Matching Algorithms: Fundamentals, Applications and Challenges

Jing Ren, Feng Xia, *Senior Member, IEEE*, Xiangtai Chen, Jiaying Liu, Mingliang Hou, Ahsan Shehzad, Nargiz Sultanova, and Xiangjie Kong, *Senior Member, IEEE*

***Abstract*—Matching plays a vital role in the rational allocation of resources in many areas, ranging from market operation to people's daily lives. In economics, the term matching theory is coined for pairing two agents in a specific market to reach a stable or optimal state. In computer science, all branches of matching problems have emerged, such as the question-answer matching in information retrieval, user-item matching in a recommender system, and entity-relation matching in the knowledge graph. A preference list is the core element during a matching process, which can either be obtained directly from the agents or generated indirectly by prediction. Based on the preference list access, matching problems are divided into two categories, i.e., *explicit matching* and *implicit matching*. In this paper, we first introduce the matching theory's basic models and algorithms in explicit matching. The existing methods for coping with various matching problems in implicit matching are reviewed, such as retrieval matching, user-item matching, entity-relation matching, and image matching. Furthermore, we look into representative applications in these areas, including marriage and labor markets in explicit matching and several similarity-based matching problems in implicit matching. Finally, this survey paper concludes with a discussion of open issues and promising future directions in the field of matching.**

***Index Terms*—Information retrieval, knowledge graph, matching theory, recommender system, stable matching.**

I. Introduction

With the rapid growth of science and technology, an extensive repository of matching demands have emerged in different fields. Therefore, identifying, analyzing, and managing resources have become increasingly challenging. The term "matching" is generally defined as two objects suitably paired together or having the same appearance. Inspired by this definition, this paper's matching problem is defined as looking for a method to pair two or more objects together so that the pairs are suitably matched or have a similar appearance. Traditional matching problems (classified as explicit matching in this paper) emphasize finding the most suitable object according to their preference list. The final goal is to reach a stable or optimal state in a specific market [1]. This kind of problem is mainly studied in the fields of economics and mathematics. However, in the age of big data, obtaining everyone's preference list is time-consuming and nearly impossible. Therefore, matching problems nowadays in computer science pay more attention to predicting users' preference lists.

Unlike the commodity market, money is not involved during the process in the matching markets. In a broad sense, matching can be found in many disciplines and fields [2], [3]. For example, face recognition is an application of image matching [4], and information retrieval needs the technology of matching based on text similarity [5]. For an explicit matching, the preference lists for the matching process are provided by the agents themselves, and the ultimate goal of an explicit matching problem is to reach a stable or optimal state in a specific context [6], [7]. As for an implicit matching, we need to calculate the matching score between every two agents, thereby obtaining the preference lists by ranking their matching scores [8], [9].

Gale and Shapley [1] first proposed the concept of matching theory in 1962, which is a mathematical framework based on the game theory, and applied it to the marriage market and college admission. Since then, this theory has attracted the interests of a panoply of economists as it can solve many matching problems where the money is not involved [10], [11]. When only one side of the agents has a preference list, it is regarded as one-sided matching, while matching with preference lists on both sides is a two-sided matching. The two-sided matching theory considers that two sets of matching agents are selfish and rational. They seek to be matched to each other, for example, men and women in a marriage market, hospitals, and interns in a labor market, or buyers and sellers in an auction market. Each agent has a preference list based on their preferences over the set of agents on the opposite side. In such matching markets, stability is considered a fundamental requirement of successful matching. This concept implies that no pair of agents attempt to leave their current partners and form a new pair with each other. The optimal matching problems, like D2D communications in the wireless network, are mainly defined as optimization problems whereby the optimization objective function and constraints are given in accordance with the context.

Manuscript received August 21, 2020; revised November 30, 2020 and January 30, 2021; accepted February 24, 2021. Date of publication April 26, 2021; date of current version May 25, 2021. This work was supported by National Natural Science Foundation of China under Grant 61872054. *(Corresponding author: Feng Xia.)*

Jing Ren, Feng Xia, and Nargiz Sultanova are with the School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, VIC 3353, Australia (e-mail: ch.yum@outlook.com; f.xia@ieee.org; n.sultanova@federation.edu.au).

Xiangtai Chen, Jiaying Liu, Mingliang Hou, and Ahsan Shehzad are with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: chenxiangtai@outlook.com; jiaying_liu@outlook.com; teemohold@outlook.com; ahsan.shehzad@outlook.com).

Xiangjie Kong is with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: xjkong@ieee.org).

Digital Object Identifier 10.1109/TETCI.2021.3067655

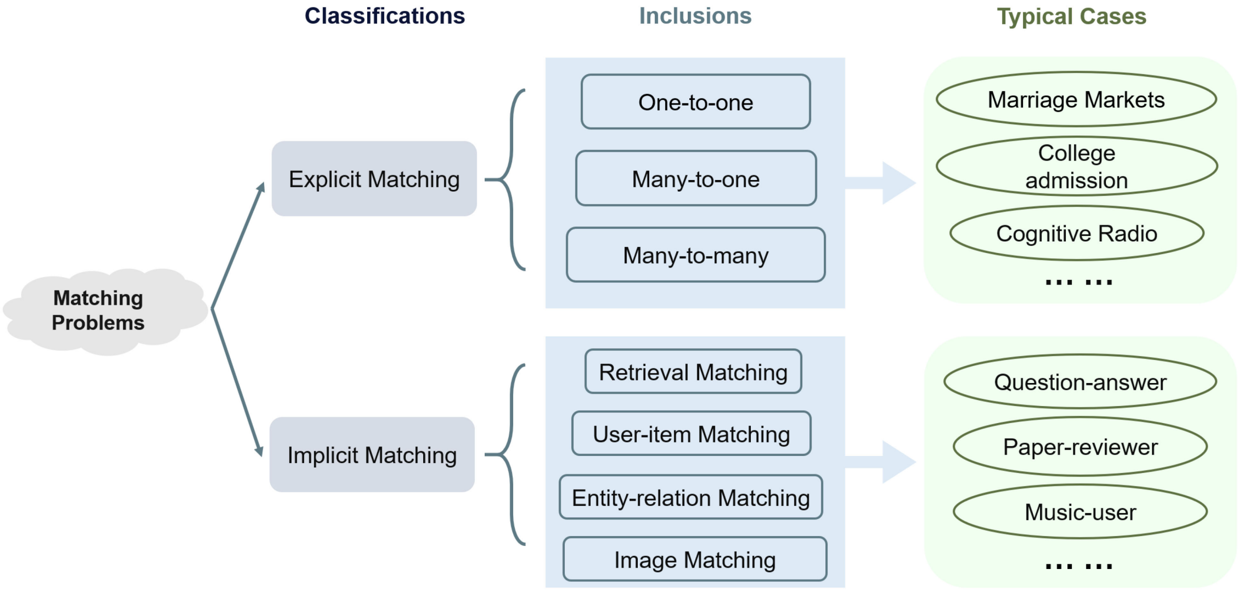

Fig. 1. The classifications and applications of matching problems.

In most cases, the preference lists of agents cannot be obtained directly due to the data sets' limitations. Therefore, the generation of preference lists becomes the first step of matching. In the field of information retrieval, a wide spectrum of methods have been proposed to calculate the similarity between texts [12]. This process is vital for many practical applications, such as question-answer matching in search engines. Another important field is recommender systems [13], [14], where we need to judge the preference of users, such as recommending music and movies to users, recommending merchandise to consumers, or recommending papers to reviewers. Furthermore, with the help of knowledge graph [15], [16], applications such as link prediction in relationship network [17], [18] and user interest prediction in recommender system [19], [20] can be realized more accurately.

Instead of selecting only one research direction, either explicit matching problems [21], [22] or implicit matching problems [23], [24], this paper comprehensively reviews both kinds of matching problems in different fields. The advantage of this survey is to provide readers with more branches in both economics and computer science, such as stable matching theory, recommendation system, information retrieval, knowledge graph, and face alignment. Besides, readers who are not familiar with the stable matching theory may get a new idea of applying it to solve real-world problems in different fields. To the best of our knowledge, this paper is the first effort in providing a review and classification of matching problems according to the access of preference lists. In explicit matching problems such as marriage market and medical intern, the stable matching theory algorithm (e.g., deferred acceptance algorithm) and its variants are used to match selfish, rational agents with known preference lists. On the other hand, implicit matching problems aim to match entities by generating preference lists, like paper-reviewer matching, user-merchandise matching, and question-answer matching. Explicit matching is mainly introduced from three perspectives: one-to-one, many-to-one, and many-to-many matching. As for implicit matching, we only present some typical matching problems and their methods in computer science. The main details of the classification of matching problems are shown in Fig. 1.

The rest of this paper is structured as follows. Section II presents the concepts and methods of explicit matching, and Section III describes parts of emerging issues and methods in implicit matching. Comprehensive applications in both explicit and implicit matching are introduced in Section IV. We discuss some future trends and challenges in Section V, and conclude this paper in Section VI. We summarize all algorithms of explicit and implicit matching problems surveyed in this paper in Table I. It should be noted that there is no systematic classification in explicit matching algorithms. Therefore, references of explicit matching shown in this table are classified according to real-world applications.

II. Explicit Matching

Explicit matching refers to problems in which the agents themselves give preference lists. This section provides the classification of these kinds of problems into three types; namely one-to-one, many-to-one, and many-to-many. Then, the process of the famous deferred acceptance algorithm is introduced in detail. Besides, an illustration of many-to-one matching is shown in Fig. 2.

A. *One-to-One*

We start by introducing the basic matching theory in the classical one-to-one marriage model [25]. There are two sets of agents in this model, men and women, represented by $M = \{m_1, m_2, \ldots, m_n\}$ and $W = \{w_1, w_2, \ldots, w_k\}$, respectively. Each agent has a complete preference list over the agents on the other side. The one-to-one matching of men and women is the outcome of the marriage problem. Assume that an agent m has

TABLE I
SUMMARISATION OF EXPLICIT AND IMPLICIT MATCHING ALGORITHMS AND APPLICATIONS

Category	Sub-category	Alogorithms	references	Applications
Explicit Matching	One-to-one Matching	-	[1], [25], [26], [27]	marriage market
	Many-to-one Matching	-	[7], [11], [28], [29], [30]	job matching
			[31], [32], [33]	pay matching
	Many-to-many Matching	Optimization algorithms	[6], [34], [35], [36]	cognitive radio networks; D2D communications

Implicit Matching	Retrieval Matching	Traditional matching algorithms	[37], [38], [39], [40], [41], [42]	machine translation;
				expertise matching;
				question-answer matching

		Representation-based algorithms	[43], [44], [45], [46]	
		Interaction-based algorithms	[47], [48], [49], [50], [51], [52], [53], [54], [55], [56]	
	User-item Matching	Basic algorithms	[57], [58]	recommendation systems
		Represenation-based algorithms	[59], [60], [61], [62], [63]	
		Matching function-based algorithms	[64], [65], [66], [67]	

	Entity-relation Matching	Factorization-based algorithms	[68], [69], [70], [71]	recommendation systems;
				knowledge fusion;
				information retrieval

		Neural network-based algorithms	[72], [73], [74], [75]	
		Translational distance based algorithms	[76], [77], [78], [79], [80]	

	Image Matching	Area-based algorithms	[81], [82], [83], [84], [85]	robot vision;
				object recognition;
				medical image diagnosis

		Feature-based algorithms	[86], [87], [88], [89], [90], [91], [92], [93], [94], [95]	

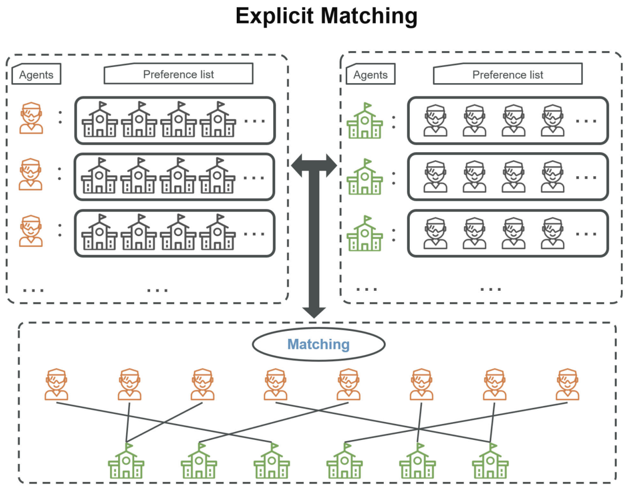

Fig. 2. An illustration of student-school matching (Explicit Matching).

a preference list $Per(m) = w_1, w_2, m, w_3, w_4, \ldots$. This means that they prefer agent w_1 to w_2 and prefer remaining single (m) over partnering with w_3 or w_4. We use $w_i >_m w_j$ to express that an agent m prefers agent w_i to w_j. In addition, $w_i \geq_m w_j$ means that for the agent m, the preference for w_i is no less than for w_j.

The One-to-one matching model (taking marriage market as an example here) is defined as follows:

Definition 1: An outcome of one-to-one marriage model is a matching μ from $M \cup W$ to $M \cup W$ such that:

- *For any $m \in M$, $\mu(m) \in W \cup \{m\}$,*
- *For any $w \in W$, $\mu(w) \in M \cup \{w\}$,*
- *For any $m \in M$ and $w \in W$, $w = \mu(m)$ if and only if $m = \mu(w)$.*

Note that in the process of one-to-one matching problems, every man m can only be matched with one woman w, and agent m remains unmatched (single) if $\mu(m) = m$. The goal of an explicit matching is to reach stable status for all pairs:

Definition 2: A matching μ is pairwise stable if any individuals or pairs do not block it.

Definition 3: A matching $\mu(x)$ is blocked if x prefers remaining single to being matched with someone (for any $x \in M \cup W$, $x >_x \mu(x)$). Both matchings $\mu(w)$ and $\mu(m)$ are blocked if w and m prefer each other to their current partners ($w >_m \mu(m)$ and $m >_w \mu(w)$).

The One-to-one matching is to obtain the most-preferred allocation that completely suits both agents' preference lists and reaches a stable status at the same time.

B. Many-to-One

Many-to-one or one-to-many matching indicates that agents of one side are allowed to be matched with more than one agent on the other side. Typical examples of many-to-one matching arise in the person-institution matching (shown in Fig. 2) problems such as student-college or doctor-hospital matching [96]. The rule behind this kind of matching is that agents on one side (e.g., institutions) can provide many of the same positions for the agents on the other side (e.g., students), but the reverse is not valid.

Firstly, many-to-one matching model requires two finite disjoint sets, $P = \{p_1, p_2, \ldots, p_n\}$ and $V = \{v_1, v_2, \ldots, v_m\}$ representing the sets of people and institutions. Similar to one-to-one matching, $Per(v) = p_1, p_2, v, p_3, p_4, \ldots$ means that institution v prefers p_1 to p_2 and prefers keeping the position unfilled over other people like p_3 and p_4.

Unlike one-to-one matching, each institution has a positive quota q to represent the maximum number of people it can be matched with. For a given institution v, its quota can be written as q_v. Therefore, in many-to-one matching, one person can only be matched with one institution and one institution can be matched with a fixed number of people (q_v) at most. Unmatched positions in the preferences of people or institutions can be regarded as self-matching.

Many-to-one matching can be defined as:

Definition 4: A matching μ is a function from the set $V \cup P$ into the set of unordered families of elements of $V \cup P$ such that:

1) *$|\mu(p)| = 1$ for every person $p \in P$ and $\mu(p) = p$ if p is unmatched;*

2) $|\mu(v)| = q_v$ *for every institution* $v \in V$*; if the number of people* k *in* $\mu(v)$ *is such that* $k < q_v$*, then* $\mu(v)$ *will have* $q_v - k$ *copies of* v*;*
3) $\mu(p) = v$ *if and only if* p *is an element of* $\mu(v)$*.*

Here, $\mu(p_1) = v$ means that person p_1 is matched with institution v and $\mu(v) = \{p_1, p_2, v, v\}$ means that the institution v with the quota $q_v = 4$ has been matched with two people p_1 and p_2 and has two unfilled matching positions. Pairwise stability in many-to-one matching is defined in the same way as one-to-one matching, and any coalition cannot block a stable matching.

C. Many-to-Many

A many-to-many matching problem refers to when the number of matchings for the agents on both sides are not restricted to one [97].

Suppose that two disjoint finite sets of agents are $S = \{s_1, s_2, \ldots, s_n\}$ and $T = \{t_1, t_2, \ldots, t_m\}$, with q_s and q_t being the respective quotas for agents $s \in S$ and $t \in T$. We generalise the definition of many-to-one matching as follows:

Definition 5: A matching μ *is a function from the set* $S \cup T$ *into the set of unordered families of elements of* $S \cup T$ *such that:*

1) $|\mu(s)| = q_s$ *for every agent* $s \in S$*; if the number of agents* k *in* $\mu(s)$ *is such that* $k < q_s$*, then* $\mu(s)$ *will contain* $q_s - k$ *copies of* s*;*
2) $|\mu(t)| = q_t$ *for every agent* $t \in T$*; if the number of agents* l *in* $\mu(t)$ *is such that* $l < q_t$*, then* $\mu(t)$ *will contain* $q_t - l$ *copies of* t*;*
3) $s \in \mu(t)$ *if and only if* $t \in \mu(s)$*.*

In the real-world matching markets, there are some classical many-to-many cases, i.e., medical intern matching [98] in the U.S., rider-driver matching in ride sharing [99], and the teacher-student matching [100]. Besides, pre-caching the various files for mobile users is also a many-to-many matching problem to improve the network performance in D2D netwrok [101].

D. Deferred Acceptance Algorithm

To achieve the goal of stable matching, the Deferred Acceptance (DA) algorithm was proposed and applied to the marriage markets and college admission problems [1]. Stable matching can also be achieved in many other markets by deploying the DA algorithm and its variants [102], [103].

In the DA algorithm, the agents on one side propose a pair formation with the agents of the other side according to their preference, and an iterative procedure follows. The other set of agents reject the offers of any agents that are not acceptable to them, and each agent that receives more than one offer rejects all but its most preferred one. Any agent whose offer is not rejected at this point is tentatively matched with the agent they proposed to; however, their current request could be rejected during the next iteration if the agent on the other set receives a better offer. In the next iteration, any agent whose proposal was rejected at the previous step makes new proposals to the agents on the other side based on the preference list. An agent will continue to propose according to the preference list as long as there are acceptable agents on the other side that are single. The agents on the other side continue rejecting the old proposals if the more preferred one comes along. The DA algorithm stops when there are no rejected agents that can make new proposals, at which stage the whole matching process terminates, and all the tentative matchings become final ones. A flow chart of the DA algorithm in the marriage market is shown in Fig 3.

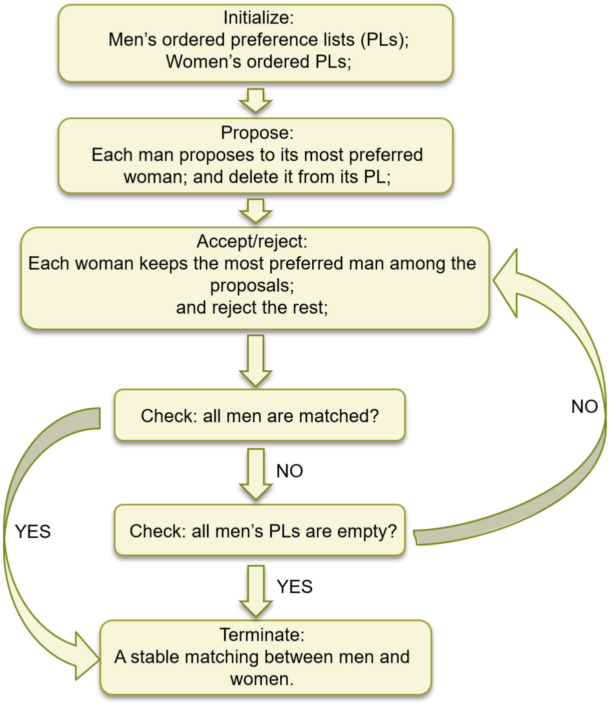
Initialize:
Men's ordered preference lists (PLs);
Women's ordered PLs;
Propose:
Each man proposes to its most preferred woman; and delete it from its PL;
Accept/reject:
Each woman keeps the most preferred man among the proposals;
and reject the rest;
Check: all men are matched?
NO
Check: all men's PLs are empty?
NO
YES
YES
Terminate:
A stable matching between men and women.

Fig. 3. Flow chart of DA algorithm in marriage market.

Due to its efficiency in solving conflicts of interest among agents in the market, Xu and Li [104] advocate using a stable matching framework instead of utility-based optimization, solve network problems. Besides, matching theory can help a network designer in selecting an appropriate matching model for a specific application in wireless communications [105]. With a general understanding of matching theory, many resource allocation problems can be regarded as matching problems between agents.

III. Implicit Matching

In this paper, we propose a new concept, implicit matching, which is defined as looking for a method to pair two or more objects with similar appearances together. Unlike explicit matching, implicit matching focuses on the process of calculating the matching score. This section will introduce some methods of implicit matching from the perspectives of retrieval matching, user-item matching, entity-relation matching, and image matching.

A. Retrieval Matching

From the perspective of information retrieval, retrieval matching is how users actively input the query that expresses their needs clearly and obtain the information they want from the search engine's database. Fig. 4 describes the whole matching process of query and document in the search engine. Search

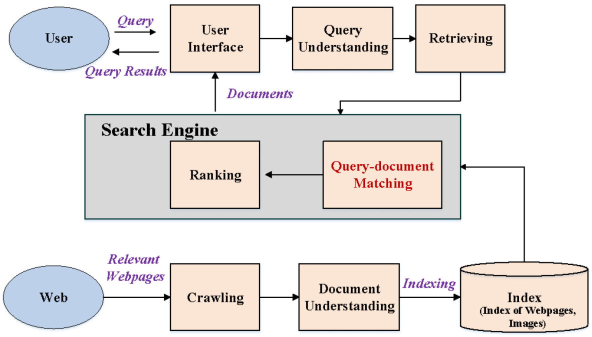

Fig. 4. The process of information retrieval. The words with purple font means the transmitted data, and red font means the problem that need to be sovled.

engines act as a bridge between users and webpages. It matches the query searched by the user and the document in the database. The search engine extracts information from webpages and stores relevant information into the database. When the user enters the query, the search engine begins to analyze the query and finds the index in database that matches the query. The extracted webpages will be comprehensively ranked according to different conditions. Generally speaking, the search engine completes the final ranking through a series of steps, including query processing, query-document matching, relevance calculation, filtering, and adjustment, then returns the ranking results to users. In other words, information retrieval can be abstracted as a textual relevance matching problem between web pages and users' search queries. With the development of technology, the algorithms for textual relevance matching can be divided into three types: traditional, representation-based, and interaction-based matching algorithms.

1) Traditional Matching Algorithms: Term Frequency-Inverse Document Frequency (TF-IDF) is a common weighting technique for textual relevance matching. It is a statistical procedure used to assess the importance of a word for a file set or one of the files in a corpus [37]. The core idea of TF-IDF is to transform the problem of finding the matching degree of query q and the document d to finding the conditional probability that query q comes from the document d. Given the individual document d from the document corpus D and the query q containing the word w, TF-IDF calculates the importance (weight) of word w in a document $d \in D$ as follows [38]:

$$TF-IDF_{w,d} = TF_{w,d} \cdot IDF_w = f_{w,d} \cdot \log \frac{|D|}{n(w)}, \quad (1)$$

where $f_{w,d}$ is the frequency with which w appears in d (TF component), $|D|$ is the number of documents in D, and $n(w)$ is the number of documents in D that contain the word w (IDF component). The value of TF-IDF is simply the product of the TF component and IDF component. Generally speaking, the larger the TF-IDF of a certain word w in the document d, the higher its importance in this document. Therefore, by calculating the TF-IDF of each word in the document, the first few words with larger TF-IDF are considered as the keywords of this document. Variations of TF-IDF are often used in search engines to measure the degree of correlation between files and users' queries.

BM25 (Best Match 25) [38], [39] is a typical TF-IDF matching algorithm for evaluating the relevance between the query and documents based on probability. The main idea is to calculate the weights of the words in the query for each document and sum these weights over the set of words in the query to measure its relevance to the document. The relevance (matching) score $score(q, d)$ of a document d in a query q can be formulated as:

$$score(q, d) = \sum_{w \in q} TF_{w,d} \cdot IDF_w. \quad (2)$$

Here $TF_{w,d}$ is a normalised TF component, and IDF_w is the weight for the word w in the query. $TF(w, d)$ is given by:

$$TF_{w,d} = \frac{f_{w,d}(k+1)}{f_{w,d} + k\left(1 - b + b\frac{|d|}{avgl(d)}\right)}, \quad (3)$$

where $|d|$ is the length of a document d, $avgl(d)$ is the average length of the documents in the corpus that d comes from, and k, b are the parameters. The default $IDF(w)$ is generally calculated as:

$$IDF_w = \log \frac{|D| - n(w) + 0.5}{n(w) + 0.5}. \quad (4)$$

The meanings of $|D|$ and $n(w)$ are the same as Eq. 1.

Latent semantic analysis (LSA) [40] is used initially in semantic retrieval to solve the problem of different words for the same objects and concepts (synonymy). In some cases, LSA is also called latent semantic indexing (LSI). The purpose of LSA is to discover the implied semantic dimension (can be understood as "topic" or "concept") from the text. The basic idea of LSA is to reduce high-dimensional documents to low-dimensional space, which is called latent semantic space. The steps of LSA can be summarized as:

1) Analyze the document corpus and establish word-document matrix, of which rows are unique words and columns represent documents;
2) Use a reduced-rank Singular Value Decomposition (SVD) to create a reduced-dimension approximation of the word-document matrix;
3) Compute similarities of documents in the reduced dimensional space (latent semantic space).

LSA can be used to compare different documents or match queries to the documents. The similarity of texts is semantic, so documents can be close even if they do not share the same words. The word-document matrix is usually created using the TF-IDF technique to assign weights to the words. The similarity of a document and a query represented as two vectors can be calculated using the cosine distance between them. Values close to 1 illustrate that they are very similar. Since LSA maps words and documents into latent semantic space, it can remove some noise in the original vector space and improve information retrieval accuracy.

Hofmann brings LSA into a probabilistic space and proposes Probabilistic Latent Semantic Analysis (PLSA) [41], which uses the Expectation Maximization (EM) algorithm to estimate the parameters of the model. PLSA is based on the idea that the documents are modeled as multinomial probability distributions of topics, and the topics are modeled as multinomial distributions of words. The number of parameters in PLSA grows linearly with the number of training documents, making it prone to overfitting. The model cannot predict distributions of new documents as it learns the topic distributions only for the documents it has trained.

To overcome these problems, Beli *et al.* [42] propose a three-level hierarchical Bayesian model named Latent Dirichlet Allocation (LDA). Being an extension of PLSA, LDA is also a probabilistic and generative model with documents modeled as mixture over a set of topics. As opposed to assuming that the topic weight parameters are fixed and unknown in PLSA, LDA treats them as a hidden random variable generated by a Dirichlet distribution.

2) Representation-Based Algorithms: Representation based information retrieval methods focus on learning the representations of the queries and documents. Sentences are encoded into their embedding without any cross-interaction [106]. An advantage of these methods is that they can be used for transfer learning to other natural language tasks. There are mainly two processes to complete the matching:

- Calculate the representation $\phi(q)$, $\phi(d)$ for the query q and the document d, respectively;
- Conduct the matching $M(\phi(q), \phi(d))$.

We will now introduce some typical methods for each process in detail. Huang *et al.* [43] propose the Deep Structured Semantic Model (DSSM), which is a supervised learning method to learn the representations of queries and documents using a Deep Neural Network (DNN) framework. In the input layer of the model, the DNN processes the word vectors of queries or documents by taking advantage of word hashing to reduce the dimensionality of the vectors, after which it pushes the hashed features through multiple layers of non-linear projections. As a result, the queries and documents are mapped into concept vectors in the reduced-dimensional semantic space. The matching score between two vectors is measured using the cosine similarity of their respective concept vectors.

DSSM uses a bag-of-words approach to treat documents and queries, which ignores the word order and context information. Shen *et al.* [44] propose a convolutional latent semantic model (CLSM, which can also be called CNN-DSSM) incorporating a convolution-pooling structure to generate the representations for queries and documents to overcome the latter drawback of DSSM. The major difference between CLSM and DSSM lies in the convolution and pooling layers of the neural network, through which CLSM can extract the contextual information for each word within a context window. However, it is challenging to preserve context information at long intervals due to the size limitation of the CLSM window (convolution core).

Palangi *et al.* [45] suggest combining Long-Short-Term Memory (LSTM) with DSSM and propose LSTM-DSSM aiming at solving the disadvantage of CNN-DSSM's inability to capture long-range context features. It uses a variant of LSTM, LSTM with peephole [107] to generate the error signal from the cosine similarity of the embedding vectors. Experiments on the information retrieval task demonstrate the effectiveness of the method.

It is acknowledged that recursive neural networks (RNNs) can embed text into fixed-length vectors, which have also shown good performance on various natural language processing tasks. Considering that RNNs require the input to be structured and make data preparation more complicated and time-consuming. Choi *et al.* [46] devise a sentence encoder model to efficiently learn to compose task-specific tree structures from plain text data, whose architecture is a tree-structured long short-term memory (Gumble Tree-LSTM). This model introduces the composition query vector to compute the candidate parents' validity and selects the appropriate parent according to validity scores.

3) Interaction-Based Algorithms: Instead of directly learning the semantic representation vectors of queries and documents, interaction based methods let them interact in advance at the bottom layer and establish some basic matching signals, i.e., the matching of word and word, and then try to integrate these basic matching signals into a matching score. There are also two major steps to complete the matching:

- Construct the basic low-level signals $s(q)$, $s(d)$ for the query q and the document d, respectively;
- Aggregate the matching patterns $A(s(q), s(d))$.

Hu *et al.* [47] adapt the convolutional strategy in matching sentences and propose two architectures ARC-I and ARC-II which can capture the structures of sentences at both the same and different levels. The first step for ARC-II is carrying out ARC-I, which aims to represent the sentences and compares the representations with a multi-layer perceptron (MLP) [48]. Specifically, it needs to set the window size and then extract the trigram vectors of two sentences. Firstly, by using the convolutional strategy, a matrix which is the matching signal of two sentences is constructed. By using continuous convolution and pooling, a vector of a certain length is produced. Finally, ARC-I uses MLP to provide the matching scores between sentences. However, as ARC-I keeps the interaction between two sentences until their final representations are created, it can lose important details for the matching. ARC-II overcomes this drawback by allowing the sentences to interact before their final representations are made.

Following [47], Pang *et al.* [49] propose MatchPyramid which is a convolutional model that tackles text matching problem through image recognition. The matching matrix which is obtained using cosine or dot-product similarity between word vectors is viewed as an image. MatchPyramid solves the problem that there is no matching signal for unigram in ARC-II.

Wan *et al.* [50] propose to solve the text matching problem recursively. The interaction between two texts at any position is presented as a composition of interactions between their prefixes and the word level interaction at the position. To model the recursive structure of the matching, Match-SRNN method utilizing neural networks is introduced. It constructs a similarity tensor to capture word interactions and a spatial RNN with gated recurrent units is subsequently applied to it. Finally, the matching score is calculated based on the global interaction using a linear function. That is, given the matching score $h(\cdot)$ of the prefixes and the similarity score $s(\cdot)$ of the words, the matching score for the query and document of lengths m and n is calculated as:

$$score(q, d) = Wh_{m,n} + b, \tag{5}$$

where W and b are parameters and

$$h_{i,j} = f(h_{i-1,j}, h_{i,j-1}, h_{i-1,j-1}, s_{i,j}), \tag{6}$$

where $i = 1, \ldots, m,\ \ j = 1, \ldots, n$. In Match-SRNN, 2D-GRU [51] is used for function f. The approach used in Match-SRNN approximates a dynamic programming process for information retrieval.

Parikh *et al.* [52] use attention mechanism to decompose the Natural Language Inference problem (NLI) into subproblems. The proposed model consists of three parts: attention, comparison, and aggregation. Assuming that the vector for each word in the query is $q = [a_{w_1}, \ldots, a_{w_m}]$, and the vector for each word in the document is $d = [b_{w_1}, \ldots, b_{w_n}]$, the elements of the query and the document are aligned by adopting the attention mechanism. By using a neural network function f, the aligned phrases $\{a_{w_i}, B_{w_i}\}_{i=1}^{m}$ and $\{b_{w_j}, A_{w_j}\}_{j=1}^{n}$, where B_{w_i} is the subphrase in d that is aligned to a_{w_i}, and likewise for A_{w_j}, are compared and the comparison vectors v_{1i} and v_{2j} are produced. Summing up the comparison vectors, respectively, and applying another neural network function to the result, we can get the similarity score.

Wan *et al.* [53] present an architecture to match two sentences with multiple positional sentence representations generated by a bidirectional long short term memory (Bi-LSTM). Through k-Max pooling and a multi-layer perceptron, the matching score is finally obtained by aggregating such interactions between these different positional sentence representations. The feature vector q obtained by k-Max pooling is first fed into a full connection hidden layer to get a higher level representation r. Then, the matching score s is calculated as:

$$r = f(W_r q + b_r), s = W_s r + b_s, \tag{7}$$

where W_r and W_s denote the parameter matrices, and b_r and b_s are corresponding biases.

In [54], the authors propose the idea that semantic matching between a question and answer is mainly relevant with semantic similarity rather than spatial positions. Therefore, instead of position-shared weighting scheme in CNNs, they combine different matching signals and incorporate question term importance learning using attention mechanism with value-shared weighting scheme.

Although attention mechanism is helpful to capture the semantic relationship and properly align the elements of two sentences, simply using a summation operation in the attention mechanism cannot retain original features enough. Therefore, Kim *et al.* [55] propose a densely-connected co-attentive recurrent neural network. The recurrent and co-attentive features are connected from the bottom to the top layer without any deformation. The proposed DRCN consists of three components: 1) word representation layer, 2)attentively connected RNN, and 3) interaction and prediction layer. In the interaction layer, the representations p and q for the two sentences P and Q are aggregated in various ways. Then, the final feature vector v for semantic sentence matching is obtained as:

$$v = [p; q; p + q; p - q; |p - q|], \tag{8}$$

wherein, all operations in the equation are performed element-wise to predict the relationship between two sentences.

Mitra *et al.* [56] first combine the interaction-based model and representation-based model. They use two separate deep neural networks, one of which matches the query and the document by using the local representation and the other using learned distributed representations. Experiments on the web page ranking task illustrate that the combination of the models outperforms the state-of-the-art methods.

In summary, most literal matching algorithms such as TF-IDF and BM25 mainly depend on the word coverage degree between two documents, which have several disadvantages, such as semantic limits, structural limits, and knowledge limits. Although LSA and other semantic analysis models can make up for some of the disadvantages, they cannot fully replace the literal matching models. With the successful application of deep learning in computer vision, speech recognition, and recommendation system in recent years, many researchers devote to applying deep semantic matching models to natural language processing tasks to reduce the cost of feature engineering. The representation-based model focuses more on constructing the representation layer, where the text is transformed into a unique overall representation vector. However, compared with the representation based methods that can calculate the document embedding in advance, interaction based methods cannot calculate the semantic vector of the text in advance during the online prediction task, which will result in high online computation costs.

B. User-Item Matching

In the era of data explosion, a recommender system provides a convenient way for users to obtain their items of interest as accurately as possible [108]. Therefore, personalized items recommendation are nowadays ubiquitous and have been performed in many practical applications, such as the recommendation of music in music player software, users on social websites and merchandise on shopping websites. Unlike searching, the recommendation is to push information or items to users by guessing their preferences or interests. An illustration of recommending music to users is shown in Fig. 5

1) Basic Algorithms: Among all recommender systems, traditional recommender systems can be classified into three primary categories from the perspective of the data source [109]: collaborative filtering (CF), content-based recommendation (CR) and hybrid recommender system.

It is commonly acknowledged that CF is the most popular recommendation algorithm at present and is mainly classified from three perspectives, user-based, item-based, and model-based CF [57]. The first two categories recommend items in terms of the similarity of users or items, while the model-based CF adopts some machine learning methods such as matrix factorization, clustering and restricted Boltzmann machine to build models.

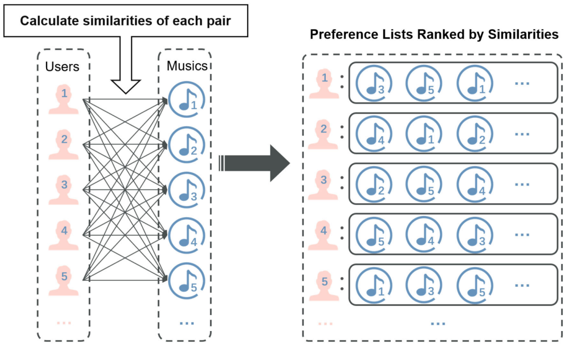

Fig. 5. An illustration of recommending musics to users (Implicit Matching). First, the similarity (matching score) between each user and music is calculated according to the users' recordings, such as searching histories. Then, every user's preference list is obtained by ranking the similarity metric values, which can be used as a reference for recommending new music to users.

The emergence of content-based recommendation algorithms is early in its development and it can be performed in three steps.

- Extract some features for each item (the content of the item) to represent it.
- Learn the user's preferable characteristics by utilizing the user's attitudes towards an item in the past.
- Recommend a list of the most relevant items for a user by comparing the user profile obtained in the previous step with the characteristics of the candidate items.

Matrix factorization [58] is a standard method in recommender systems. A sparse user-item matrix is decomposed into a product of a user embedding matrix $U^{m\times k} = [p_1, \ldots, p_m]^T$ and an item embedding matrix $V^{n\times k} = [q_1, \ldots, q_n]^T$, where m is the number of users, n is the number of items, and k is the number of latent factors, $k < min(m, n)$. The predicted preference value of item v by the user u, denoted by $\widehat{y}_{uv}$ is represented by the inner product of the user embedding $p_u \in \mathbb{R}^k$ and item embedding $q_v \in \mathbb{R}^k$,

$$\widehat{y}_{uv} = {p_u}^T q_v. \tag{9}$$

We minimize the square of the difference between the predicted and the true preferences of the user, and to avoid the problem of over-fitting, a regularization function is added as a constraint. The loss function is therefore defined as follows:

$$L = \underbrace{\sum_u \sum_v w_{uv}(y_{uv} - \widehat{y}_{uv})^2}_{Prediction\,error} + \underbrace{\lambda \left(\sum_u \parallel p_u \parallel^2 + \sum_v \parallel q_v \parallel^2 \right)}_{L2\,regularizer}. \tag{10}$$

Here w_{uv} is a parameter that is used to balance the number of zero and non-zero values in the user-item matrix.

2) Representation-Based Algorithms: The methods of representation learning aim to learn the embeddings of the user and the item by the neural network. A simple inner product or the cosine value between their embeddings can be calculated as their final matching score. Methods of representation learning are introduced from two perspectives, CF methods where models are built merely based on the user-item interaction matrix, and methods of CF based on combining both user-item interaction and auxiliary information at the same time.

Sedhain *et al.* [59] propose a novel autoencoder framework for collaborative filtering named AutoRec, which is a discriminative model based on autoencoders using a backpropagation algorithm to make the output of the model equal to the input. Compared with matrix factorization approaches, the item-based AutoRec model only embeds items into latent space. Subsequently, Wu *et al.* [60] present a novel method called Collaborative Denoising Auto-Encoder (CDAE) for the top-N recommendation, which assumes that the observed user-item interactions are corrupted and, as such, is trained to reconstruct the data from the partially corrupted version of the preference set. CDAE differs from AutoRec by adding the userID to the input and excluding it from the reconstructed output layer, which is inspired by the thinking of SVD++ [61]. To make the best of explicit ratings and non-preference feedback, Xue *et al.* [62] propose a new deep matrix factorization model with a neural network that projects users and items into low-dimensional vectors in the latent space. The input matrix for the model incorporates both explicit and implicit feedback from the users. A new loss function based on cross entropy is constructed that considers both explicit ratings and implicit feedback for the optimization.

The user u_i and item v_j are mapped to a latent space as follows:

$$p_i = f_{\theta_n^U}(\ldots f_{\theta_3^U}(W_{U2} f_{\theta_2^U}(Y_{i*} W_{U1}))\ldots), \tag{11}$$

$$q_j = f_{\theta_n^I}(\ldots f_{\theta_3^I}(W_{V2} f_{\theta_2^I}(Y_{*j}^T W_{V1}))\ldots). \tag{12}$$

Here W_{U1} and W_{V1} are the first layer weighting matrix for U and V, respectively, and W_{U2} and W_{V2} for the second layer, and so on. From the user-item interaction matrix Y, each user p_i is represented as a high-dimensional vector of Y_{i*}, which represents the i_{th} user's ratings across all items. Each item q_j is represented as a high-dimensional vector of Y_{*j}, which represents the j_{th} item's ratings across all users. The predicted $\widehat{Y}_{ij}$ is calculated from the dot product of p_i and q_j:

$$\widehat{Y}_{ij} = p_i^T q_j. \tag{13}$$

Additionally, a new loss function is designed to consider both explicit ratings and implicit feedback for better optimization, which is shown as follows:

$$L = - \sum_{(i,j)\in Y^+ \cup Y^-} \left(\frac{Y_{ij}}{\max(R)} \log \widehat{Y}_{ij} + \left(1 - \frac{Y_{ij}}{\max(R)}\right) \log(1 - \widehat{Y}_{ij}) \right). \tag{14}$$

Here, R denotes the rating matrix, where R_{ij} is the rating of user i on item j. The max score in all ratings $max(R)$ is used for normalization, so that different values of Y_{ij} have different influences to the loss.

In multimedia recommendation, to address the problem of implicit feedback, Chen *et al.* [63] propose a novel CF framework named Attentive Collaborative Filtering (ACF), which is an improvement based on SVD++ [61]. By seamlessly incorporating two attention modules into neighborhood models, they can infer the underlying user preferences encoded in the implicit user feedback.

3) Matching Function-Based Algorithms: The matching function learning methods are not directly based on learning users' and items' embeddings but by using a neural network to fuse the basic matching signal and subsequently obtain their matching score. Based on using CF models, He *et al.* [64] can learn an arbitrary function from data and first devise a general framework named NCF for collaborative filtering based on neural networks. To design a better, dedicated interaction function for modeling the latent feature interactions between users and items, the offered solution is to let Generalized Matrix Factorization (GMF) and Multi-Layer Perceptron (MLP) share the same embedding layer, and then combine the outputs of their interaction functions. In order to provide more flexibility to the joint model, GMF and MLP are allowed to learn separate embeddings, and then be combined by concatenating their last hidden layer. The formulation of which is given as follows:

$$\begin{aligned}\phi^{GMF} &= p_u^G \odot q_i^G,\\ \phi^{MLP} &= a_L\left(W_L^T\left(a_{L-1}\left(\ldots a_2\left(W_2^T\begin{bmatrix}p_u^M\\ q_i^M\end{bmatrix}+b_2\right)\ldots\right)\right)+b_L\right),\\ \widehat{y}_{ui} &= \sigma\left(h^T\begin{bmatrix}\phi^{GMF}\\ \phi^{MLP}\end{bmatrix}\right).\end{aligned} \tag{15}$$

Here, ϕ^{GMF} and ϕ^{MLP} are seperate embeddings learned by GMF and MLP, respectively. $\widehat{y}_{ui}$ is the final matching score by concatenating their last hidden layers. p_u^G and p_u^M denote the user embedding for GMF and MLP parts, and similar notations of q_i^G and q_i^M are used for item embeddings. W_x, b_x, and a_x stand for the weight matrix, bias vector, and activation function respectively in the x-th layer perceptron of the MLP model. h is the edge weight of the ouput layer.

Tay *et al.* [65] propose a new neural architecture named Latent Relational Metric Learning (LRML) for collaborative ranking with implicit feedback, which learns the relationship between users and items in metric space using latent relation vectors. The vectors are generated utilizing a newly devised auxiliary memory module - Latent Relational Attentive Memory (LRAM) controlled by the user-item interactions, thus making the generated vector specific to each user-item pair.

When using the feature-based models, it is crucial to capture interactions and relationships of the features. Considering that the linearity of a single Factorization Machine (FM) can be insufficient and the complex structure of deep neural networks may be difficult to train, He and Chua [66] propose a novel model named Neural Factorization Machine (NFM) for learning higher-order and non-linear feature interactions, which couples the utility of FM and neural network in modeling second-order and higher-order feature interactions respectively. Based on a novel proposed Bi-Interaction operation, this model can learn more informative feature interactions at the lower level. To discriminate the importance of different feature interactions, Xiao *et al.* [67] present a novel model named Attentional Factorization Machine (AFM), which uses the attention mechanism to learn the weight of feature interaction. In this work, the AFM improves both the representation ability and the interpretability of a FM model.

In conclusion, the basic methods mentioned above are mainly based on matrix factorization, while the last two kinds of algorithms are deep learning-based matching models. The difference between these models is as follows. Representation-based learning focuses on learning the representations of the two items to be matched. Specifically, in the process of recommendation, model structures such as AutoEncoder, MLP, CNN or RNN can be used depending on the available information like text, image, and voice. For matching function-based learning, the entities are matched at the bottom layer. The underlying matching signal is fused with a neural network to get the final matching score. In the recommendation process, the focus of this kind of algorithm is to find ways to combine features.

C. Entity-Relation Matching

Another important matching is entity-relation matching in the knowledge graph representation. Recent years have witnessed a proliferation of knowledge graphs in many real-world applications such as semantic parsing [110], [111], information extraction [112], link prediction [113], [114], recommender systems [115], [116], question answering [117], [118] etc. Knowledge graph (KG) representation aims at transforming the symbolized components (e.g., entity and relation in a triplet) to vector, matrix, or tensor, which is easy to manipulate by computer. We categorize the algorithms into three groups: factorization-based algorithms, neural networks-based algorithms and translational distance-based algorithms.

For a given triplet $\langle h, r, t\rangle$ that represents a piece of knowledge, where h, r and t denote the head entity, relation, and tail entity, respectively, a typical representation process first projects h, t and r into a continuous vector space based on some methods (randomly or another pre-processing process such as Word2vec [119]). It then defines a matching function on this triplet to measure the similarity or plausibility between entities in the triplet. During the training process, by randomly replacing entities or relations, the final representation vectors in a "golden" triplet should achieve the max matching function compared with other "negative" training triplets. We find that from the perspective of matching, the knowledge graph representation process can be easy to understand. The reason behind this is that the essence of knowledge representation is to discover the relation among large amounts of information which is equivalent to matching.

1) Factorization-Based Algorithms: Based on the previous introduction, we can see that the basic components in knowledge graphs are entity and relation in a triplet. The matrix and tensor are two commonly used forms to represent a knowledge graph. Factorization-based models construct a distributed representation in low-dimensional space for each entity and each relation while preserving the relationship between them. Afterward, the specific tasks such as prediction can be performed by using appropriate similarity calculation formula like inner product.

Sutskever *et al.* [68] propose the Bayesian Clustered Tensor Factorization (BCTF) model which constructs the matching score according to the partition to which relation belongs. They assume entities in the same cluster should obey similar distributed representations. The matching degree of distributed representations should be determined by the value of $E_h^T W E_t$ where $W \in \mathbb{R}^{d \times d}$ is a weight matrix representing relation in a triplet, E_h and E_t are the vectors of head entity and tail entity.

RESCAL [69] model the entity and relation matching problem by a three-dimensional tensor. Each matrix slice can be factorized as a product of entity matrix and relation matrix. Unlike BCTF, RESCAL learns a unique space for entities in different domain which is independent of their occurrence in a relation. Furthermore, RESCAL can capture a more fine-grained matching between entities than BCTF.

In knowledge graph representation, we take each relation as a matching operator during the process of learning the embedding of entities and relations. However, there are multi-matching patterns in the real world. Like the matching theory in economics, the relation matching in knowledge graph can be one-to-one or many-to-many. BCTF and RESCAL use three-way models to construct the matching relations. However, the three-way model assigns each matching pattern the same capacity which can be friendly to frequent matching patterns and be problematic for rare matching patterns. The two-way model can be more feasible, in which each triplet $\langle h, r, t \rangle$ is decomposed into three binary interactions (h, r), (r, t), and (h, t). To design a more general model for multi-matching patterns in knowledge graph, Alberto *et al.* [70] propose Tatec (for Two And Three-way Embeddings Combination) that combines two-way model and three-way model for adopting different matching patterns in the knowledge graph representation. Their matching function consists of two parts: a Bigrams for modeling two-way matching and a Trigram for modeling three-way interactions. The matching function in Tatec is defined as

$$f(h, r, t) = B(h, r, t) + T(h, r, t), \tag{16}$$

where $B(h, r, t)$, $T(h, r, t)$ correspond to two-way and three-way matching, respectively.

Liu *et al.* [71] propose another factorization-based model ANALOGY aiming at representing entity and relation through analogical inference. In an analogical inference task, one can find the correlation between two systems, and the unobserved part in one system can be inferred from the corresponding known part in another system. In ANALOGY, linear maps are used to represent the matching between entity and relation. The analogical structure is represented through the commutative properties of the linear maps. In a directed graph where the nodes are entities and the edges represent relations, the directed paths that share the same starting node and end node form the compositional equivalence. In addition to that, normal matrices are used for the linear maps for their convenient properties desirable for relation modeling. In ANALOGY, to reduce the matching problem's search space, the analogy is defined by adding normality and commutativity constraints to the matching function to capture the compositional equivalence.

2) Neural Networks-Based Algorithms: As previously mentioned, matching in knowledge graph representation is defined as mapping the entities and relations into a low-dimensional vector space, thereby obtaining the similarity score to measure how close the entities are. The factorization-based models process this problem practically in a linear way. However, entities and relations maintain considerable non-linear semantic information, which may not be represented by factorization. Therefore, some methods leverage neural networks to model the nonlinearity in knowledge representation.

SME (Semantic Matching Energy) [72] transform the representation of entity and relation in knowledge graph to a semantic matching energy problem based on the energy-based learning theory [120]. According to this theory, the plausible triplets should be assigned low energies. The semantic energy matching function measures energy loss which transforms head entity to tail entity, therefore, the lower the value of the function, the more plausible the given triplet is. Triplets are first mapped to their embeddings E_h, E_r, E_t. The related-matching between entity and relation in a triplet can be formed as $g_1(E_h, E_r)$ and $g_2(E_t, E_r)$. The matching function defined in SME is:

$$f(h, r, t) = g_1(E_h, E_r)^T g_2(E_t, E_r). \tag{17}$$

Here g corresponds to a bilinear activation function defined as:

$$g(x, y) = (W \circ y^T) x^T + b, \tag{18}$$

where $W \in \mathbb{R}^{p \times d \times d}$ is a weight tensor and $b \in \mathbb{R}^p$ is a bias.

Unlike SME, NTN [73] designs a novel neural network by changing the hidden layer to a bilinear tensor which aims at capturing the interactions between entities. Furthermore, instead of initializing the representation vectors by random sampling from a noise distribution, NTN chooses the average of word vectors to initialize the input. This improvement can preserve statistical characteristics of the input data. The matching function defined in NTN is:

$$f(h, r, t) = u_r^T g(E_h^T W_r^{[1:k]} E_t + V_r [E_h, E_t]^T + b_r), \tag{19}$$

where $g = tanh$ is an activation function and $W_r[1:k] \in \mathbb{R}^{d \times d \times k}$ is a tensor. The $E_h^T W_r^{[1:k]} E_t$ is a tensor product embedding in a vector space $\mathbb{R}^k$ with ith element computed as $E_h^T W_r^{[i]} E_t$. The other parameters $V_r \in \mathbb{R}^{k \times 2d}$, $u_r \in \mathbb{R}^k$ and $b_r \in \mathbb{R}^k$ are standard form of a neural network for relation r.

MLP (Multi Layer Perceptron) [74] represents entity and relation as a single vector by leveraging CNN. The weights in the neural network preserve the interactions between entities and the activation function capture the nonlinearity in the knowledge graph. The matching function in MLP can be defined as:

$$f(h, r, t) = \sigma(u^T g(W^T [E_h, E_r, E_t])), \tag{20}$$

where $W \in \mathbb{R}^{d \times 3d}$ represents the input layer weights and $u \in \mathbb{R}^d$ represents the hidden layer weights. g is a non-linear function such as $tanh$. Compared with NTN, MLP model achieves almost the same performance but requires less parameters.

NAM (Neural Association Model) [75] uses a deep neural network to model the association between two events A_1 and A_2. This deep neural network with a multi-layer nonlinear activation function can be adopted to compute the likelihood that two events A_1 and A_2 are to be matched using conditional probability $P(A_2|A_1)$. For a triplet $\langle h, r, t \rangle$, event A_2 represents head entity

h and relation r, and A_1 is an event that shows whether the tail entity t is true or false. For a binary classification task, the corresponding activation part can be a $sigmoid$ function, and for multiple output tasks, the activation part can be a $softmax$ function. Therefore, this architecture of a deep neural network can be general for a lot of specific tasks. There are two structures of NAM, one is the traditional DNN and the other is RMNN (Relation-Modulated Neural Network), which is more suitable for modeling multi-matching tasks in the knowledge graph representation.

3) Translational Distance-Based Algorithms: Inspired by word embedding work such as Word2Vec [119], translational distance-based models adopt the matching process to a translation operator from head entity to tail entity. An example for this intuition is that two triplets $\langle king, isA, man\rangle$ and $\langle queen, isA, woman\rangle$ with the same relation should infer the equation $king - man \approx queen - woman$.

Bordes *et.al* [76] propose TransE which assumes that the translation between entities can be seen as a natural mapping process in representing them. The matching score function in TransE is defined as:

$$f(h,r,t) = \|E_h + E_r - E_t\|_{L_2}. \tag{21}$$

Here $E_h, E_r, E_t \in \mathbb{R}^d$, the distance measurement is computed by L_2 norm and the score function should be small if the relationship triplet $\langle h, r, t\rangle$ holds.

From the perspective of matching, we can find that the matching process of TransE can not deal with the reflexive matching, many-to-one and many-to-many matching. The reason behind it is that TransE ignores distributed representation of entities in different relations. To solve this, TransH [77] is proposed by projecting relation into a hyperplane space. In other words, compared with TransE which projects entity and relation in the same space, TransH maps the translation or matching operator to another space which distinguishes the different semantic roles one entity plays in different matching scenes. The head entity and tail entity are first projected into the relation-hyperplane as:

$$E_{h\perp} = E_h - w_r^T E_h w_r,\; E_{t\perp} = E_t - w_r^T E_t t w_r, \tag{22}$$

where w_r is the normal vector to the hyperplane. Here $E_{h\perp}$ and $E_{t\perp}$ should be close in the projection hyperplane if they describe a plausible fact. The score function in TransH is defined as:

$$f(h,r,t) = \|E_{h\perp} + E_r - E_{t\perp}\|_{L_2}. \tag{23}$$

Through another hyperplane, the matching process in TransH can be more fine-grained in representing the entity and relation in knowledge graph.

Although TransH constructs a hyperplane for relation, it still assumes the embeddings of entities and relations should be in the same space. However, entities and relations of knowledge graph preserve various semantic information in different scenarios such as $\langle apple, kindOf, fruit\rangle$ and $\langle apple, isA, company\rangle$. TransR [78] constructs different spaces for entity and relation. The matching process from entity space to relation space can be done by a specific projection matrix M_r, with the entity vector projections defined as follows:

$$\hat{E}_h = E_h M_r, \hat{E}_t = E_t M_r. \tag{24}$$

According to this, the matching function in TransR is defined as:

$$f(h,r,t) = \|\hat{E}_h + E_r - \hat{E}_t\|_{L_2}. \tag{25}$$

TransH and TranR focus their attention on modeling the diversity of relations in the semantic matching process which ignores the diversity of entities. In the matching theory we have introduced in other sections, the diversity of matching is related to the objects of matching. TransD [79] considers both the diversity of entity and relation. Two vectors are used to represent entity and relation in TransD, with one used to capture the semantic information and the other to construct the mapping matrix in translation.

From TransE to TransD, we find that these methods construct more fine-grained ways to model the complexity and diversity of entity and relation semantic information. However, the matching score functions defined in these methods are oversimplified by using inflexible metrics such as Euclidean distance. Such an over-simplified metric may miss a lot of information in modeling the nonlinearity with a spherical equipotential hypersurface. TransA [80] defines a matching function using elliptical hyper-surfaces to better model complex embedding topologies of complex relations. The matching function defined in TransA is:

$$f(h,r,t) = |E_h + E_r - E_t|^T W_r |E_h + E_r - E_t|, \tag{26}$$

where W_r is a relation-specific symmetric weight matrix with non-negative elements. By distorting the original equipotential hyper-surfaces to an elliptical one, TransA enlarges the differences between entities at some dimensions, which can improve embedding complex entities in the knowledge graph.

This subsection mainly focuses on representing knowledge graph by mapping entities and relations into low-dimensional vectors while capturing their semantic meanings. Factorization-based algorithms process this problem essentially in a linear way. However, entities and relations maintain considerable non-linear semantic information, which may not be represented by factorization. Therefore, some methods leverage neural networks to model the nonlinearity in knowledge representation. In translational distance based models, the matching between entities and relations is regarded as a translation problem. It can be viewed as a semantic matching task in the natural language process.

D. Image Matching

Computer vision comes to its new age due to the dramatic boom of deep learning in recent years [121]. Image matching as a basic algorithm can be applied to many research fields that require the ability to recognize and search for matching images, such as image retrieval, object tracking, face recognition, and object detection. Image matching is the process of making two images consistent in space so that the matching pixels in the two images are the same as the area to be matched. Here, we divide image matching algorithms into two types as [122]: area-based matching and feature-based matching.

1) Area-Based Algorithms: Area-based methods are also called correlation-based methods or template matching. Area-based methods merge the feature detection step with the matching part, which is different from feature-based methods. Levine *et al.* [81] incorporated an adaptive correlation window as a solution to object detection. They helped a robot to analyze the environment by representing three-dimensional objects in a scene into the depth map. Classical correlation measures were used within an adaptive window size. The area-based similarity metric can be used to define the most likely correspondence between the same sub-areas from two different views. In many applications, the dot product correlation function is used as a similarity metric, where the maximum score represents the best matching result.

Normalized Cross-Correlation(NCC) and its modifications are classical area-based methods. Gruen [82] proposed a powerful image matching technique that uses an adaptive least squares correlation. This technology can be applied to feature extraction, change detection, and line tracking of multi-spectral and multi-temporal images. Wu *et al.* [83] proposed a fast, highly accurate NCC image matching algorithm. A constructed wavelet pyramid can reduce the searching and matching times of the feature point. An NCC image matching algorithm was then proposed to obtain the coarse matching points in the matching image.

NCC algorithm is considered a time-consuming approach. To increase computing speed, the Fourier [123] method is required for the images under different environments. In these images with noise, the Fourier method outperforms correlation methods. Fourier methods exploit the Fourier representation of images in the frequency domain. A technology for quickly matching image with a number of images in a database, which extracts Fourier-Mellin phase features from images, was proposed by Ishiyama *et al.* [84] under the geometric changes of rotation and scale.

In the mutual information based registration method, the joint probability of comparable pixel intensities in the matching images is estimated. To detect and recognize the small dimensionality target, Yang *et al.* [85] proposed an image matching algorithm based on mutual information. The algorithm calculates the joint entropy of the matching image and the image to be matched. It then takes the coordinate with the maximum mutual information obtained in rough matching as the center position, and compares matching pixel by pixel to obtain the final matching score.

2) Feature-Based Algorithms: The work of image matching and image feature extraction can be traced back to 1981 [124]. A corner detector was applied for stereo matching, but the Moravec detector was time-consuming and sensitive to noise. Soon after, Harris and Stephens [86] developed a combined corner and edge detector based on the local auto-correlation function by improving the Moravec detector in 1988. The detector was composed of gradient information and eigenvalues of symmetric positive definite 2×2 matrix, and it was shown to perform with good consistency on natural imagery. Harris corner detector is sensitive to scale, which does not bring about a good matching performance for images of different sizes. Moreover, its application was limited to the stereo scene and short-range motion tracking.

Scale invariant feature transform (SIFT) [87] is one of the most widely used methods. It has good performance on matching and recognition due to its invariance to rotation, scale, and translation. Bay *et al.* [88] revealed that Speed-Up Robust Features (SURF) is an effective SIFT implementation method, which calculates the derivations of the image by applying integral images. Rublee *et al.* [89] proposed an efficient methods ORB alternative to SIFT or SURF, which is a very fast binary descriptor based on BRIEF. ORB is rotation invariant and robust to noise. Babri *et al.* [90] made a comparative study on feature-based image matching algorithms. Furthermore, they concluded that the quality of features detected by SURF is better than SIFT because SIFT cannot match a large number of features it detects. For distorted images whose angle of rotation is proportional to 90 degrees, Karami *et al.* [91] showed that ORB and SURF outperform SIFT.

In previous years, research was mainly focused on face recognition under controlled conditions, among which simple classical methods provided excellent performance. Nowadays, the focus of research is on unconstrained conditions. Deep learning technology [125] is becoming more and more popular because it provides strong robustness and can resist a large number of variations in the recognition process. In industry, face recognition as an application of image matching is generally divided into three steps: face detection, feature extraction, and feature matching. To extract high-quality features, face alignment usually comes before feature extraction. In the face matching part, two matching images are compared to obtain a similarity score, which gives the possibility that they belong to the same subject.

Facebook proposed Deepface [92] in 2014. It is the foundation work of the deep convolutional neural network in the field of face recognition. The 3D model was used in the face alignment task. Then, the deep convolutional neural network implemented multi-class classification learning for the aligned face patch, using the classic cross-entropy loss function ($Softmax$) to optimize the problem. Finally, a fixed-length face feature vector was obtained through feature embedding. Later, Google proposed FaceNet [93], which used the Triplet Loss function instead of the Softmax for optimization on a hypersphere space to make the distance of clusters closer. Recently, ResNet [94] has become the most popular choice for many target recognition tasks. The main novelty of ResNet is the introduction of a building block that uses a shortcut connection to learn the residual mapping. ResNet facilitates the flow of information across layers, therefore the cross-layer connections allow training on deeper architectures. Nevertheless, face recognition is still facing many issues, such as racial bias in biometric that has not been thoroughly studied in deep face recognition [95].

In summary, the area-based methods are used to achieve dense matching without any obvious feature points detected from the image. They are more popular in highly overlapping image matching (such as medical image registration) and narrow baseline stereo images (such as binocular stereo image matching) [23]. Though the area-based methods have lower computational complexity, they limit the size of windows in large-scale images. Feature-based image matching can effectively solve the limitation problems in large viewpoint, wide baseline, and severe non-rigid image matching. It can be used for feature detection, discriminant description, and reliable matching, usually including transformation model estimation [23]. By combining multiple methods, more accurate and reliable matching solutions can be obtained. Therefore, compound deep learning approaches would be the solution to achieve more accurate results in some applications such as real-time detections [126].

IV. Applications

In previous sections, we introduced specific models and methods of two kinds of matching problems, which could be used to address a lot of serious problems in the real world. As a matter of fact, the matching theory is first proposed and studied in economics, which has path-breaking articles that developed intuitive algorithms. With the development of interdisciplinary research, the matching theory has been applied to more and more applications, and the term "matching" has extended its concepts to other fields. In this section, we present some practical applications in both explicit and implicit matching to better understand the concepts and methods in different matching problems.

A. Applications in Explicit Matching

1) Matching in Wireless Networks: Wireless networks consist of selfish and rational agents that naturally seek their maximum benefit from the system without considering other agents. In some complex wireless networks, various agents with different characteristics solicit communications with each other, where matching theory is particularly applicable to develop suitable and effective solutions. Here, we will introduce some of the significant applications of matching theory in wireless communication.

Cognitive Radio Networks: Decentralized operation and efficient resource management are necessary for cognitive radio networks, which require licensed primary users to occupy channels that must be accessed by unlicensed secondary users. In other words, cognitive radio networks require stable solutions in matching licensed primary users and unlicensed secondary users. A number of recent works have corroborated the suitability of matching theory for cognitive radio [34], [36].

Leshem *et al.* [36] took one of the first steps in applying matching theory into cognitive radio network. In this work, the association of licensed primary users with unlicensed secondary users is described as a one-to-one matching problem. The same utility function is used to get both sides' preferences. Hence, they find the stable allocation in a time-efficient way using a modified version of the DA algorithm. Later, Naparstek *et al.* [34] extended this work from the perspective of energy efficiency.

Device-to-Device (D2D) Communications: Device-to-Device (D2D) Communications is a technology devised to overcome the ever-increasing wireless capacity crunch. By introducing D2D in cellular networks, new challenges will arise in terms of interference management and resource allocation [127]. Therefore, matching theory can be applied broadly in this area.

In [35], a form of "cheating" in the preference lists was incorporated to improve the DU's utilities. DUs can smartly change their preferences by cheating, thereby reaping more performance gains. In the final experimental results, the authors find that using such cheating strategies can simultaneously improve DU's and system utility compared with the DA algorithm.

2) Matching in Large Firms: Firms spend significant resources to hire the right employee and give different wages to workers in different occupations. Kelso and Crawford [28] explore a general many-to-one matching model of firms for any number or type of workers. Kremer [31] explores O-ring production model which indicated several stylized facts, such as positive correlation among wages of workers in different occupations within a firm. Tervio [33], and Gabaix and Landier [32] develop a matching model of firm size and CEO talent, and calibrate it using US data to analyze CEO pay. They show that the model exhibits a superstar property: small differences in talent can have a drastic impact on pay at the top.

B. Applications in Implicit Matching

1) Expertise Matching: Within the context of big scholarly data [128], expertise matching can be regarded as the process of finding the alignment between experts and queries. In other words, it is a process of finding individuals with the required knowledge and skills. Unlike information retrieval systems, expertise matching needs an expert retrieval system for facilitating knowledge exchange. Methodologically, existing methods can be divided into two classifications: probabilistic model and optimization model with multiple constraints, such as load balance, authority balance, and topic coverage [129]. Tang *et al.* [130] regards the expertise matching as an optimization problem, the objective is to assign some experts to each query by satisfying certain constraints. Additionally, to validate their algorithms' effectiveness, they apply their framework to a practical conference system in terms of assigning experts to review papers. To support the rapid exchange of knowledge in innovation clusters, Babkin *et al.* [131] propose a method of ontology-aided expertise matching based on a new methodology of ontology concepts matching. They also make several contributions to the advances of knowledge processing. Recently, Qian *et al.* [132] developed a model named weakly supervised factor graph (WeakFG) by considering two problems, how to trade off the degree between expertise and topic, and to what extent the invited users are willing to answer questions. They incorporate a number of correlations based on social identity theory into the WeakFG model, which combines expertise matching and correlations between experts. Furthermore, they design an online system to demonstrate the advantages of their proposed model.

2) Question-Answer Matching: Baidu team proposes a multi-view question-answer model [133] in 2016, which is similar to the hierarchical structure and jointly considers the word-level matching and utterance-level matching. The SMN [134] model proposed by Microsoft Research Lab-Asia (MSRA) combines the representation based method and the interaction based method and integrates the word-level interaction and the segment-level interaction. Alibaba Group has officially launched the first exclusive AI customer service robot for Taobao and Tiancat apps users, AliMe Assis.[1] It is built on a multi-turn conversation model, MT-hCNN [135] using convolutional neural networks. Zhang *et al.* [136] focus on retrieval-based response matching for multi-turn conversation and propose the DUA model. It transmits important information in each utterance by introducing self-matching attention. After matching the response with the refined utterance and sharp turns aggregation, the final matching score can be obtained. Zhou *et al.* [137] from the Baidu research team propose Deep Attention Matching Network (DAM) model to realize response selection of chat robot in the multi-turn conversation understanding. It uses a transformer encoder to get the multi-granularity representation of the text. Then, it designs two interactive ways to get self-attention-match and cross-attention-match alignment matrices.

3) Recommender Systems: Recommender systems, especially deep recommender systems, have been widely used in industry and e-commerce scenarios [138]. An efficient recommender algorithm is at the core of big businesses for advertisements, media services, and online retailers, promotes business growth, and brings many economic benefits. The following introduces some typical recommender systems at the industry level.

Google proposes DCN (Deep&Cross Network) to predict user click through rate (CTR). Taking advantage of DNN models, DCN introduces a novel cross network to learn bounded-degree feature interactions. DCN contains four critical parts, including embedding and stacking layer, cross network and deep network, and combination output layer. Feature crossing at each layer makes DCN require no manual feature engineering. Similarly, Facebook adopts DLRM (Deep Learning Recommendation Model) [139] for a personalized recommendation. DLRM combines collaborative filtering and predictive analysis. Specifically, DLRM first encodes user features as one-hot vectors, and obtains user representations by embedding lookup. The initial feature embeddings will learn the distinguishing feature representations in the process of model optimization. Finally, feature representations and their interactions are inputted in MLP, and the final click probability is predicted by the sigmoid function. Other typical algorithms for CTR, including LS-PLM [140], DeepFM [141], and NFM [142] have been applied to different advertising scenes such as Alibaba.

From the perspective of media services, personalized recommendation plays an important role in social media. For example, 80% of the content that users watch on Netflix comes from recommendations. Netflix's recommendation system is divided into three parts: offline, nearline, and online. From offline to online, the real-time performance of data increases, while the scale and processing capacity of data decreases. Several core recommendation algorithms used by Netflix include Personalized Video Ranker (PVR), Top-N Video Ranker, Trending Now, Continue Watching, and Video-Video Similarity.[2] YouTube is one of the largest and most complex industrial recommendation systems. It is designed to help more than 1 billion users discover personalized video content from a growing collection of videos. Before deep learning, YouTube's recommendation system was mostly based on user profiling and collaborative filtering [143]–[145]. The core deep learning algorithm [146] of the system consists of two neural networks, i.e., candidate set generation and deep ranking. The deep collaborative filtering model can effectively assimilate features and model their interactions, solving large scale problems, freshness, and noise in YouTube recommendation.

V. Future Trends and Challenges

In this section, we make a detailed discussion about the future trends and challenges of matching algorithms based on two key matching factors.

A. Preference List

The preference list is hidden in large-scale data with the changes and development of matching. Therefore, how to extract or infer the preference list which can reflects the real expectation rank of agents is crucial in existing matching scenarios.

1) Information Fusion for Preference List Inference: Instead of giving a strict or clear rank order of preference for other agents in matching, the preference list is hidden in various data sources. Many matching algorithms are devoted to inferring a list that can reflect the real preference of agents. In the future, various sources of data will be collected or fused in matching systems and matching scenarios with the continuous deepening of computational intelligence. Therefore, it is worth exploring how to effectively fuse these heterogeneous data to obtain the preference list. A number of works have been proposed to explore this trend in retrieval matching [147]–[149], user-item matching [150]–[152], entity-relation matching [153], [154], and image matching [155], [156]. However, extracting the decisive features from the problematic data such as data with missing values, noise, or outliers to complete the task of preference list inference poses a great challenge [157].

2) Dynamic Preference List Inference: Another problem to be solved in preference list inference is its uncertainty. On one hand, the real preference information of matching agents may be hidden in various sources of data. On the other hand, this preference list may change with time. The typical instance in user-item matching is that users may change their preferences for items after sales promotion [158]. Therefore, interactive matching systems such as interactive recommendation [159]–[161], interactive information retrieval [162], [163] have been witnessing a proliferation of attention in recent years to collect the feedback of matching agents to infer a more accurate preference list. Therefore, the challenge behind this task is to design an effective framework that can collect feedback information (e.g., incentive mechanism [164]) and feedback information analysis (e.g., implicit feedback [165]).

B. Matching Principles

The matching principle can be understood as the expected outcome of the matching problems. For example, stability is the

[1] https://consumerservice.taobao.com/online-help

[2] https://netflixtechblog.com/

basic principle when researchers design matching algorithms for men and women in marriage markets. While in implicit matching, the ultimate goal of the model is to match the most similar pairs. In a nutshell, a matching algorithm develops in accordance with application scenarios. Compared with classical matching market theory, data accessibility leads to more loose and complicated matching methods. Here, we outline three existing challenges when designing matching algorithms, namely fairness, interpretability and privacy-protection.

1) Fairness: Fairness is the inherent topic in matching. In classical matching theory, it has been proved that there is not a solution to guarantee fairness for both sides under a minimal set of axioms [166]. However, with the widespread use of AI systems in our daily life, it is important to consider fairness while designing matching algorithms, especially algorithms based on deep learning. The bias of an unfair algorithm usually comes from the heterogeneous data [167]. A matching algorithm trained on biased data may lead to unfair and inaccurate results. The fairness principle of matching algorithms will attract increasing attention in the future [168]–[170]. In addition, a clear definition of fairness and bias, as well as the dataset of unfairness samples still need to be explored in specific matching problems [167].

2) Interpretability: Interpretability is also an important consideration when designing matching algorithms, which can also be understood as the rationalizability in AI system [171]. The classical matching market theory is explainable inherently, as it is designed based on the known preference lists. However, most implicit matching algorithms based on deep learning are black-box, which are incapable of answering ‘how’ the preference lists are inferred. Therefore, the interpretability of existing matching algorithms should be explored to design more credible matching algorithms. However, the heterogeneity of data in existing matching systems poses great challenges for implicit matching algorithms when considering interpretability [172]. Some preliminary related works about interpretable matching algorithms can be found in [173]–[175].

3) Privacy-Protection: As we have discussed previously in this section, it is difficult to obtain accurate and personalized preference lists with an increasing amount of information incorporated into matching systems. For example, in POI (Point of Interest) recommendation system, the location information, contact list information, and check-in information can be accessed and collected only if the platform gets the permission of users. Otherwise, it may cause the problem of privacy information disclosure. Therefore, protection of the privacy information is an important principle that need to be considered when designing implicit matching algorithms [176]–[179]. However, in the matching systems, balancing the cost in protecting the privacy information and benefits is still a controversial topic.

VI. Conclusions

In this survey, we systematically summarized the common real-world matching problems and divided the matching problems into two categories according to the availability of the preference lists: namely explicit matching and implicit matching. In explicit matching, the matching problems are classified according to the agent requirements, namely one-to-one, many-to-one and many-to-many. In implicit matching, we mainly presented some common matching problems such as retrieval matching, user-item matching, entity-relation matching, and image matching. To better understand the concepts and methods, we introduced some practical applications for both categories of matching problems. Additionally, the future trends and challenges are also discussed according to two key matching factors. This article is expected to provide a comprehensive overview of matching problems and models suitable for the demands in various practical scenarios to address technical challenges in current and future matching problems.

Acknowledgment

The authors would like to thank Jianshuo Xu and Jiaxing Li for their help with the first draft of this paper.

References

[1] D. Gale and L. S. Shapley, “College admissions and the stability of marriage,” *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[2] L. Wang, Y. Li, J. Huang, and S. Lazebnik, “Learning two-branch neural networks for image-text matching tasks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 394–407, Feb. 2019.

[3] M. Akbarpour, S. Li, and S. O. Gharan, “Thickness and information in dynamic matching markets,” *J. Political Economy*, vol. 128, no. 3, pp. 783–815, 2020.

[4] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003.

[5] A. Berger and J. Lafferty, “Information retrieval as statistical translation,” *Proc. ACM SIGIR Forum*, vol. 51, no. 2, 2017, pp. 219–226.

[6] T. Hoessler, P. Schulz, E. A. Jorswieck, M. Simsek, and G. P. Fettweis, “Stable matching for wireless urllc in multi-cellular, multi-user systems,” *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5228–5241, Aug. 2020.

[7] Y.-K. Che, J. Kim, and F. Kojima, “Stable matching in large economies,” *Econometrica*, vol. 87, no. 1, pp. 65–110, 2019.

[8] F. Xia, Z. Chen, W. Wang, J. Li, and L. T. Yang, “Mvcwalker: Random walk-based most valuable collaborators recommendation exploiting academic factors,” *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 364–375, Sep. 2014.

[9] W. Wang, Z. Gong, J. Ren, F. Xia, Z. Lv, and W. Wei, “Venue topic model enhanced deep graph modelling for citation recommendation in scholarly big data,” *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 1, pp. 1–15, 2020.

[10] S. A. Kazmi, A. Ndikumana, A. Manzoor, W. Saad, C. S. Hong, “Distributed radio slice allocation in wireless network virtualization: Matching theory meets auctions,” *IEEE Access*, vol. 8, pp. 73494–73507, 2020.

[11] A. Ismaili, N. Hamada, Y. Zhang, T. Suzuki, and M. Yokoo, “Weighted matching markets with budget constraints,” *J. Artif. Intell. Res.*, vol. 65, pp. 393–421, 2019.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv:1810.04805*.

[13] N. Y. Asabere, F. Xia, W. Wang, J. J. Rodrigues, F. Basso, and J. Ma, “Improving smart conference participation through socially aware recommendation,” *IEEE Trans. Human Mach. Syst.*, vol. 44, no. 5, pp. 689–700, Oct. 2014.

[14] F. Xia, N. Y. Asabere, A. M. Ahmed, J. Li, and X. Kong, “Mobile multimedia recommendation in smart communities: A survey,” *IEEE Access*, vol. 1, pp. 606–624, 2013.

[15] J. Liu, J. Ren, W. Zheng, L. Chi, I. Lee, and F. Xia, “Web of scholars: A scholar knowledge graph,” in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 2153–2156.

[16] J. Liu, J. Tian, X. Kong, I. Lee, and F. Xia, “Two decades of information systems: A bibliometric review,” *Scientometrics*, vol. 118, no. 2, pp. 617–643, 2019.

[17] F. Xia, A. M. Ahmed, L. T. Yang, J. Ma, and J. J. Rodrigues, "Exploiting social relationship to enable efficient replica allocation in Ad-hoc social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3167–3176, Dec. 2014.

[18] H. D. Bedru *et al.*, "Big networks: A survey," *Comput. Sci. Rev.*, vol. 37, 2020, Art no. 100247.

[19] F. Xia, N. Y. Asabere, H. Liu, Z. Chen, and W. Wang, "Socially aware conference participant recommendation with personality traits," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2255–2266, Dec. 2017.

[20] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1202–1212, Jun. 2017.

[21] A. Abdulkadiroglu and T. Sönmez, "Matching markets: Theory and practice," *Adv. Econ. Econometrics*, vol. 1, pp. 3–47, 2013.

[22] R. Hakimov and D. Kübler, "Experiments on matching markets: A survey," *WZB Discuss. Paper, Tech. Rep*., 2019.

[23] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1–57, 2020.

[24] W. H. Gomaa *et al.*, "A survey of text similarity approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013.

[25] A. E. Roth and M. Sotomayor, "Two-sided matching," *Handbook Game Theory Economic Appl.*, vol. 1, pp. 485–541, 1992.

[26] G. S. Becker, "A theory of marriage: Part I," *J. Political Economy*, vol. 81, no. 4, pp. 813–846, 1973.

[27] T. C. Bergstrom and M. Bagnoli, "Courtship as a waiting game," *J. Political Economy*, vol. 101, no. 1, pp. 185–202, 1993.

[28] A. S. Kelso, Jr. and V. P. Crawford, "Job matching, coalition formation, and gross substitutes," *J. Econometric Soc.*, vol. 50, no. 6, pp. 1483–1504, 1982.

[29] S. Rosen, "The economics of superstars," *Amer. Economic Rev.*, vol. 71, no. 5, pp. 845–858, 1981.

[30] A. E. Roth, "Stability and polarization of interests in job matching," *J. Econometric Soc.*, vol. 52, no. 1, pp. 47–57, 1984.

[31] M. Kremer, "The o-ring theory of economic development," *Quart. J. Economics*, vol. 108, no. 3, pp. 551–575, 1993.

[32] X. Gabaix and A. Landier, "Why has CEO pay increased so much?," *The Quart. J. Economics*, vol. 123, no. 1, pp. 49–100, 2008.

[33] M. Tervio, "The difference that CEOs make: An assignment model approach," *Amer. Economics Rev.*, vol. 98, no. 3, pp. 642–68, 2008.

[34] O. Naparstek, S. Zafaruddin, A. Leshem, and E. Jorswieck, "Distributed energy efficient channel allocation," 2014, *arXiv:1401.1671*.

[35] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Cheating in matching of device to device pairs in cellular networks," in *Proc. IEEE Glob. Commun. Conf.*, 2014, pp. 4910–4915.

[36] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 82–95, Jan. 2012.

[37] J. Ramos *et al.* "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instructional Conf. Mach. Learn.*, vol. 242. Piscataway, NJ, 2003, pp. 133–142.

[38] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[39] S. Robertson *et al.*, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[40] S. T. Dumais, "Latent semantic analysis," *Annu. Rev. Inf. Sci. Technol.*, vol. 38, no. 1, pp. 188–230, 2004.

[41] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Conf. Uncertainty Artif. Intell*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[42] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.

[43] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage*., 2013, pp. 2333–2338.

[44] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage*., 2014, pp. 101–110.

[45] H. Palangi *et al.*, "Semantic modelling with long-short-term memory for information retrieval," 2014, *arXiv:1412.6629*.

[46] J. Choi, K. M. Yoo, and S. Lee, "Learning to compose task-specific tree structures," in *Proc. AAAI Conf. Artif. Intell*. AAAI Press, 2018, pp. 5094–5101.

[47] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.

[48] Y. Bengio *et al.*, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[49] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 2793–2799.

[50] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng, "Match-SRNN: Modeling the recursive matching structure with spatial RNN," in *Proc. 25th Int. Joint Conf. Artif. Intell*. AAAI Press, 2016, pp. 2922–2928.

[51] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[52] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2249–2255.

[53] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *Proc. 13th AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 2835–2841.

[54] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "aNMM: Ranking short answer texts with attention-based neural matching model," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 287–296.

[55] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6586–6593.

[56] B. Mitra, F. Diaz, and N. Craswell, "Learning to match using local and distributed representations of text for web search," in *Proc. 26th Int. World Wide Web Conf. Steering Committee*, 2017, pp. 1291–1299.

[57] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, 2009.

[58] T. Tran, K. Lee, Y. Liao, and D. Lee, "Regularizing matrix factorization with user and item embeddings for recommendation," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 687–696.

[59] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 111–112.

[60] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, 2016, pp. 153–162.

[61] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2015, pp. 77–118.

[62] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proc. Int. Joint Conf. Artif. Intell.* 2017, pp. 3203–3209.

[63] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 335–344.

[64] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web. Steering Committee*, 2017, pp. 173–182.

[65] Y. Tay, L. Anh Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in *Proc. World Wide Web Conf. Steering Committee*, 2018, pp. 729–739.

[66] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 355–364.

[67] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proc. 26th Int. Joint Conf. Artifi. Intell.*, 2017, pp. 3119–3125.

[68] I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov, "Modelling relational data using Bayesian clustered tensor factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1821–1828.

[69] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," *Proc. Int. Conf. Mach. Learn.*, vol. 11, 2011, pp. 809–816.

[70] A. García-Durán, A. Bordes, and N. Usunier, "Effective blending of two and three-way interactions for modeling multi-relational data," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*. Berlin, Germany: Springer, 2014, pp. 434–449.

[71] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70. JMLR. org, 2017, pp. 2168–2178.

[72] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, 2014.

[73] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 926–934.

[74] X. Dong *et al.*, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 601–610.

[75] Q. Liu *et al.*, "Probabilistic reasoning via deep learning: Neural association models," 2016, *arXiv:1603.07704*.

[76] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[77] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, vol. 28, no. 1, pp. 1112–1119.

[78] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 2181–2187.

[79] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 687–696.

[80] H. Xiao, M. Huang, Y. Hao, and X. Zhu, "TransA: An adaptive approach for knowledge graph embedding," 2015, *arXiv:1509.05490*.

[81] M. D. Levine, D. A. O'Handley, and G. M. Yagi, "Computer determination of depth maps," *Comput. Graph. Image Process.*, vol. 2, no. 2, pp. 131–150, 1973.

[82] A. Gruen, "Adaptive least squares correlation: A powerful image matching technique," *South Afr. J. Photogrammetry, Remote Sens. Cartogr.*, vol. 14, no. 3, pp. 175–187, 1985.

[83] P. Wu, W. Li, and W. Song, "Fast, accurate normalized cross-correlation image matching," *J. Intell. Fuzzy Syst.*, vol. 37, no. 4, pp. 4431–4436, 2019.

[84] R. Ishiyama, T. Takahashi, K. Makino, and Y. Kudo, "Fast image matching based on fourier-mellin phase correlation for tag-less identification of mass-produced parts," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2018, pp. 380–384.

[85] Q. Yang and Q. Wei, "An image matching algorithm based on mutual information for small dimensionality target," in *Proc. 14th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discov.*, 2018, pp. 1166–1171.

[86] C. G. Harris *et al.*, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[87] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[88] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[89] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[90] U. M. Babri, M. Tanvir, and K. Khurshid, "Feature based correspondence: A comparative study on image matching algorithms," *Int. J. Adv. Comput. Sci. Appl*, vol. 7, no. 3, pp. 206–210, 2016.

[91] E. Karami, S. Prasad, and M. Shehata, "Image matching using sift, surf, brief and orb: Performance comparison for distorted images," 2017, *arXiv:1710.02726*.

[92] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[93] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[94] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[95] M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, "Racial faces in the wild: Reducing racial bias by information maximization adaptation network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 692–702.

[96] N. Shimada, N. Yamazaki, and Y. Takano, "Multi-objective optimization models for many-to-one matching problems," *J. Inf. Process.*, vol. 28, pp. 406–412, 2020.

[97] X. Kong *et al.*, "Many-to-many collaborator recommendation based on matching markets theory," in *Proc. IEEE Intl Conf. Dependable, Autonomic Secure Comput., Intl Conf. Pervasive Intell. Comput., Intl Conf. Cloud Big Data Computing, Intl Conf. Cyber Sci. Technol. Congr.*, 2019, pp. 109–114.

[98] M. A. O. Sotomayor, *Two-Sided Matching: Study in Game-Theoretic Modeling and Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[99] C. Bin, H. Feng, W. Kai, X. Jinting, Z. Liwei, and F. Jing, "Uroad: An efficient method for large-scale many to many ride sharing matching," *J. Comput. Res. Develop.*, vol. 56, no. 4, pp. 866–883, 2019.

[100] M. Kurino, "Credibility, efficiency, and stability: A theory of dynamic matching markets," *The Japanese Econ. Rev.*, vol. 71, no. 1, pp. 135–165, 2020.

[101] S. Qian, B. Wang, S. Li, Y. Sun, Y. Yu, and J. Wang, "Many-to-many matching for social-aware minimized redundancy caching in d2d-enabled cellular networks," *Comput. Netw.,*, vol. 175, 2020, Art no. 107249.

[102] M. T. Islam, A.-E. M. Taha, S. Akl, and M. Abu-Elkheir, "A stable matching algorithm for resource allocation for underlaying device-to-device communications," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[103] T. Bendlin and H. Hosseini, "Partners in crime: Manipulating the deferred acceptance algorithm through an accomplice," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9917–9918.

[104] H. Xu and B. Li, "Seen as stable marriages," in *Proc. IEEE INFOCOM*, 2011, pp. 586–590.

[105] S. Bayat, Y. Li, L. Song, and Z. Han, "Matching theory: Applications in wireless communications," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 103–122, Nov. 2016.

[106] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," 2020, *arXiv:2004.03705*.

[107] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. IJCNN 2000. Neural Comput.: New Challenges Perspectives for New Millennium*, vol. 3, 2000, pp. 189–194.

[108] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, 2019.

[109] B. T. Betru, C. A. Onana, and B. Batchakui, "Deep learning methods on recommender system: A survey of state-of-the-art," *Int. J. Comput. Appl.*, vol. 162, no. 10, pp. 17–22, 2017.

[110] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1533–1544.

[111] A. Celikyilmaz, D. Hakkanitur, P. Pasupat, Sarikaya, and Ruhi, "Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems," *Genre*, 2010.

[112] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.*, 2011, pp. 541–550.

[113] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, "Random walks: A review of algorithms and applications," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 2, pp. 95–107, Apr. 2020.

[114] L. Wang, J. Ren, B. Xu, J. Li, W. Luo, and F. Xia, "Model: Motif-based deep feature learning for link prediction," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 503–516, Apr. 2020.

[115] H. Liu *et al.*, "Trucom: Exploiting domain-specific trust networks for multicategory item recommendation," *IEEE Syst. J.*, vol. 11, no. 1, pp. 295–304, Mar. 2017.

[116] F. Xia, H. Liu, I. Lee, and L. Cao, "Scientific article recommendation: Exploiting common author relations and historical preferences," *IEEE Trans. Big Data*, vol. 2, no. 2, pp. 101–112, Jun. 2016.

[117] A. Bordes, J. Weston, and N. Usunier, "Open question answering with weakly supervised embedding models," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*. Berlin, Germany: Springer, 2014, pp. 165–180.

[118] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in *Proc. Conf. Empir. Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 615–620.

[119] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[120] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predict. Struct. data*, vol. 1, 2006.

[121] Z. Zhou, Q. M. J. Wu, S. Wan, W. Sun, and X. Sun, "Integrating SIFT and CNN feature matching for partial-duplicate image detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 5, pp. 593–604, Oct. 2020.

[122] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, 2003.

[123] B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Trans. Image Process.*, vol. 5, no. 8, pp. 1266–1271, Aug. 1996.

[124] H. P. Moravec, "Rover visual obstacle avoidance.*" Int. Joint Conf. Artif. Intell.*, vol. 81, 1981, pp. 785–790.

[125] G. Guo and N. Zhang, "A survey on deep learning based face recognition," *Comput. Vis. Image Understanding*, vol. 189, 2019, Art no. 102805.

[126] J. Joglekar and S. S. Gedam, "Area based image matching methods: A survey," *Int. J. Emerg. Technol. Adv. Eng*, vol. 2, no. 1, pp. 130–136, 2012.

[127] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

[128] F. Xia, W. Wang, T. M. Bekele, and H. Liu, "Big scholarly data: A survey," *IEEE Trans. Big Data*, vol. 3, no. 1, pp. 18–35, Mar. 2017.

[129] F. Xia, A. M. Ahmed, L. T. Yang, and Z. Luo, "Community-based event dissemination with optimal load balancing," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 1857–1869, Jul. 2015.

[130] W. Tang, J. Tang, and C. Tan, "Expertise matching via constraint-based optimization," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1, 2010, pp. 34–41.

[131] E. Babkin, N. Karpov, and T. Babkina, "A method of ontology-aided expertise matching for facilitating knowledge exchange," in *Proc. 2nd Int. Workshop Ontologies Inf. Syst.*, 2014, p. 43.

[132] Y. Qian, J. Tang, and K. Wu, "Weakly learning to match experts in online community," in *Proc. 27th Int. Joint Conf. Artifi. Intell.*, 2018, pp. 3841–3847.

[133] X. Zhou *et al.*, "Multi-view response selection for human-computer conversation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 372–381.

[134] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li, "Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots," in *Proc. 55th Ann. Meet. Associat. Comput. Linguist.*, 2017, vol. 1, pp. 496–505.

[135] M. Qiu *et al.*, "Transfer learning for context-aware question matching in information-seeking conversations in e-commerce," in *Proc. 56th Annu. Meet. Associat. Comput. Linguist.*, 2018, vol. 2, pp. 208–213.

[136] Z. Zhang, J. Li, P. Zhu, H. Zhao, and G. Liu, "Modeling multi-turn conversation with deep utterance aggregation," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3740–3752.

[137] X. Zhou *et al.*, "Multi-turn response selection for chatbots with deep attention matching network," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics* (Volume 1: Long Papers), 2018, pp. 1118–1127.

[138] J. Liu *et al.*, "Artificial intelligence in the 21st century," *IEEE Access*, vol. 6, pp. 34 403–34 421, 2018.

[139] M. Naumov *et al.*, "Deep learning recommendation model for personalization and recommendation systems," 2019, *arXiv:1906.00091*.

[140] K. Gai, X. Zhu, H. Li, K. Liu, and Z. Wang, "Learning piece-wise linear models from large scale data for ad click prediction," 2017, *arXiv:1704.05194*.

[141] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for ctr prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1725–1731.

[142] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 355–364.

[143] S. Baluja *et al.*, "Video suggestion and discovery for youtube: Taking random walks through the view graph," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 895–904.

[144] J. Davidson *et al.*, "The youtube video recommendation system," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 293–296.

[145] M. Bendersky, L. Garcia-Pueyo, J. Harmsen, V. Josifovski, and D. Lepikhin, "Up next: Retrieval methods for large scale related video suggestion," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 1769–1778.

[146] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proc. ACM Conf. Recommender Syst.*, 2016, pp. 191–198.

[147] T. Li, J. Zhang, K. Bao, Y. Liang, Y. Li, and Y. Zheng, "Autost: Efficient neural architecture search for spatio-temporal prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 794–802.

[148] J. Huang, H. Wang, M. Fan, A. Zhuo, and Y. Li, "Personalized prefix embedding for poi auto-completion in the search engine of baidu maps," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2677–2685.

[149] X. Niu *et al.*, "A dual heterogeneous graph attention network to improve long-tail performance for shop search in e-commerce," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 3405–3415.

[150] G. Guo, B. Chen, X. Zhang, Z. Liu, Z. Dong, and X. He, "Leveraging title-abstract attentive semantics for paper recommendation," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, 2020, pp. 67–74.

[151] R. Xie, Z. Qiu, J. Rao, Y. Liu, B. Zhang, and L. Lin, "Internal and contextual attention network for cold-start multi-channel matching in recommendation," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 2732–2738.

[152] Y. Xu, L. Zhu, Z. Cheng, J. Li, and J. Sun, "Multi-feature discrete collaborative filtering for fast cold-start recommendation," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 270–278.

[153] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.

[154] N. Park, A. Kan, X. L. Dong, T. Zhao, and C. Faloutsos, "Multiimport: Inferring node importance in a knowledge graph from multiple input signals," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 503–512.

[155] H. V. Vo *et al.*, "Unsupervised image matching and object discovery as optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8287–8296.

[156] M. Manzo, "Graph-based image matching for indoor localization," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 3, pp. 785–804, 2019.

[157] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1172–1180.

[158] X. Wang, P. Yin, Y. Gao, and L. Guo, "A dynamic recommender system with fused time and location factors," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 23, no. 1, pp. 25–33, 2019.

[159] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenburg, and J. Leskovec, "Hierarchical temporal convolutional networks for dynamic recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 2236–2246.

[160] W. Lei *et al.*, "Interactive path reasoning on graph for conversational recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2073–2083.

[161] H. Chen *et al.*, "Large-scale interactive recommendation with tree-structured policy gradient," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3312–3320.

[162] I. Ullah and S. Khusro, "On the search behaviour of users in the context of interactive social book search," *Behav. Inf. Technol.*, vol. 39, no. 4, pp. 443–462, 2020.

[163] Q. V. Liao, W. Geyer, M. Muller, and Y. Khazaen, "Conversational interfaces for information search," in *Understanding Improving Information Search*. Berlin, Germany: Springer, 2020, pp. 267–287.

[164] V. Weber, J. Argo, and S. Moore, "The impact of feedback on recommendations: What happens when others dislike the recommendation," *ACR North Amer. Adv.*, 2019.

[165] Y. Liu *et al.*, "Diversified interactive recommendation with implicit feedback," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4932–4939.

[166] F. Masarani and S. S. Gokturk, "On the existence of fair matching algorithms," *Theory Decision*, vol. 26, no. 3, pp. 305–322, 1989.

[167] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," 2019, *arXiv:1908.09635*.

[168] G. K. Patro, A. Chakraborty, N. Ganguly, and K. Gummadi, "Incremental fairness in two-sided market platforms: On smoothly updating recommendations," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 181–188.

[169] H. Abdollahpouri, M. Mansoury, R. Burke, and B. Mobasher, "The connection between popularity bias, calibration, and fairness in recommendation," in *Proc. 14th ACM Conf. Recommender Syst.*, 2020, pp. 726–731.

[170] M. D. Ekstrand, R. Burke, and F. Diaz, "Fairness and discrimination in retrieval and recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 1403–1404.

[171] Y.-S. Ong and A. Gupta, “Air 5: Five pillars of artificial intelligence research,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 5, pp. 411–415, Oct. 2019.

[172] Y. Zhang and X. Chen, “Explainable recommendation: A survey and new perspectives,” *Found. Trends Inf. Retrieval*, vol. 14, no. 1, 2020, pp. 1–101.

[173] J. Singh and A. Anand, “Exs: Explainable search using local model agnostic interpretability,” in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 770–773.

[174] T. Chen, H. Yin, G. Ye, Z. Huang, Y. Wang, and M. Wang, “Try this instead: Personalized and interpretable substitute recommendation,” in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 891–900.

[175] Z. Fu *et al.*, “Fairness-aware explainable recommendation over knowledge graphs,” in *Proc. 43rd Int. ACM SIGIR Conf. Research Develop. Inf. Retrieval*, 2020, pp. 69–78

[176] Y. He, G. Meng, K. Chen, X. Hu, and J. He, “Towards security threats of deep learning systems: A survey” *IEEE Trans Softw. Eng.*, 2020, doi: 10.1109/TSE.2020.3034721.

[177] S. Zimmerman, A. Thorpe, C. Fox, and U. Kruschwitz, “Investigating the interplay between searchers’ privacy concerns and their search behavior,” in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 953–956.

[178] S. Zimmerman, A. Thorpe, J. Chamberlain, and U. Kruschwitz, “Towards search strategies for better privacy and information,” in *Proc. Conf. Hum. Inf. Interact. Retrieval*, 2020, pp. 124–134.

[179] L. Qi, X. Zhang, S. Li, S. Wan, Y. Wen, and W. Gong, “Spatial-temporal data-driven service recommendation with privacy-preservation,” *Inf. Sci.*, vol. 515, pp. 91–102, 2020.

Jing Ren received the bachelor’s degree from Huaqiao University, Quanzhou, China, in 2018 and the master’s degree from the Dalian University of Technology, Dalian, China, in 2020. She is currently working toward the Ph.D. degree with the School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, VIC, Australia. Her research interests include data science, social computing, and graph learning.

Feng Xia (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently an Associate Professor and a Discipline Leader with the School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, VIC, Australia. He has authored or coauthored two books and more than 300 scientific papers in international journals and conferences. His research interests include data science, social computing, and systems engineering. He is a Senior Member of theACM.

Xiangtai Chen received the B.Sc. degree in educational technology from Northwest Minzu University, Lanzhou, China. He is currently working toward the master’s degree in software engineering with the Dalian University of Technology, Dalian, China. His research interests include big scholarly data, data mining, and analysis of complex networks.

Jiaying Liu received the B.Sc. degree in software engineering from the Dalian University of Technology, Dalian, China, in 2016. She is currently working toward the Ph.D. degree with the School of Software, Dalian University of Technology. Her research interests include data science, big scholarly data, and social network analysis.

Mingliang Hou received the B.Sc. degree from Dezhou University, Dezhou, China and the M.Sc. degree from Shandong University, Shandong, China. He is currently working toward the Ph.D. degree in software engineering with the Dalian University of Technology, Dalian, China. His research interests include network science, data science, and urban computing.

Ahsan Shehzad was born in 1990. He received the B.S. degree in computer science systems from Bahauddin Zakariya University, Multan, Pakistan, in 2015 and the M.S. degree in computer science from Air University Islamabad, Islamabad, Pakistan, in 2018. He is currently working toward the Ph.D. degree in software engineering with the School of Software, Dalian University of Technology, Dalian, China. His research interests include digital image and video analysis, object re-identification, machine learning, and data science.

Nargiz Sultanova received the bachelor’s degree in applied mathematics from Baku State University, Baku, Azerbaijan, the master’s degree in mathematics, and the Ph.D. degree in optimization from the University of Ballarat (currently Federation University Australia), Ballarat, VIC, Australia. She is currently a Lecturer of mathematics with the School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, VIC, Australia. Her main research focuses on nonsmooth optimization and its various applications. She is a Member of AustMS and its ANZIAM division.

Xiangjie Kong (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently a Professor with the college of computer science and technology, Zhejiang University of Technology, China. He has authored or coauthored more than 100 scientific papers in international journals and conferences. His research interests include network science, data science, and computational social science. He is a Senior Member of the CCF and a Member of ACM.