIT Press.