Machine Intelligence Research
www.mi-research.net

19(2), April 2022, 89-114
DOI: 10.1007/s11633-022-1323-6

Knowledge Mining: A Cross-disciplinary Survey

Yong Rui Vicente Ivan Sanchez Carmona Mohsen Pourvali Yun Xing
Wei-Wen Yi Hui-Bin Ruan Yu Zhang

Lenovo Research, Beijing 100094, China

Abstract: Knowledge mining is a widely active research area across disciplines such as natural language processing (NLP), data mining (DM), and machine learning (ML). The overall objective of extracting knowledge from data source is to create a structured representation that allows researchers to better understand such data and operate upon it to build applications. Each mentioned discipline has come up with an ample body of research, proposing different methods that can be applied to different data types. A significant number of surveys have been carried out to summarize research works in each discipline. However, no survey has presented a cross-disciplinary review where traits from different fields were exposed to further stimulate research ideas and to try to build bridges among these fields. In this work, we present such a survey.

Keywords: Knowledge mining, knowledge extraction, information extraction, association rule, interpretability.

Citation: Y. Rui, V. I. S. Carmona, M. Pourvali, Y. Xing, W. W. Yi, H. B. Ruan, Y. Zhang. Knowledge mining: a cross-disciplinary survey. *Machine Intelligence Research*, vol.19, no.2, pp.89–114, 2022. http://doi.org/10.1007/s11633-022-1323-6

1 Introduction

Automatic extraction of knowledge from diverse sources of data is a challenging task across different fields. For example, in natural language processing (NLP), research on the extraction of structured knowledge bases from natural language text has received much attention due to its applications (e.g., automatically building these knowledge structures from biomedical text to understand drug-drug interactions). In data mining (DM), a wide area of research has focused on mining rules from structured databases that can help people discover novel associations between items or features and make decisions in diverse contexts such as business or education. Furthermore, in the field of machine learning (ML), plenty of effort has been advocated towards extracting knowledge, mainly in the form of logic rules, from both machine learning system′s predictions and parameters in order to build an interpretable representation that helps to explain the system′s decisions (the so-called interpretability problem); a scenario highly sought in medicine, for example.

Extracting or mining knowledge from data (be it unstructured, structured, or behavioral data) is an open problem that has been tackled across different research fields. This wide scenario has not only led to different definitions and ways to represent the construct of knowledge (and consequently, to define the task of knowledge mining), but it has also resulted in diverse research perspectives, which seem to use different methodologies to extract knowledge and different metrics to evaluate the consistency of the knowledge extracted. For example, Fayyad et al.[1] operationalize knowledge as a pattern extracted from data that surpasses some interestingness threshold (a domain-dependent metric characteristic in the data mining field) such as a rule or a linear component in a regression model characterizing a subset of features in a database; in this way, Fayyad et al.[1] define the term of knowledge discovery in databases (KDD), an interchangeable concept with knowledge mining, as the process of mining databases to discover useful (or interesting) knowledge in the form of patterns. On the other hand, in the NLP field, a knowledge base is usually represented as a tensor structure where each entry usually corresponds to a probabilistic assignment of the belief of a fact. (For example, in [2], a knowledge base is operationalized as a matrix where rows represent different pairs of entities, $E=((e_i,\ e_j),\cdots,(\mathrm{e}_m,e_n))$, and columns represent different relation types, $R=(r_1,\cdots r_n)$, which can be applied to entity pairs; thus, a cell in the matrix can be interpreted as a confidence score of a fact: confidence of $(r_i\,(e_i,e_j))$.) Thus, knowledge in this case is operationalized as a particular piece of concrete information (factual knowledge), and the task of extracting such information is that of mining such entity pairs and their relations from text. Finally, in the field of machine learning, the problem of knowledge mining has been motivated by the problem of trying to understand and validate ML systems which due to their complexity are not

Review
Manuscript received September 15, 2021; accepted January 29, 2022; published online March 10, 2022
Recommended by Associate Editor Zhi-Yuan Liu
Colored figures are available in the online version at https://link.springer.com/journal/11633

easy to be inspected manually. Similarly, the choice of the representation of knowledge has been constrained to be understandable by humans, where a widely common and accepted representation in this area are logic rules[3].

From this brief overview of knowledge mining across fields, we can observe the diversity of objectives and constructs and the wide scenario we claimed at the beginning, which leads us to the questions: How is knowledge mining characterized across research fields? What are their proposed approaches and shared traits? And how can we consolidate them? We note that while there are already several in-depth surveys in the literature of each field showing the methods and algorithms to extract knowledge, to the best of our knowledge, there is no survey that jointly traverses these research areas to answer the above questions. Furthermore, the importance of mining knowledge has permeated different fields and has also impacted the industry. Therefore, we believe that a cross-disciplinary literature review, in a landscape-oriented approach, that encompasses all these varying degrees of freedom underlying the problem of mining knowledge from data is on the call.

In this paper, rather than surveying a plethora of methods and previous works across these three research areas, we intend to overview the nuances, and attached idiosyncrasies, of the approaches taken to extract knowledge from a target data source. Hence, this paper advocates for an additive overview of the problem of extracting knowledge across the fields of natural language processing, data mining and machine learning to show their key objectives, methods, and evaluations, and how some previous works have made links among these areas for the task of knowledge mining.

The final aim of this paper is to stimulate and provoke new ideas and research agendas among researchers from the different disciplines so that new bridges among the areas surveyed can emerge to further advance in the task of knowledge mining. Following this approach, we avoid providing a single definition of knowledge and knowledge mining, and rather present how these constructs have been embraced across fields. Thus, we depart from a common starting point across fields. We fix the choice of knowledge representation to that of logic, or logic-like formulas, which is a representation highly used across these fields. Based on this knowledge representation, in Sections 2−4, we will walk through the different goals and key approaches of each field, in a problem-oriented perspective, to gain a refined insight into how knowledge mining is embodied and what traits we find in these research areas. Finally, in Section 5, we will summarize these approaches and their shared traits, and pinpoint some examples of previous efforts in forging a bridge among the three fields while closing with a proposed future research direction. We believe this paper will contribute to creating future research directions for the task of knowledge mining that encompass the three, so far unlinked, research areas of NLP, DM, and ML.

2 Knowledge extraction from natural language text

Extracting relational knowledge from text written in a natural language such as English means automatically identifying spans of text that correspond to named entities, classifying them into their corresponding entity types, and classifying the type of relation holding between these entities (if any). The outcome of this process is information stated as logic predicates[1] [5, 6]. For example, from a newspaper text, we aim to automatically detect the people and organizations the text is referring to and save that information as logic formulas — grounded predicates — such as person(Joe Biden), organization(United Nations), or country(USA). Another piece of information we aim to extract from the text is possible relations between these entities, such as president_of(Joe Biden, USA), which are also represented in the form of logic predicates. A collection of these relational facts structured into a database is known as a knowledge base[2] (KB). The task of recovering the targeted factual knowledge from text is commonly known as information extraction (IE) in the NLP community.

In what follows, we provide preliminaries of state-of-the-art methods and models in NLP in Section 2.1. We introduce the most common learning approaches for information extraction, namely supervised learning (classification and sequence labeling), distant supervised learning, and unsupervised learning in Section 2.2. Then, we provide an account of the two IE problems that have received much attention in the NLP community, namely named entity recognition in Section 2.3 and relation extraction in Section 2.4, as well as the methods to evaluate how well an NLP system performs at any of these tasks in Section 2.5. Finally, we review some current challenges in NLP related to the problem of IE in Section 2.6.

2.1 Preliminaries

In this section, we briefly introduce notation and basic notions of the most popular machine learning models used for the task of information extraction. Then, in Sec-

[1] First-order logic (FOL) allows us to represent facts (objects and their relations) through predicates[4]. In this way, a predicate accounts for a relation type, and a predicate symbol refers to the name of a relation. The arity of a predicate indicates the number of arguments it can receive. Representing factual knowledge using FOL not only aligns with the traditional way of representing knowledge in artificial intelligence[4], but it also satisfies some useful characteristics of a representation for natural languages such as verifiability, avoidance of unambiguity, inference, and expressiveness among others[5].

[2] Usually, knowledge bases contain several thousands of semantic relation types.

tion 2.2, we will briefly introduce the learning approaches for these models.

The current literature on IE is dominated by neural approaches, i.e., neural networks (NNs). An early example of NN is the feed-forward model, or multi-layer perceptron, classifying an input into a class label. This input, in the form of a vector, is processed through layers of hidden neurons (usually one or two layers) where a non-linear function, such as the tanh or sigmoid function, is applied at each neuron after a weighted linear combination of the input signals to the neuron. At the last layer of the NN, a probability distribution over the possible classes is computed.

Recent models use feed-forward models as a layer on top of another type of neural network, a representation learning model or neural encoder; this type of NN works as an encoder of input information rather than as a classifier leaving the job of classification to a feed-forward model. More concretely, the neural encoder models receive a vector representation of a portion of text as input (usually at the word-level or sentence-level) and, through layers of non-linear hidden neurons, encode (transform) the input into a hidden vector during the training process of the NN (or during the test process for some NN models). This vector may contain information of the context of the input, e.g., lexical information from the words surrounding the current input word. We exemplify this idea with the two NN models most used for IE, namely the long-short term memory (LSTM)[7] and bidirectional encoder representations from transformers (BERT)[8] models.

An LSTM is a type of recurrent neural network (RNN) that is able to process an input sequence one token at a time with the capability of remembering (and forgetting) long-range dependencies between the tokens in the sequence. On the other hand, BERT is a recent language model built from transformer networks, which pretrains vector representations of words from a large corpus. The input of BERT can be a single sentence, or two sentences (separated by a [SEP] mark), where the first token is usually a classification mark ([CLS]). During pretraining, it randomly masks some words, and the goal is to predict the original word. BERT is jointly pre-trained through two supervised tasks: masked language model and next sentence prediction. Different from other embedding models, BERT representations can be fine-tuned on downstream NLP tasks, such as text classification, by adding an additional output layer; thus, the resulting vector representations are called contextualized.

2.2 Learning approaches

Supervised learning. In this learning approach, a machine learning model is trained to learn patterns from a set of labeled data, i.e., a model is trained to correlate a set of input features to a target output by tuning its parameters until it learns patterns relevant for this correlation. The type of output varies according to the task. In a classification task, the output is a label symbol indicating a class in a domain. For example, given the named entities Eiffel Tower and Paris, a classifier can predict the relation type (class label) that best applies to those two entities, namely located_in. In a sequence labeling task, a classifier's output is a sequence of labels given a sequence of words as input (one label for each word). An example of this can be seen in Fig. 1. In this way, a trained ML model can classify relations or named entities from texts that were not part of the training procedure (a test dataset).

The	University	of	Oxford	is	located	in	the	UK.
O	B-ORG	I-ORG	E-ORG	O	O	O	O	B-LOC

Fig. 1 Example of the named entity recognition task. For a given sentence in a natural language, the objective is to label each of the tokens according to the type of entity they correspond to (in the case where a token does not correspond to any entity, the special label O is given). In order to do the labeling, a special tagging scheme is used (in this figure, the BIOE tagging is used). In this example, there are two named entities, namely University of Oxford and UK. The first entity is composed of three tokens; thus, the first token receives the label B-ORG, which means that it is the beginning of the entity type organization since universities are considered as such; the second and third tokens receive the I-ORG and E-ORG labels signaling the interior and end parts of the entity. The second entity, a single token, receives one label indicating its type, a location.

Distant supervised learning. This is a type of semi-supervised learning where the supervision signal does not come from a manually annotated dataset but from an external source such as a knowledge base: Instances from the knowledge base are automatically aligned to portions of unannotated text from a corpus. In this way, the text is automatically labeled according to the target task and the type of knowledge in the KB.

Unsupervised learning. In this approach, there is no need for an annotated text to train a classifier. Instead, an algorithm can automatically find patterns or estimate probability distributions based on the unlabeled data. The most common unsupervised algorithms are clustering algorithms, which group instances into a set of clusters based on their feature similarity according to a similarity function.

2.3 Named entity recognition

Named entity recognition (NER) is a two-fold task: First, we need to find spans of text in a document that are mentions of particular entities; some entities may span over one or more tokens[3], such as the entity John Smith, which refers to one person; after that, we need to classify each of the entities found into their type (John Smith is a person). The result is a logic-based representa-

[3] A token can be a word or a punctuation mark.

tion of the concrete information about entities that we found in a piece of text (e.g., person(John Smith)). The types of entities to be predicted for each text span candidate depend on the dataset used to train and test the classifiers. One of the most widely used datasets in the literature is CoNLL2003[9] which consists of 1 393 news articles split into training (946), development (216), and test (231) sets. The news articles in this dataset are split into sentences, where each token is annotated according to the type of entity it represents. There are four types of target entities in this data, namely person, organization, location, and miscellaneous names. Another NER dataset frequently used is OntoNotes 5.0[10].

Overall, NER is a task mainly delimited by two dimensions: The way the input information (a sentence) is encoded and the type of approach used to process the input (the machine learning model and its training regime). Next, we will structure our survey of previous works in NER according to these two dimensions.

Encoding an input sequence. In most of the early works in NER, the input sequence was encoded through hand-crafted features accounting for lexical and syntactical information of the tokens, which could help an ML system distinguish between a target named entity and an ordinary token. For example, some previous works[11] encoded each token in a sentence into a set of features (usually binary features) denoting whether the token was capitalized or not (which can help to detect proper names), the shape of the token (number of characters and position of capital letters if any), and prefixes and suffixes of varying length (which can help, for example, detecting English verbs in past tense that end in ed); furthermore, lexical resources such as gazetteers and databases of people names, organization names or geographical entities have been used to complement the hand-crafted features where each token, or sequences of n tokens (an n-gram), in the input sequence, can be compared against the entries of these lexical resources also to obtain a binary feature (the current n-gram is in the database or not) which can help in the decision of labeling a token[11, 12].

Recent approaches to NER have used so-called word embeddings as input features. A word embedding is a distributed vector representation of a word; thus, each word in a vocabulary corresponds to a unique vector of continuous numbers where the dimensionality is manually chosen. This representation can be automatically learned either extrinsically, in an unsupervised way on a large corpus of texts, or intrinsically, as a part of the training process of the neural encoder used for a downstream task such as the NER task. Furthermore, pre-trained word embeddings (those learned extrinsically) can be used as the input for neural encoders (as explained in Section 2.1). These features, although they can be complemented with hand-crafted features, usually replace previous features from the literature. Nevertheless, previous works such as [13] have used lexicons to inject some lexical information into word embeddings. Overall, many of the latest approaches to NER use neural encoders with word embeddings as input features.

Supervised based approaches. Along our second dimension characterizing the task of NER, the main approach for building NLP systems that uncover named entities from text is supervised learning, where methods range from classification to sequence labeling (with a significant focus on the latter), as explained in Section 2.1. Nevertheless, some of the earliest approaches focused on building hand-crafted rules to identify and predict the entities on text. For example, Appelt et al.[14] used trigger words and finite-state machines to recognize lexical and syntactic patterns. However, rule-based approaches have the problem of low recovery due to their inflexibility in uncovering named entities whose features lie outside the enclosure of the rules. Despite their acceptable precision performance, their overall score is usually low compared to supervised approaches. (Nevertheless, domain-dependent rule-based NER models can still be found as applications[15].)

Classifiers, such as support vector machines (SVMs)[16] or feed-forward neural networks[12], were preferred over rule-based models due to their ability to generalize to new cases, which were relatively different from those in the training data, where input sentences were usually encoded using hand-crafted features and lexical resources. However, this supervised approach encountered some problems. First, feature hand-crafting implies laborious annotator work, which may be either unscalable to large corpora or expensive due to the huge amount of human work required. Second, some lexical resources can contain entities that can be ambiguous; e.g., according to [11], the word China can be extracted from a lexicon as either a location, as a string that forms part of an organization, and even as a person's name. A final problem is how these classifiers predict a label for a given token. Labeling one word is done independently of any surrounding words; nevertheless, the label assignment of a word at time $t{-}1$ can help the classifier to predict the best label at time t (for example, once a classifier has predicted label B-ORG for token w_{i-1}, it is unreasonable to predict the same label for token w_i). Even though fixing the last problem, using the classifier as a sequence labeling model, reduces to providing the classifier with features from surrounding tokens to the token to be labeled, hand-crafted features remain an open problem since, as explained before, it may be expensive to obtain them. The resulting feature vectors will be sparse (most of the features will have value $f_i = 0$ since only a few of these are found in a given sentence), which may hinder the training process of the classifier.

Mainstream approaches to NER use sequence labeling models, where neural encoders have been widely preferred. Suitable models to encode an input sequence and to decode a sequence of labels are conditional random fields (CRFs)[17], a type of discriminative probabilistic model, and recurrent neural networks (RNNs) (mainly neural encoders). While early works used CRFs solely for the NER task[18], recent approaches have used a combination of both CRFs and RNNs[19–21] since it has been seen that this combination yields significantly better performance. For example, Ma and Hovy[20] used two of the most common types of RNNs, namely LSTMs[4] and Bi-LSTMs (Bidirectional LSTMs). In that work, each token from the input sequence was encoded using the concatenation of two representations, namely word embeddings and character embeddings[5]. These representations served as input to an LSTM, which encoded them via its hidden units. In this way, a token t_i will be represented with a new hidden vector representation, $\overrightarrow{h}_i$, from the LSTM, where information from the left tokens (t_0 to t_{i-1}) is passed and encoded in the representation of token t_i. In the case of a Bi-LSTM, signals from the right context of the current token are encoded in another hidden vector representation $\overleftarrow{h}_i$ of token t_i; then, both learned representations (left and right) are concatenated to serve as the final vector representation of the current token. This representation is then passed to a CRF layer which uses it to predict the label for the token. (We note that in the latest works, only learned representations are passed to the CRF layer; however, a combination of these representations with hand-crafted features is possible.) Thus, the CRF layer computes the posterior probability of a sequence of labels y given the input sequence, namely $P\left(y|X\right)$.

The type of CRF used is a linear chain, which means that in order to compute the posterior probability across output sequences, this model aggregates local feature functions, which can use the information from the current label predicted y_i for token t_i, the previous label predicted y_{i-1} and the input sequence. In addition, to obtain the best possible sequence, a dynamic programming algorithm is used, Viterbi algorithm being the most popular:

$$\hat{Y} = \arg\max_{y} \, p\left(y|X\right). \tag{1}$$

Parameters of both the Bi-LSTM and the CRF layer are trained via stochastic gradient descent. As mentioned before, word vector representations can either be learned from a random initialization or be fine-tuned if using pre-trained word embeddings.

We note that the latest approaches to NER have used contextualized embeddings. This type of vector representation (at either the character or word level) of a lexical unit is fine-tuned with information from context units in the same input sequence which means that the vector representation of a unit depends on its surrounding units. Notable contextualized embedding models for NER are BERT (Section 2.1), the contextual string embeddings model[23], and the pooled contextualized embeddings model[24], which are currently the state-of-the-art with an $F1$ score of over 0.93 points. Similar to the case of a Bi-LSTM, these embedding models can use a CRF layer at the top to jointly infer labels of named entities.

Furthermore, recent approaches aim to handle noisy-labeled entities. For example, Liu et al.[25] propose a confidence estimation method with calibration for noisy-labeled named entity recognition. Furthermore, they apply local and global independence assumptions on an LSTM-CRF model and further integrate a self-training framework, which brings strong performance gains in both general multi-lingual noisy settings and distant supervision settings.

2.4 Relation extraction

Relation extraction (RE) is the process of extracting relations from an unstructured (e.g., text) or semi-structured (e.g., HTML table) source. For example, the task of RE in a text could be learning that a person is born in a particular city, a piece of relational knowledge represented as a grounded logic predicate (of arity $n = 2$) as in Fig. 2. A wide variety of methods have been applied for learning to recover this information from text: From unsupervised methods, such as clustering words appearing between target entities, to supervised, distantly supervised methods (such as convolutional neural network-based encoders[26]), and open relation extraction methods that may fall under either of the three previous approaches.

Popular datasets to train and evaluate RE systems are the New York Times corpus (NYT)[27] which contains around 1.8 million articles from the period 1987–2007; the TAC Relation Extraction Dataset (TACRED)[28], which targets relation types from the TAC KBP task[29] and consists of almost 120 000 instances; the Automatic Content Extraction (ACE) dataset[30, 31] where the latest version comprises 10 000 documents across different genres including news and weblogs; and the SemEval 2010 Task 8 dataset[32] where relation types correspond to semantic relations such as cause_effect or product_producer. In what follows, we will survey key works of relation extraction.

Supervised based approaches. Extracting relations from sentences through a learning-based framework over hand-labeled examples is done following the classification approach presented in Section 2.2. Given features from a sentence (including the target entity types), a classifier will predict a relation label.

We categorize this approach into two main methods,

[4] Probably, the earliest work for NER using an LSTM is that of [22].

[5] Character embeddings can be pre-trained on a corpus and then fine-tuned on the target dataset via a convolutional neural network.

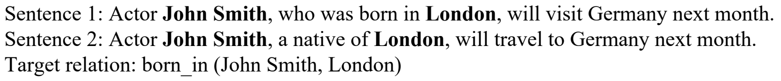
Sentence 1: Actor **John Smith**, who was born in **London**, will visit Germany next month.
Sentence 2: Actor **John Smith**, a native of **London**, will travel to Germany next month.
Target relation: born_in (John Smith, London)

Fig. 2 Example of the task of relation extraction. Both sentences encode the same relation between the entities John Smith and London, namely that this person was born in London. Despite the difference in the textual patterns, an RE system has to predict for both instances the label of the true target relation type, namely born_in.

namely non-neural and neural. The non-neural method includes traditional relation extraction approaches. Here an RE system extracts instances of the target relations from a corpus for which there are already given names and labeled examples of the relations[33]. Within the non-neural method, we also find modern approaches which use lexical features like those used for training NER systems (see Section 2.3), part-of-speech (POS) tags (such as noun, adjective, or adverb) for each token, and dependency parse trees (a syntactic representation of a sentence) to design a learning framework to classify relation types between pairs of entity mentions in a corpus. However, one of the shortcomings of leveraging syntactic features such as dependency trees is that the accuracy of the RE system heavily depends on the accuracy of the dependency tree parser. A variety of machine learning and probabilistic models are used for the learning framework, such as kernel-based approaches like support vector machines[34−36] with results on an early version of the ACE dataset slightly above the $F1$=0.55 points, and models using conditional random fields[33] which extract a large fraction of relations by relying on only a small set of POS tags patterns in the English language. In addition, there are methods like [37] that provide a joint framework to extract both named entities and relation types at the same time through a linear chair CRF (which predicts the entities) and a maximum entropy model (to extract the relation between such entities) with an $F1$ score of 0.521 points on a latter version of the ACE dataset; however, in fact, this method suffers from high computational complexity.

Recent works in RE mainly tend to apply neural networks for extracting relations. A variety of neural-based RE methods exist; e.g., they integrate information from the input sequence and its corresponding dependency tree using Bi-LSTM encoders[38, 39]. Alternatively, other methods consider the interaction of not only pairs of entity mentions with their relations but also those between relation types which have common entity mentions, as in [40], where a Bi-LSTM sentence encoder and graph convolutional networks are learned to encode pair-wise word features and both linear and dependency structures to enhance relation extraction, where results reach up to 0.619 $F1$-points on the NYT corpus. In other approach, the RE task is mapped to another task like Question Answering[41] to extract both entities and relations by asking the NLP system questions like Who is mentioned in the text? In this approach, $F1$-scores slightly surpass the 0.60 points on the latest version of the ACE corpus.

Latest approaches to RE exploit pre-trained neural encoder models (including BERT and Sci-BERT[42]). For example, Zheng et al.[43] decompose the RE task into three subtasks: relation judgement, entity extraction, and subject-object alignment and propose an end-to-end framework (PRGC), which leads to a performance gain against a variety of baselines. On the other hand, Lai et al.[44] leverage pre-trained language models to encode entities which are acronyms, specialized terms, and abbreviations. On the other hand, Wang and Lu[45] design two distinct encoders (table-sequence encoders) to capture information about the entity name and the relation information, respectively, where the two encoders help each other in learning the representation; this approach addresses the issue that a single encoder may not be enough to capture all the relevant information when two different tasks do not share the same space.

A shortcoming of supervised approaches is that producing a labeled training dataset is always expensive and thus limited in quality[46]. In this section, we survey works where there is no need for manually annotated datasets to build an RE system.

Distant supervised approaches. This approach, as proposed by [46], holds the assumption that a relation r_i between two named entities (e_i, e_j) in a knowledge base is likely to express the relation between those entities whenever they appear in a sentence, i.e., the fact $r_i(e_i, e_j)$ is assumed to hold true. Based on this assumption, given a corpus C to be used for creating a training dataset and a knowledge base (KB) including a fact $r_i(e_i, e_j)$, for every appearance of the mentions of the named entities (e_i, e_j) together in the corpus C, the relation r_i is assigned to these entities as their relation[6]. After that, the same techniques used in the supervised approaches can be used to train a classifier. The big-scale training data generated makes distant supervision an attractive option for extracting relations not only at the sentence-level but also at the document-level[47].

However, distant supervision through a KB might produce incorrect labeling (noisy data) or noisy patterns in the resulting training set, since it is a strong assumption[26]. A relation between two entities in an already existing knowledge base that was either heuristically generated from texts or generated from manual annotation (such as crowdsourcing) may not convey all the topics that those entities may share in their co-occurrences. As noted in [48], when using Freebase as the KB and the

[6] Negative instances can be generated by pairing entities that have no relation in the knowledge base, assigning them the label no-relation, and extracting features from sentences where these two entities co-occur.

NYT corpus as the target text, the distant supervision assumption is violated in 38% of the cases for the relation nationality in a sample of 100 sentences containing entity pairs corresponding to this relation type.

The problem of noisy data, which always accompanies training data in distant-supervised approaches, decreases the performance of this method; to this end, selecting valid sentences may be a solution for that. Multi-instance learning[49] is a way of dealing with such a problem, in which the training data consists of many labeled bags, each bag containing many unlabeled instances. A bag, in this case, is the set of sentences where a target entity pair (e_i, e_j) occurs, and the label of a bag is the target relation r_i observed in a KB[50]. For example, Riedel et al.[48] proposed to predict whether relation mention candidates, in sentences where the entity pair (e_i, e_j) occurs, encode the target relation type, r_i, which is observed in the knowledge base applied to the target entity pair $r_i(e_i, e_j)$. The underlying assumption is that at least one sentence in the corpus will express the target relation. Thus, it is assumed that some bags of instances contain at least one positive instance. This method allowed [48] to reduce prediction error by 31%, going from a *Precision* score of 87% when using the standard distant supervision approach to a score of 91% on the NYT dataset using Freebase as the KB. Zheng et al.[26] also treated distant supervised RE as a multi-instance problem and designed an objective function at the bag-level to deal with the resulting wrong labels in training data while using piecewise convolutional neural networks to automatically extract features from the sentences. Recent works like [50, 51] have taken further advantage of NNs to provide sentence-level attention models which can learn weights for sentences in a bag and thus select more than one valid sentence. In addition, using sentences from an external knowledge base (e.g., Wikipedia) that describe the entities can also provide better entity representations for these attention models[50].

Other recent works use attention models at both levels, inter-bag and intra-bag, to cope with noise present in bags and their instances[52]. On a further level, Xiao et al.[47] used pre-trained models to denoise the document-level distant supervision data.

In a related line of work, Wang et al.[53] took advantage of the knowledge graph embedding model TransE[54] and encoded head (h) and tail (t) entities of a relation triple in a knowledge graph to generate (t−h) as the relation (r) which is derived from the translation law ($h + r \approx t$) in KG embedding models[54−56]. Indeed, this line of work distantly supervises the learning process through (t−h) instead of using a hard relation label (r), and then trains a sentence encoder by the margin loss between (t−h) and a sentence embedding.

Unsupervised based approaches. Unsupervised relation extraction aims at automatically detecting the underlying semantic structure linking entities in a large text corpus without manually-labeled data and existing knowledge bases. Thus, opposed to supervised or distant supervised approaches, this approach does not require any lexical database or manual annotation and can be used for detecting new relation types.

There have been some promising solutions over the past decades. The first approach[57, 58] relies on a clustering algorithm and the assumption that named entity pairs from the same cluster share similar context words between each co-occurrence and thus represent a relation type. More concretely, Hasegawa et al.[57] used a trained NER system to detect named entities in the NYT corpus. Then, specific entity-pair types were defined, such as person-organization. After that, entity pair mentions under those entity-pair types were searched for in the corpus to extract their contexts (the sequence of tokens in-between the two named entities). For example, the NER system would label the named entity pair (Albert Einstein, Princeton University) as person-organization; then, target sentences containing this entity pair would be searched for, such as "Albert Einstein taught at Princeton University", "Albert Einstein, who had a teaching position at Princeton University, ⋯", or "Albert Einstein worked in Princeton University between the years ⋯". The contexts between this entity pair across sentences are then aggregated in a context vector using a bag-of-words strategy where each word is weighted by its TF-IDF (term frequency − inverse document frequency). This process is carried out for other named entity pairs under the same domain of person-organization, e.g., for (Stephen Hawking, University of Cambridge). Once context vectors are obtained, a hierarchical clustering algorithm, namely complete linkage, is applied to obtain groups of named entities where cosine similarity is used as the similarity function. Finally, the most frequent context words in a cluster are taken as representative of the topic of that cluster which is then used as a label for the relation type. A manual evaluation of the NYT dataset shows the effectiveness of this approach in two domains: person-gpe ($F1$=0.82) and company-company ($F1$=0.77). On the other hand, Rosenfeld and Feldman[58] proposed a single-linkage hierarchical clustering algorithm with a novel threshold selection technique, which outperforms other clustering algorithms.

Other clustering techniques, such as generative approaches[59, 60], have also been proposed. For example, Yao et al.[59] used three generative models (similar to topic models) for modeling tuples of entity mention pairs and their syntactic dependency path between them. *Recall* scores using Freebase as the source of ground truth on which to align and compare the resulting clusters vary from 46.9% to 74.2% according to the target relation type: author_of and film_director, respectively. On the other hand, Yao et al.[60] proposed a sense-LDA model in order to overcome sense ambiguation issues. Moreover, with the aid of various knowledge sources and an effective multi-membership clustering algorithm (MMClustering), an ensemble method[61] was developed for tackling polysemy and synonymy problems, which are widely used frequently on the web.

More recently, while Marcheggiani and Titov[62] proposed a discrete-state variational autoencoder (VAE), which uses entity prediction (reconstruction) as training signals to train a relation predictor, Tran et al.[63] proved that relation types can be induced by using named entities. In addition, the ability of language models has also been explored for extracting text representations. Inspired by state-of-the-art architectures (specifically, BERT), as well as Harris′ distributional hypothesis on relations, Soares et al.[64] built task-agnostic semantic relation representations from the entity-linked text (entities in a corpus that have been linked to identifiers) by masking a pair of named entities in a sentence with the objective of learning the vector representation of the relation between such pair, across sentences, which leads the language model to recover the observed entity pair; this approach has been proved useful not only for the task of relation extraction (classifiers tuned on relation representations obtained $F1$-scores of 0.895 and 0.715 for the datasets SemEval 2010 and TACRED, respectively) but also for other closely related tasks such as exemplar-based relation extraction.

Open relation extraction. Efforts carried out in open RE can be divided into two parts, namely, open information extraction (Open IE) and relation discovery[65]. Open IE aims to generate a structured representation of information from plain text in the form of relation tuples; for example, given the sentence "Born in Tupelo, Elvis Presley was an American singer", an Open IE system tries to extract (Elvis Presly, born_in, Tupelo). An Open IE system is usually domain-independent and does not rely on a pre-defined ontology schema, and typically the relation′s name is just the text linking two arguments. Accounting for early approaches in the literature, Yates et al.[66] proposed the first Open IE system by using a self-supervised learning approach; Fader et al.[67] leveraged POS tag patterns; Del Corro and Gemulla[68] decomposed a sentence into clauses, and Stanovsky et al.[69] created the first annotated corpus by an automatic translation from the Question-Answer Meaning Representation dataset and developed an Open IE system using a Bi-LSTM with a BIO tagging scheme. More recently, Ro et al.[70] included two classifiers for predicate and arguments; they use hidden states of a BERT model to extract predicates, and then the concatenation of predicate average, BERT hidden sequence, and position embedding are used as inputs for multi-head attention blocks for argument extraction. Wang et al.[71] proposed a text-to-triple translation framework that includes generating and ranking steps; it uses Beam search over BERT attention score to generate relevant triples and then rank the generated results using a contrastive pre-trained model. On the other hand, relation discovery aims at discovering unseen relation types from unsupervised data; e.g., [72] is a recent work in the literature that casts the task of relation discovery as a clustering task.

2.5 Evaluation

The most common metric to evaluate the performance of NER systems is $F1$, which is defined as the harmonic mean of two metrics, namely *Precision* (P) and *Recall* (R):

$$F1 = \frac{2 \times P \times R}{P + R} \tag{2}$$

$$P = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{3}$$

$$R = \frac{\text{true positives}}{\text{true positives+false negatives}}. \tag{4}$$

On the one hand, *Precision* measures the ability of a system to get a correct prediction by computing the ratio of correctly predicted positive labels and all the labels predicted as positive (including false positives). On the other hand, *Recall* measures the ability to recover correct labels by computing how many correct positive labels the system recovered from all the gold labels.

These metrics target the evaluation of positive instances, different from the *Accuracy* metric which weights the importance of both positive and negative instances the same. Thus, $F1$ is mainly used for imbalanced datasets where the number of negative instances greatly surpasses that of positive instances. In addition, this metric is used when there is a manually annotated test set of instances with gold labels on which to compare the predictions of a system.

We note that the evaluation of an NER system is not at the word-level, but rather at the entity-level[5]; i.e., $F1$ scores show the precision and recall abilities of the NER system under evaluation to detect full named entities rather than to label single tokens.

In the case of RE systems, the $F1$ metric is used whenever there exists a manual annotation of a corpus (the supervised scenario). In the case of distantly supervised approaches, there are two options. The first one is to manually check if every positive prediction is indeed a true positive (for example, if the system predicts that the relation $r_i(e_i, e_j)$ holds as true, then an annotator would corroborate this fact), which is done on a sample of predictions. The second option is to pick a set of relation types from a knowledge base and hold out half of the instances (for every target relation type) from the training set to check if the RE system correctly extracts them at test time (similarly, a portion of documents from the corpus are not used at training time so they can be used for testing); in this case, *Precision* at different *Recall* levels[5] can be computed[7]:

$$P = \frac{\#\text{ of correctly predicted relations}}{\text{total }\#\text{ of predictions in a sample}}. \tag{5}$$

In the case of unsupervised RE, different approaches for evaluation can be taken. For example, a manual evaluation of the resulting clusters using an annotated dataset can lead to using the $F1$ metric. Other evaluation approaches are 1) to compute *Recall* scores from a KB, 2) to use a gold clustering model and compute the Jaccard coefficient to measure the similarity of the resulting clusters against the golden model, or 3) to rely on a KB as the ground truth on which to align and compare the resulting clusters.

2.6 Current challenges

Traditional NER datasets such as CoNLL 2003 or OntoNotes are mainly used to train and test NER systems′ abilities to detect and label flat entities, i.e. entities which do not overlap with, or are nested within, other entities. For example, in the sentence “the Bank of England closed its doors this morning”, the entity England, a country, is nested within the organization entity Bank of England. This scenario is not considered in the NER task described in Section 2.3, making it a new challenge in the NLP community. The recent approach of [73] tackles this problem inspired by a computer vision method used to detect nested objects in images. First, text span candidates of named entities are generated from the input sequence, filtered to keep only those that are more likely to contain entities. Second, the boundaries of the remaining text spans are adjusted to better locate the target entities. Finally, a classifier labels each of the entity candidates. This work used the datasets ACE 2004, ACE 2005, KBP17 and GENIA, on which state-of-the-art results are obtained where the lowest score, $F1$=0.805, is on the GENIA dataset.

Challenges in RE are varied and have been documented in [65]. More concretely, Han et al.[65] provide a review on RE and analyze two key challenges, namely learning from text or names and datasets towards special interests, while targeting more complex RE scenarios. They also propose further required efforts in RE, such as the need for high-quality training data, performing more efficient learning, extracting inter-sentence and inter-documents relations, transferring from pre-defined based RE frameworks to automatically extracting undefined relations in open domains.

A wider open challenge in the NLP field is the creation and leverage of language models that can better aid in the task of information extraction. Pre-trained dynamic models such as BERT have been proven to be successful for knowledge extraction (as we saw in Section 2.4). Primarily, compared with static embeddings such as Word2Vec[74] and GloVe[75], dynamic models aim to capture contextualized semantic information as dynamic embeddings, which provides an effective solution to the problem of polysemy. Moreover, deep pre-trained networks can capture higher-level information like long-term dependencies, anaphora, and negation, which is crucial for enhancing performance on a series of knowledge extraction downstream tasks. In addition, traditional knowledge induction systems only find rules within a knowledge base, while recent autoregressive pre-trained models such as GPT[76] and T5[77] bring rich, expressive ability via generative ways[78].

3 Knowledge mining from transactional databases

We saw in Section 2 that information extraction aims to find, or recover, relational knowledge from text such as named entities and different types of relations between such entities. Automatically recovering this information saves us effort and time while helping us build a structured representation (a knowledge base) on which to operate for other purposes like industry applications. However, often, our objective is to discover, or mine, novel patterns (patterns difficult to spot at first glance by manually inspecting the data) rather than recovering explicit knowledge from a source of data.

A popular problem in the data mining field is to mine association rules from databases — the so-called association rule mining problem — being the market basket analysis a well-studied example of this problem. In the market basket analysis scenario, we aim to mine novel purchasing patterns from a transactional database; these patterns take the form of association rules[8]. A classic example in the literature of a novel pattern is the rule diapers → beer which tells us that people who bought diapers tended to very frequently buy beer as well. Even though at first glance it is difficult to think of a strong reason for this purchasing pattern, let alone to come up with this pattern by pure intuition, methods from data mining allow us to not only discover this type of surprising patterns but also to have confidence that this is a

[7] *Recall*, as shown in (4) in Section 2.5, cannot be computed since we do not know all the instances from which relation labels are to be recovered. In order to alleviate this problem, different *Recall* levels are proposed for increasingly bigger samples where predictions from the system are firstly ranked from higher to lower confidence scores.

[8] Association rules extracted from transactional data are a type of Boolean association rules where an item either appears or does not appear in the rule. In turn, we can see these types of rules as FOL-like rules. For example, the association rule diapers → beer can be written as $\text{buys}\,(\text{X}, \text{diapers}) \rightarrow \text{buys}\,(\text{X}, \text{beer})$, as noted in [79].

strong and interesting pattern and is not due to chance[9].

Mining association rules from a transactional database (see Fig. 3 for an example) can be framed as a two-step task: Finding frequent itemsets and generating association rules[81, 82]. An itemset refers to a set of items (products) from a transaction (the basket of products purchased by a person); thus, finding frequent itemsets is the task of extracting sets of items that appear at least a certain number of times across different transactions in a database[80]. Based on the itemsets extracted, we aim to generate association rules of the form $A \rightarrow B$ where we can have more than one item in both the antecedent (A) and the consequent (B) of the rule as long as both sets of items are disjoint. However, not all candidate rules that we could generate will necessarily be interesting or useful for us. Thus, methods for pruning and evaluating candidate rules are necessary, which we show in Section 3.3.

Transaction ID	Items purchased
1	{Eggs, Water, Chicken}
2	{Water, Soda, Diapers, Food_for_baby}
3	{Chicken, Water, Beer, Diapers}
4	{Diapers, Food_for_baby, Soda, Beer, Milk, Ham}
5	{Milk, Eggs, Bread, Ham, Beer}
6	{ Beer, Chicken}
7	{ Food_for_baby, Diapers, Milk, Bread, Beer}
8	{Eggs, Beer, Soda, Diapers, Food_for_baby}

Fig. 3 An example of a transactional database. Each row represents a transaction.

The main challenge in mining both frequent itemsets and association rules lies in the exponential complexity of the search[82], and thus in the capacity of the main memory of a computer to perform the corresponding computations since typical transactional databases tend to be extremely large in the number of transactions. In what follows, we survey some methods from the data mining field to efficiently mine the knowledge expected.

3.1 Frequent itemset generation

Frequent itemset generation is the problem of finding all the sets of items — itemsets — that occur more frequently than a given count threshold in a transactional database. Then, in a posterior step, we can extract association rules with high confidence from the database. We refer to these frequent sets of items as frequent itemsets, to their frequency of occurrence in the database as their support count and the count threshold as the minimum support.

The main challenge of this problem is the computational complexity involved. In a database containing k items, a brute-force search in the space of the items can generate $2^k - 1$ frequent-itemset candidates[82]. Furthermore, to check which candidates are frequent, each candidate needs to be searched for across all transactions (every time a candidate is found in a transaction, its support increases by one), an operation with exponential complexity. Numerous techniques have been proposed in order to improve the efficiency of frequent itemset generation, mainly from three perspectives: Reducing the number of frequent-itemset candidates, reducing the number of comparisons of each candidate against a database to obtain its support count, and reducing the number of transactions[82]. In what follows, we survey some of the main approaches to the problem of frequent itemset generation.

Apriori. The Apriori algorithm[83] is an efficient way to reduce the number of candidate itemsets, which is based on the principle that any subset of a frequent itemset must also be frequent. Apriori applies a level-wise algorithm that traverses the transactional database searching for frequent itemsets, from 1-itemsets (itemsets consisting of one item) to the frequent maximum-itemsets; i.e., it uses the frequent itemsets L_{k-1} generated at a previous level ($level = k - 1$) as the seed for the actual level ($level = k$) candidate generation, namely candidate itemsets C_k. In order to obtain this set of candidates, the algorithm joins the itemset L_{k-1} with itself across all itemsets which share $k - 2$ items[79]. Thus, from each join operation of two frequent itemsets l_i and l_j (both itemsets in L_{k-1}), the size of the resulting candidate itemset $c_i \in C_k$ will be bigger by one item than each of the itemsets joined by appending the items shared with those not shared; for example, assume the frequent 3-itemsets $l_1 = \{i_1, i_3, i_5\}$ and $l_2 = \{i_1, i_3, i_7\}$; after joining them, we obtain the candidate itemset $c_1 = \{i_1,\ i_3,\ i_5,\ i_7\}$. After generating the candidate k-itemsets, Apriori employs a support-based pruning approach known as the generation-and-test strategy to efficiently eliminate invalid candidate itemsets without counting the actual support of these itemsets; from the previous example, all the k−1-itemsets which are a subset of the candidate c_1 are compared against the set of frequent itemsets L_{k-1}, namely $\{i_1, i_3, i_5\}$, $\{i_1, i_3, i_7\}$, $\{i_1, i_5, i_7\}$, $\{i_3, i_5, i_7\}$; if any of these itemsets is not part of L_{k-1}, then, candidate c_1 is discarded because, due to the Apriori principle, a k-itemset to be frequent requires all its k−1-itemsets to be frequent. After that, the candidate k-itemsets not pruned in the previous step need to have computed their support count to check if it is above the minimum support; in order to do so, the algorithm needs to traverse the database to count the frequency of occurrence of each candidate. Candidates with a support count surpassing the threshold are inserted into the list of frequent k-itemsets, namely L_k. This process iterates again, taking the itemsets as seeds in L_k to obtain k+1-itemset candidates. At the end of the

[9] As a note, a plausible explanation behind the association rule diapers → beer seems to be due to a change of activities of people who used to frequent bars but cannot do so anymore because they have now the activity of parenting. Thus, these people purchase now the target product in supermarkets rather than in a bar[80].

process, we obtain a list of frequent k-itemsets for $k \geq 2$, which will be used to mine association rules.

FP-growth. The FP-growth algorithm[84] is proposed to further eliminate the number of candidate itemsets and reduce the number of comparisons by implementing a novel frequent pattern tree (FP-tree) structure and an efficient FP-tree-based mining method. An FP-tree is a condensed prefix-tree data structure that stores a compressed representation of the data (transactions); it is constructed by traversing all the candidate itemsets, then mapping every itemset onto a path in the FP-tree in which each node represents an item and also holds the support of the item; this method only needs one pass through the data and is much more efficient when there exists much overlap between different itemsets. Furthermore, for mining frequent itemsets, it proposes an FP-tree-based pattern fragment growth mining method and a partitioning-based, divide-and-conquer method, which significantly reduces the search space.

Eclat. Different from Apriori-like or FP-tree-based methods which rely on horizontal data layout, the Eclat (equivalence class transformation) algorithm[85] relies on a vertical database layout where each item is represented by a tidset (set of transaction ID); the benefit of this format is that the size of a tidset represents its support count, hence infrequent itemsets can be discarded in a single data pass. The main challenge that Eclat faces is the intersection from current tid-lists to next level tid-lists; approaches such as bottom-up search, top-down search or hybrid search are employed with vertical to horizontal transformation on-the-fly.

HUI-Miner. Previous approaches generate a considerable number of invalid candidates itemsets which causes a high computational complexity. The HUI-Miner[86], which stands for high utility itemset miner, introduces a novel concept called utility; it is used to estimate the importance of an itemset and other concepts derived from utility to provide heuristic information during pruning. Besides a transaction table, a novel structure named utility-list is proposed in order to store utility relevant information. An initial version of a utility-list can be constructed by only scanning the database twice. With the help of the initial utility-lists and a novel pruning method, the HUI-Miner can efficiently mine all high-utility itemsets.

PrePost. The PrePost algorithm[87] proposes a novel data structure called N-list for itemset representation. An N-list is achieved via constructing a PPC-tree where each node comprises five types of information: item-name, count, children-list, pre-order and post-order; thus, the N-list is another form to represent information stored in a PPC-tree. On the other hand, the efficiency of the mining method remains a challenge; the PrePost adopts the single path property of N-list as pruning strategy while the next version PrePost$^+$ uses superset equivalence; nevertheless, both of them have been proved to be faster than traditional mining methods including the FP-growth algorithm.

DPT. Opposite to FP-growth-like approaches, which take a considerable amount of time on constructing several conditional prefix trees, the DPT (dynamic prefix tree) algorithm[88] only needs one prefix tree as well as to introduce a novel concept named post-conditional database. When DPT traverses each node in the prefix tree in a depth-first way, the post-conditional database of a node (itemset) can be constructed simultaneously, which is the key technique that ensures an efficiency improvement.

3.2 Association rule mining

A highly desirable type of pattern to be discovered by methods in data mining is a set of association rules as introduced in [83]. Considering a table in which each row includes a transaction, and each transaction consists of a set of items, an association rule is an expression $(X \Rightarrow Y)$, where X and Y are sets of items[89]. Returning to the example from the beginning of Section 3, consider a large dataset of transactions in a supermarket, where a considerable number of customers who buy diapers also buy beer; the following rule (hidden relationship) can be extracted:

$$\{\text{diapers}\} \rightarrow \{\text{beer}\} \tag{6}$$

where, indeed, it provides a suggestion to the supermarket owner that there might be a relationship between the sale of diapers and beer. There have been plenty of efforts in this field of study, especially since massive amounts of transaction data have been collected and stored in databases due to the use of computers and automated data collection tools[90]. In this section, we aim to provide the roots and fundamental principles of mining association rules.

The seminal and influential work of [83], namely the Apriori algorithm, generates association rules on top of the frequent itemsets discovered in the first step of the algorithm, as discussed in Section 3.1. Apriori considers all possible association rules formed by combining all possible subsets of a frequent itemset. For example, assuming the frequent itemset $\{i_1, i_3, i_7\}$, the possible rules generated are: $\{i_1,\ i_3\} \rightarrow i_7$, $\{i_1,\ i_7\} \rightarrow i_3$, $\{i_3,\ i_7\} \rightarrow i_1$, $i_1 \rightarrow \{i_3,\ i_7\}$, $i_3 \rightarrow \{i_1,\ i_7\}$, $i_7 \rightarrow \{i_1,\ i_3\}$. Nevertheless, not all candidate rules will be considered a strong pattern; thus, a confidence threshold is manually proposed to filter weak rules (similar to filtering infrequent itemsets). Each rule candidate $A \rightarrow B$ is thus associated with a confidence value computed as the conditional probability $p\,(A|B)$, which is obtained through counting items A and B in the transactional database[79].

Given the simplicity and efficacy of the Apriori algorithm in generating association rules, most of the subsequent studies in association rule mining adopted an Apriori-like approach[91].

For example, Srikant and Agrawal[89] extracted association rules between items at any level of a given taxonomy (is-a hierarchy). Instead of finding rules at a single level, Han and Fu[90] proposed a multiple-level association rule mining method. On the other hand, Agrawal and Srikant[92] introduced sequential patterns mining, where each sequence is a list of transactions ordered by transaction-time. Srikant and Agrawal[93] later generalized the sequential patterns mining by adding some time constraints, a user-specified time window, and allowing sequential patterns to include items across all levels of a given user-defined taxonomy (is-a hierarchy).

We note that many of the early studies in association rule extraction such as [94] focused on Boolean attributes to discover interesting associations between items, such as:

$$(\text{Pizza} = \text{Yes})\ \text{AND}\ (\text{Coke} = \text{Yes})\ \Rightarrow\ (\text{Potato} = \text{Yes})\,. \tag{7}$$

However, in addition to Boolean attributes, databases in the real world may have numeric attributes. To account for this problem, Fukuda et al.[95] proposed association rule mining for numeric attributes, and later they proposed a method to generate two-dimensional association rules; this method is able to find a rule for more than two attributes, like:

$$((\text{Age}, \text{Balance}) \in P) \Rightarrow (\text{Cardloan} = \text{Yes}) \tag{8}$$

where P is a planar region. Later, while Lent et al.[96] proposed a geometric-based algorithm to cluster two-dimensional association rules, Yoda et al.[97] investigated the problem of finding useful regions for two-dimensional association rules, multi-dimensional association rules mining by [98], and quantitative association rules by [99].

3.3 Evaluation of association rules

Given the high number of candidate rules that could possibly be generated by an algorithm, it is necessary to filter out those rules that are neither strong nor interesting. As we saw before, a strong rule is one that has support and confidence values greater than minimum-support and minimum-confidence thresholds, respectively, where the support is evaluated during the frequent itemset generation step[10], and the confidence is computed during the rule generation step. Nevertheless, this measure by itself is not enough to properly evaluate the final usefulness of a rule[79, 82]. Thus, several other metrics have been proposed to evaluate the interestingness of an association rule. These metrics provide an estimate of how correlated or associated are the antecedent and consequent itemsets of an association rule. Table 1 shows some of the most common or basic metrics[79, 82, 100].

Table 1 Some of the most basic metrics to evaluate the interestingness of association rules

Metric name	Metric expression
All_confidence	$\min\left(p\left(A\vert B\right),\ p\left(B\vert A\right)\right)$
Max_confidence	$\max\left(p\left(A\vert B\right), p\left(B\vert A\right)\right)$
Kulczynski	$\frac{1}{2}\left(p\left(A\vert B\right)\ +\ p\left(B\vert A\right)\right)$
Cosine	$p\left(A,B\right)/\sqrt{p(A)p(B)}$
Jaccard	$p\left(A,B\right)/\left(p\left(A\right)+p\left(B\right)-p\left(A,B\right)\right)$

As we can see from Table 1, all the metrics involve probability scores from the antecedent and consequent itemsets in a rule as a form to evaluate a degree of association between the two. One of the simplest metrics is All_confidence which measures the minimum confidence level of a rule, i.e., it computes a confidence score for each of the two possible forms of an association rule, namely $A \to B$ and $B \to A$, and returns the lowest of the scores to provide a lower-bound on the confidence of a rule.

A slightly more complex metric is the Cosine which provides a score on the similarity or relatedness of itemsets A and B by computing co-occurrence scores (in the extremes, a score of 0 indicates no relationship between the two itemsets and a score of 1 indicates a perfect relatedness of these two). This metric is similar in form to another popular metric, namely Lift[100], which computes the ratio of the co-occurrence of A and B to no co-occurrence, i.e., $p\left(A,\ B\right)/p\left(A\right)p\left(B\right)$; thus, Lift provides an estimate of how independent are itemsets A and B. Perfect independence is indicated with a $\text{Lift}\left(A,B\right)=1$ score meaning that the joint probability equals the factorized marginal probabilities (a score less than one can be interpreted as a negative correlation while a score greater than one as a positive correlation). However, Lift, as opposed to Cosine (and any of the metrics in Table 1) is not null-invariant, which means that it is affected by the number of transactions in which neither A nor B appear making it a sensitive metric.

While other desirable properties for metrics, besides null-invariance, have been proposed in the literature (see [100] for a thorough review), as well as other metrics based on information theory, such as [101], it is important to manually assess which metric to use based on the final objective of the user, since different metrics have different behavior and different interpretations.

[10] The support count of an itemset can be seen as the unnormalized support of a rule. Formally, the support of a rule can be computed as the joint probability of the itemsets in the antecedent and consequent of the rule: $p\left(A \cap B\right)$.

3.4 Current challenges

Frequent itemset generation has received significant attention in the DM field since it is the bottleneck problem for mining association rules. As a result, several efforts have been directed towards proposing algorithms that can reduce the computational complexity of recovering frequent itemsets from databases. Nevertheless, two open and current challenges are receiving increasing attention, namely deriving a small set of high-quality frequent itemsets that are highly representative of the database under study[102, 103] and handling uncertain (probabilistic) databases[103, 104].

As we saw in Sections 3.1 and 3.2, traditional techniques of frequent pattern mining have been successful in mining high-quality patterns. However, the importance of patterns is often measured not only by their frequency but also by their utility, interestingness, weight, risk, or profit. In order to address the limitations of traditional algorithms, Gan et al.[105] proposed an efficient algorithm, high utility occupancy pattern mining (HUOPM), for mining high-utility occupancy patterns in transactional databases. Moreover, they designed two novel data structures, frequency-utility tree (FU-tree) and FU-table, for efficiently pruning. Furthermore, [106] is the first work that works on mining potential high utility-occupancy patterns in uncertain databases. The proposed algorithm, named high-utility-occupancy pattern mining in uncertain databases (UHUOPM), measures support, probability, and utility occupancy as user preferences. A series of pruning methods and data structures are also applied to enhance efficiency and reduce the consumption of computing resources. Since most of the mining algorithms often return a large number of pattern candidates, Vo et al.[107] aim at tackling the inefficiency issue of mining closed potential high-utility itemsets (CPHUIs) from uncertain databases. The proposed CPHUI-List outperforms previous work (CHUI-miner) for real-life databases in terms of running time and memory cost.

4 Knowledge extraction from machine learning systems

In Sections 2 and 3, we surveyed methods to mine knowledge from two types of data, namely natural language text and transactional databases. These methods allow us to recover concrete information (factual knowledge) or novel patterns in the form of logic formulas that help us both better understand the data and draw some conclusions. However, another direction of knowledge mining is oriented towards extracting a logical rationale, in the form of logic rules, of how trained, complex machine learning systems make a prediction.

Therefore, the objective in this scenario is to understand the decision process of ML systems. Many of these systems consist of a big set of parameters (sometimes in the order of millions of parameters) that process the input information through non-linear functions; this setup makes the decision process of these systems difficult to be interpreted by a human. Thus, understanding how an ML system transforms an input instance into the observed output remains a big challenge. Nevertheless, the logic behind the decision process of practical applications of ML systems in critical areas such as medicine, credit risk assessment, or education needs to be clearly understood by its users, who can then validate the output of the system and decide whether to use it or discard it. This problem can be framed as the problem of extracting human-understandable knowledge from complex ML systems, which is known as the task of Interpretability[11].

In this way, the final objective of the interpretability problem is that the knowledge extracted from an ML system, grounded in logic rules, faithfully mimics the predictive behavior of this system. This means that, ideally, the logic rules will encode in human-comprehensible way characteristics or patterns of the system so that a user understands how the system transformed an input to the observed output; thus, we assume that the logic rules will be able to mirror every correct and incorrect prediction from the system under study.

In this section, we present different approaches to extracting the knowledge learned by complex ML systems, also known as black-box systems, due to their un-interpretability. Similar to previous sections, we mainly target works in the literature where the knowledge extracted is in the form of logic rules (this is one of the most popular types of knowledge representation in the interpretability literature). Most of the black-box systems we review in this section are neural networks due to their wide acceptance and use in ML and related fields. As noted in [108−110], the problem of interpretability can be characterized across several dimensions: The type of methodology used to extract the knowledge (pedagogical VS. decompositional); the scope of the interpretation (global VS. local); the choice of how to represent the knowledge extracted (e.g., logic-based representations or probabilistic representations); the comprehensibility of the knowledge extracted (i.e., how easy it is for a person to understand the working logic of the black-box system via the knowledge extracted); the algorithmic complexity of the knowledge extraction method; the user expertise (this varies according to the background knowledge of the user, from inexperienced to expert); and the form of the ML system to be studied. We identify two dimensions as the most important dimensions for this survey paper, namely the methodology type used for knowledge extraction and the type of target system.

The two most common approaches to extracting knowledge, in the form of logic rules, from a trained ML

[11] Sometimes also referred to as explainable artificial intelligence (xAI).

system are the pedagogical and decompositional methods. The former method can take either a supervised or unsupervised learning approach to learn a set of logic rules from an ML system's behavioral data. The latter method requires to manually open the black-box system to characterize at the parameter-level[12] (neuron-level if the target system is an NN) the internal functioning of the system through a set of logic rules (usually, one rule per parameter or set of parameters[111]). In both scenarios, the extracted logic rules serve as a proxy to understand the decision process of the ML system. From now on, we shall refer to this set of rules as either the proxy or the interpretable model.

In the dimension of type of target system, we distinguish between two types of neural networks, namely feed-forward (or multi-layer perceptron) networks and neural encoders, similar to those described in Section 2.1.

In Sections 4.1 and 4.2, we provide a survey of previous works across the two dimensions described above.

4.1 Pedagogical approach

In this approach, it is necessary to collect behavioral data from the target ML system to induce the proxy model. If the system is a classifier (a very common type of system), building a behavioral dataset reduces to showing an input instance x_i to the black-box system, recording its output $\hat{y}_i$ in order to build a training instance $(x_i,\ \hat{y}_i)$ for the proxy model, and repeating this operation across a set of instances $(X,\ \hat{Y})$. In this way, we can obtain a training set that shows the predictive behavior of the target system and on which to learn the interpretable model. In what follows, we show examples of previous work under this type of approach for both feed-forward networks and neural encoders.

Feed-forward networks. An example of the pedagogical approach in the literature is the seminal work of [112] which introduces the algorithm TREPAN. This algorithm induces a decision tree[13] as a global[14] proxy model for understanding the predictive behavior of feed-forward NNs[15]. The training data to induce the tree is obtained by having the target neural network relabel the class label of each of the instances used to train it. In this way, the decision tree is expected to represent the concept learned by the NN and thus learns to label instances in the same way as the NN. A salient feature of TREPAN with respect to other tree learning algorithms in the literature, such as classification and regression trees (CART)[116], is its ability to learn expressions of the type $m-of-n$ for each internal node of the tree. This expression indicates that m constraints are to be satisfied out of n Boolean conditions. Taking an example from [112], given three Boolean features a, b, and c, with $m=2$, the expression to satisfy at a given internal node n_i of the decision tree is $2-of-\{a,b,c\}$ which is equivalent to satisfying the expression (a AND b) OR (a AND c) OR (b AND c).

Similar pedagogical approaches have been proposed in the literature. For example, Ribeiro et al.[117] aim to provide local explanations by perturbing a target instance to obtain a training set in the vicinity of such an instance to train a proxy model, that can include decision trees and linear models. Domingos[118] extracts propositional logic rules from an ensemble of ML systems by expanding the original training set used to train the ensemble with behavioral data from the ensemble. On the other hand, d'Avila Garcez showed that while pedagogical approaches are sounded, their complexity is usually greater than that of decompositional approaches.

Neural encoders. Some of the first works to extract logic rules from neural encoders, similar to those reviewed in Section 2, are the works of [119, 120], where the target system is a matrix factorization system, similar to that of [2], which learns vector representations for relations and pairs of entities from a knowledge base. More concretely, the objective of this ML system is to populate a knowledge base (based on the observed facts from this KB); this system is represented as a matrix where rows correspond to named entity pairs and columns to relation types. Thus, a prediction in this system (a cell in the matrix) corresponds to a probability score of the likelihood of a fact being true; e.g., given the pair of entities (Beijing, China) and the relation locatedIn, a prediction from this system would be p(locatedIn(Beijing, China)) =

[12] Or set-of-parameters-level.

[13] As noted in [113, 114], a decision tree can be converted into IF-THEN logic rules where each internal node of the tree in a path from top to bottom represents an antecedent (the IF part of the rule), and leaf nodes deciding on the class label of an instance represent consequents (the THEN part of the rule). Thus, a path on a decision tree has a logic rule counterpart of the form IF $f_i = value_i$ AND $f_j = value_j$ AND $f_k = value_k$ THEN $class\_label = \hat{y}$, where f_i, f_j and f_k represent features of the input space.

[14] We note that the scope of a proxy model can either be at the local or global level. While a local explanation of an ML system targets a single prediction, a global explanation aims to account for the predictive behavior of the black-box system across a collection of instances; thus, a global proxy model can explain the system's behavior for any input instance. Choosing one or the other explanation type usually corresponds to the algorithmic complexity of the method to extract the proxy model (global explanations may be NP-Hard to compute in some cases[115]).

[15] It is assumed that each input neuron in a neural network receives an input value (a feature) that is human-understandable. For example, if the concept that the target NN learns is to classify houses as cheap or expensive, possible input features are the number of rooms in a house, the age of the house, or the size of the house in square meters.

0.99 indicating that the confidence of the system in the fact locatedIn(Beijing, China) being true is very high[16]. To explain how this system reaches a prediction, Sanchez Carmona and Riedel[120] extracted logic rules of the form $A\,(x,y) \Rightarrow B\,(x,y)$ where the predicate symbols A and B represent relation types and the arguments x, y represent named entities. The extraction of these rules was done via unsupervised learning: Predictions in each cell of the matrix were binarized to $\{0,\ 1\}$ with a threshold $t \leq 0.5$ which yielded a binary matrix of predictions. Each column in the matrix was taken as the sample of an independent variable, and the label of that column (the relation type) bounded a logic predicate symbol. To generate logic rules, a similar method to generate association rules from Section 3 was taken: Each possible pair of columns A, B generate two candidate rules, namely $A\,(x,y) \Rightarrow B(x,y)$ and $B\,(x,y) \Rightarrow A\,(x,y)$. In order to accept a candidate rule, mutual information (support measure) and conditional probabilities (confidence measure) are computed for each pair of columns; if both measures surpass a threshold, a rule is accepted. To explain a prediction of the ML system, the application of modus-ponens (a type of logic inference) is carried out on the set of extracted rules where observed facts from the knowledge base used to train the ML system act as triggers by bounding antecedents of a subset of rules until the target rule containing the predicted fact as a consequent is bounded. For example, the above prediction could be explained through the observed fact capitalOf(Beijing, China) and the rules {capitalOf(x, y) $\Rightarrow$ cityOf(x, y), cityOf(x, y)$\Rightarrow$locatedIn(x, y)}.

However, in the work of [120], the logic rules proposed were not capable of faithfully mimicking the decision process of the black-box system due to both the complexity of such system containing around 4 000 relation types across thousands of entity pairs and the probabilistic nature of the system. The best interpretable model found to mimic this system was a tree-structured Bayesian network, a type of probabilistic model that can compute the joint probability of all random variables, with *Precision* scores over 70% across different *Recall* levels.

Other works in the literature have been proposed afterward. For example, based on the approach of [119], Peake and Wang[121] extracted the same type of association rules ($A \rightarrow B$) using the Apriori algorithm where the target system was also a matrix factorization system used as a recommendation system. Thus, in this case, the consequent of a rule is an item recommended by the black-box system, and the antecedent corresponds to an item with which the user previously interacted with. On the other hand, Gusmão et al.[122] proposed a system-agnostic method to extract weighted Horn rules from neural encoders similar to the one used in [119], where entities and relations are mapped to vector embeddings.

4.2 Decompositional approach

Feed-forward networks. The seminal work of [111] proposes an early method for extracting global explanations in the form of logic rules from feed-forward networks[17] (though this method can be applied to RNNs as well). This method extracts classification IF-THEN rules of the form IF $x_1 \in [a_1,\ b_1]$ AND $x_2 \in [a_2,\ b_2]$ AND $\cdots$ AND $x_n \in [a_n,\ b_n]$ THEN class C=k where each x_i corresponds to an input variable, each interval $[a_i, b_i]$ corresponds to a real-valued interval, and C corresponds to a class label predicted by the NN. To induce a set of IF-THEN rules that globally account for the behavior of the target NN, Thrun[111] used a mathematical procedure called validity interval analysis[123].

This procedure can be seen as a search in the space of intervals of the form $[a_i,\ b_i]$, where the activation value (output) y_i of each neuron in a neural network (including input, hidden, and output neurons) is bound to its own interval $i = [a_i,\ b_i]$. The objective is to find the tightest interval for the activation value of each neuron; i.e., the procedure aims to find the max a_i and min b_i constrained to $y_i \geq a_i$ and $y_i \leq b_i$. Since the final goal of this search is to find maximum and minimum values for each interval subject to linear constraints, Thrun[111] set this search as a linear programming problem. In this way, the problem of rule extraction from a NN reduces to generating the set of IF-THEN rules with antecedents corresponding to a set of valid intervals characterizing the input space that map to an output from the NN. A strategy to find this set of rules is to start a search with rules that under-partition the input space and to gradually find the set of valid intervals (rules antecedents) through the validity interval analysis that correctly map a set of input instances to the corresponding NN′s predicted class label.

Other algorithms to extract rules operate in a similar principle; they extract rules from each neuron (or sets of neurons) where features or activation values (outputs from a neuron) are used as antecedents and/or consequents in each rule; e.g., in the work of [124], this approach is used to extract interpretable rules from deep neural network classifiers trained on several types of datasets (including vision datasets). For other similar decompositional approaches, we refer to the works of [125, 126].

Neural encoders. An early decompositional algorithm to induce logic rules from neural encoders is the work of [127], where neural encoders are similar to those reviewed in Section 2 (the neural encoders learn vector or matrix embeddings for both relations and entities). The proposed algorithm aims to find Horn rules of the form

[16] This system makes a prediction by applying a sigmoid function to the dot product of the vector representations corresponding to an entity pair and a relation.

[17] We hold the same assumption as in Section 4.1, namely that the input features of a neural network are human interpretable.

$B_1(a,\ b)$ AND $B_2(b, c) \Rightarrow H(a, c)$, where predicates B_1, B_2, H correspond to relation types and their arguments correspond to entities. To do so, the algorithm first proposes three sets of relations, one of the head predicates (H), one of starting relations (B_1), and one of ending relations (B_2). After that, the algorithm seeks for head relations similar (according to the Euclidean metric) to the composition of pairs of body relations, $b_i \in B_1 \circ b_j \in B_2$, where the composition function can be a vector addition or matrix multiplication according to the form of the relation representation. Candidate rules are pruned by ranking them according to their similarity to target head relations. Furthermore, to constrain the search space, the types of the entities are taken into account to avoid wrong entity types grounding arguments of the predicates; e.g., the predicate born_in(a, b) can only take entities of type person and country, respectively. Examples of rules extracted are athlete_play_in_team(a, b) AND team_play_sport(b, c) $\Rightarrow$ athlete_play_sport(a, c), born_in_location(a, b) AND location_in_country(b, c) $\Rightarrow$ nationality(a, c). Evaluation of the rules extracted shows *Precision* scores to degrade as the number of entity pairs and relation types increase with *Precision* scores between 30%−40% when the number of predictions is around the tens of thousands.

Another example of a decompositional approach for neural encoders is the work of [128], where the target ML system is an LSTM used for sentence classification. The first step of the proposed method is to analyze the importance of each word in a phrase to the final classification done by an LSTM. The hypothesis is that the target system may learn a different importance weight for each word, thus contributing in a different way to the final output. For example, for a sentiment classification task, the system may classify the sentence “the movie was horrible” with the class label Negative mainly due to the presence of the word horrible. In order to extract the importance weights, each input to the system is analyzed in the context of the internal components (more concretely, the cell states and forget gates) to factorize the output according to such weights. The second step of the method applies a rule-based classifier on top of the extracted patterns to mimic the predictive behavior of the LSTM.

As we can see from the works surveyed, decompositional approaches are ad-hoc to the type of model under study; thus, specific algorithms must be developed for each system. Furthermore, as noted in [3], decompositional approaches tend not to be as sounded as pedagogical ones, though their complexity tends to be lower than that of pedagogical approaches.

4.3 Evaluation of the knowledge extracted

The most common evaluation is the measure of fidelity or faithfulness (how well the proxy model mimics the predictive behavior of the black-box system), which is usually evaluated through the $F1$ metric (and in cases where the dataset is balanced through accuracy scores), as explained in Section 2.5. However, the evaluation may depend on the type of task carried out by the black-box system. For example, in [119], the fidelity of the interpretable models was evaluated using *Precision-Recall* scores (similar to the evaluation of distant-supervised models from Section 2; see Section 2.5) since part of the task of the black-box system is to rank the predicted relations. Regardless of the metric used to evaluate the faithfulness of an interpretable model, this model is tested on an unseen test set where, in the same way as the training set, the instances are not labeled by a human annotator but by the black-box target system.

Generalization ability is another type of evaluation, where the proxy model is evaluated on a test set where labels come from human annotation. In this way, the $F1$ score of the proxy model is compared against that of the target system; if the proxy model is faithful to the target system, similar $F1$ scores and generalization abilities are to be seen[120].

Manual evaluation of the induced rules, which may be followed by *Precision* scores, is another type of assessment though this is the less frequent type of evaluation due to the human effort and time required. Nevertheless, as we will see in Section 4.4, this type of evaluation has recently received an increasing attention.

4.4 Current challenges

While most of the recent efforts in interpretability are focused on developing new methods for computing local explanations (e.g., using game theory theorems[129], information theory principles[130], or a case-based reasoning approach[131]), one of the biggest challenges is evaluating both the faithfulness of extracted knowledge and the usefulness of such knowledge for humans to understand the black-box model′s decision process[132].

Evaluating how faithful the extracted knowledge is to the true black-box model′s knowledge is challenging, since such true knowledge is often unknown[133]. Recent work has provided testbeds and benchmarks to measure the fidelity of the extracted knowledge, especially for local explanations, where mimicking the behavior of the black-box system is more challenging since only one behavioral data point is considered for the target explanation. For example, Sippy et al.[133] evaluated local explanation methods by corrupting in a controlled way part of the training data of three text classification datasets, which induced the expected wrong behavior of the black-box system. In this way, it was measured to what extent the explanation methods could recover such a wrong behavior. Similarly, Bastings et al.[134] induce a bias in a dataset by augmenting it with instances that contain this bias; local explanation models are then evaluated on their ability to recover such a bias by comparing them on two scenarios:

One where the explanation model extracts an explanation from the black-box system trained on the original dataset, and another one where the target system is trained on the augmented, biased dataset. Results from both [133, 134] have revealed conflicts among different explanation methods suggesting further work on this challenge.

Assessing the final use of the explanations, on the other hand, has received recent attention from the ML community. Even though assessing how well logic rules, and other interpretable models, convey the decision-making process of a particular system to people has been investigated before[113], until recently, more studies have been carried out. For example, Yuan et al.[135] investigated display factors that optimize the way IF-THEN logic rules can be displayed for people to better understand them. Lage et al.[136] carried out controlled experiments using crowdsourcing to investigate which factors (such as the size of the explanation) lead people to better understand an explanation in the form of decision sets (a special type of logic rules).

5 Discussions and conclusions

Throughout Sections 2–4, we surveyed previous works in each field for the problem of knowledge mining. Even though the overall problem is similar across fields — extract knowledge from a data source and structure it into a particular representation — some of their traits vary. In Section 5.1, we provide an account of these traits. Then, in Section 5.2, we will explore previous work where a bridge has been built across these three fields to further stimulate forging new bridges. Finally, in Section 5.3, we will provide what we believe to be a long-term research direction for Knowledge Mining.

5.1 Research traits across fields

We identify five dimensions that we believe characterize the knowledge extraction work across fields, namely objectives, methods, research orientation, data, and evaluations. In what follows, we provide a comparison of the Knowledge Mining problem for the NLP, DM, and ML fields across these five traits.

Objectives. At a high level, the three fields share this trait, where the overall goal is to extract knowledge from a dataset in order to understand the data. However, at a lower level, subtle differences arise across the fields.

In the NLP field, the target knowledge type is factual knowledge in the form of grounded logic predicates where predicate symbols correspond to relation types and arguments correspond to named entities. Automatically building a knowledge base of facts from text is one of the main goals in NLP, and it has the fundamental purpose of avoiding human effort towards recovering this explicit information. Furthermore, this KB is intended to be not only helpful to understand the source text but also to be used in subsequent steps of a bigger decision process, such as the prediction of new facts in the biomedical domain.

On the other hand, for the data mining field, a highly sought type of pattern is an association rule. As we saw in Section 3, this pattern allows us to represent relationships between sets of items in a transactional database. Moreover, association rules allow us to better understand purchasing behavior and to implement effective marketing strategies based on such knowledge.

In the case of the ML field, most of the recent ML models are increasingly complex, and obtaining an explanation of their decision process with the purpose of both validating their knowledge and discovering possible biases through a simple inspection of their parameters is extremely difficult. As we showed in Section 4, a solution to this problem is to extract the knowledge from a trained ML system and to structure it into logic rules so it can be understandable to users. Thus, the main goal of understanding behavioral data in this scenario is to inspect and verify the consistency of the knowledge encoded in a black-box system, a goal that is extremely important in fields such as medicine or education.

Methods. This trait, interestingly, is not unique across fields; we find an overlap between NLP and ML methods and the beginning of another one between DM and ML methods that we believe to be an interesting future research direction. As we saw in Section 2, the methods used in the NLP field to extract relational knowledge heavily rely upon machine learning and probabilistic models, especially neural encoders and CRFs, where the main learning paradigms used are supervised, distant-supervised, and unsupervised learning. On the other hand, methods in data mining to mine association rules mainly rely on search algorithms to effectively discover both frequent itemsets and associations among them. In the case of the problem of interpretability, the machine learning field has made use of two methods, machine learning models and algorithms; while the first method is mainly used in the pedagogical approach to learning a proxy model, the second method is mainly used in the decompositional approach to search for logic rules that can explain the input-output behavior of a set of parameters of the black-box system. Furthermore, as we saw in Section 4.1, some works from the ML community used variants of the Apriori algorithm to find logic rules that can serve as a proxy model. We believe research in interpretability can take advantage of the search algorithms developed in the data mining field to search for logic rules in a similar way to how association rules are mined.

Research orientation. At this point, it seems clear that one of the main directions in the NLP field for the task of information extraction is towards building more accurate machine learning systems that are better able to generalize to different text domains (such as finance, medicine, news, etc.) in order to build consistent knowledge bases that accurately reflect the world′s true knowledge. As for the data mining field, the main efforts in research are directed to designing algorithms with less computational complexity that are more efficient in discovering frequent itemsets and association rules. Also, other efforts in data mining are directed towards better using different types of computer memory to handle bigger databases in less time. Finally, research efforts in the ML field for interpretability are not only oriented towards improving their methods — obtaining more faithful models — but also towards studying the usefulness of such methods in helping people understand the decision process of complex ML systems.

Data. The type and nature of the data used across fields for the task of knowledge extraction may be the trait that better allows us to distinguish from each of them. The NLP field is mainly focused on natural language text — unstructured data — to extract facts, where the most popular datasets for both NER and RE tasks are mainly collected from thousands of news articles spanning across years. On the other hand, the market basket analysis problem (and its variants) from DM is only concerned with transactions stored in a database — structured data — where the size of a database can be in the order of millions of transactions. Different to the NLP and DM datasets, the ML field mainly uses predictions from complex machine learning systems — behavioral data — as the source from where to extract hidden knowledge, where the size of these behavioral datasets varies according to the type of explanation sought: While global explanations require thousands of predictions, local explanations are usually extracted from a dozen or hundreds of predictions due to restrictions on the nature of the problem[18].

Evaluations. Evaluating the performance of NLP systems reduces to assessing how well the system is able to both recover text spans that refer to target entities and classify the type of the entities and their relationship. This evaluation is done through several metrics such as *Precision*, *Recall*, and $F1$. Evaluating association rules mined from databases can be done by measuring how strong and interesting are such rules through a myriad of metrics such as support, confidence, Cosine, and Jaccard, among others, that can provide us with a score for these two properties. A typical example of a strong and interesting rule is the pattern diapers → beer, which allows us to see an unexpected purchasing behavior that, even though it is highly frequent in a database, would be difficult to discover manually. The main evaluation of interpretable models measures how faithful they are to the behavior of the black-box system to be explained. Similarly to the evaluation of NLP systems described above, the $F1$, *Precision*, and *Recall* scores are obtained to compare the behavior of the interpretable model with that of the target system. However, unlike the NLP evaluation, the gold test set is not human-annotated but rather machine annotated, since we mainly care about how well the proxy model matches the predictions of the black-box system.

5.2 Intertwining three fields for knowledge mining: NLP, DM and ML

We have seen in previous sections the main works in the literature and the main traits of the knowledge mining research for each of the fields we target in this paper, namely NLP, DM and ML. We surveyed and described their main methods, goals, evaluations, and other characteristics. We saw that, even though these three fields have the same high-level goal of extracting knowledge from a source of data to better understand such data, each field has its own research approach (with some traits shared among them, such as evaluation metrics and methods). However, we believe that it is not only possible to consolidate the knowledge extraction task across these three fields but doing so is a new research direction calling to build bridges with the aims of proposing new methods, improving evaluations, exploring new frontier problems, studying new types of data, building more complex and accurate applications[19], and ultimately, advancing in the quest of accurately understanding the data under study.

In this section, we pinpoint efforts in the literature to intertwine the fields of NLP, DM, and ML for the task of knowledge mining.

Early works have tried to build bridges across these disciplines, such as [137, 138] where association rule mining methods, similar to those surveyed in Section 3, are proposed to extract rules from knowledge bases that are automatically built from text using methods similar to those presented in Section 2. This bridge between the NLP and DM fields can provide a valuable pipeline for applications in the medical domain to extract novel relationships from text among entities such as diseases and symptoms.

In a related line of work, there have been efforts to make interpretable, to some degree, the decision process of complex neural encoders, similar to those described in

[18] As we saw in Section 4.1, a single instance is perturbed to generate the behavioral dataset; thus, the number of predictions obtained will be proportional to the number of perturbations performed; however, some of the perturbations may yield out-of-domain instances which may not be representative of the domain where the black-box system was trained on; thus, these instances may elicit an inconsistent behavior from the black-box system.

[19] Domains such as medicine require applications to extract knowledge such as gene and drug entities and their different types of relations from biomedical texts or explanations from machine learning systems predicting relationships between entities as accurately as possible.

Section 4.1, for the task of knowledge base completion where given a knowledge base populated with some relational facts (as those extracted by an NLP system described in Section 2), the aim is to fully populate the knowledge base. For example, Xie et al.[139] added a sparse attention mechanism to a neural encoder to allow sharing knowledge between different relation embeddings and interpreting through the weights learned by the attention mechanism to what extent each piece of latent knowledge is shared by specific relation types.

Another type of bridge built between fields is solving the problem of extracting logic rules from knowledge bases as those used in the NLP field. For example, in the DM field, Galárrage et al.[140] extracted Horn logic rules from knowledge bases through a search in the space of logic predicates (in each step of the search, a predicate is added to the rule) which is then evaluated through confidence and support scores; these rules can then be used to predict new facts or to better understand the knowledge stored in the knowledge base. In the ML field, Yang et al.[127] took a different approach in which the KB was first encoded by a neural encoder, and then logic rules were generated from the vector embeddings learned by the encoder.

Another example is the work of [119, 120] (described in Section 4.1) where the three fields, NLP, DM and ML, are intertwined in a single work: An interpretable model in the form of logic rules is used to understand the predictions of a neural encoder (similar to those described in Section 2) which is learned under a pedagogical approach (as described in Section 4) using a variant of the Apriori algorithm (Section 3). As we can see in this example, particular factors of a research problem coming from the three fields are intertwined: The need for an interpretation (the problem) of a black-box system used for predicting relations between entities (the subject of study) leads to extract association rules from this system (the method).

The works cited above are some representative examples of bridges built so far. However, we believe that other possible ways to intertwine these three areas can lead to other types of bridge that will result not only in new approaches but also in new directions for the problem of knowledge mining.

5.3 Future research direction

We believe the task of knowledge mining has an extensive set of choices for future research directions: From short-term research directions such as proposing new machine learning methods, algorithms, and evaluation metrics to middle-term research directions such as consolidating the three fields targeted in this paper (possibly following and extrapolating from the trends described in Section 5.2). In this section, we opt to briefly focus on one possible long-term research direction aligned to the golden objective of artificial intelligence of human-level intelligent agents, namely knowledge mining as a component of an intelligent agent.

As proposed in [141], an intelligent agent is composed of two parts: A component that is able to understand human language and store information in its knowledge base, and a component which can retrieve knowledge from large amounts of different types of data (such as text and structured databases), make inferences with this knowledge, and provide an explanation of such inferences. It is in the second component where knowledge mining plays a role. The research works surveyed in Sections 2−4 can be used as the building blocks to achieve this component. Furthermore, to advance in the capabilities of this component, and as proposed in [142], the component embodied by a knowledge mining system will not only be able to integrate knowledge from various types of data sources in an interpretable and consistent way, but it will also be able to generate new knowledge which may be achieved by extrapolating from the patterns extracted from the data while abstracting it at a human conceptual level. Ultimately, we believe, knowledge mining will play a significant role in such intelligent agents.

While the research direction proposed here may rest in the distant future, we believe the efforts from the fields surveyed are contributing to this golden objective.

We hope this paper will motivate future cross-disciplinary research agendas that take us a step closer to a new state-of-the-art in knowledge mining.

Open Access

References

[1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, vol. 17, no. 3, pp. 37–54, 1996. DOI: 10.1609/aimag.v17i3.1230.

[2] S. Riedel, L. M. Yao, A. McCallum, B. M. Marlin. Relation extraction with matrix factorization and universal

schemas. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, Atlanta, USA, pp. 74–84, 2013.

[3] A. S. d'Avila Garcez, K. Broda, D. M. Gabbay. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, vol. 125, no. 1–2, pp. 155–207, 2001. DOI: 10.1016/S0004-3702(00)00077-1.

[4] S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*, 3rd ed., Harlow, USA: Pearson Education, 2010.

[5] D. Jurafsky, J. H. Martin. Speech and Language Processing, [Online], Available: https://web.stanford.edu/~jurafsky/slp3/ed3book_dec302020.pdf, 2021.

[6] T. Rocktäschel, S. Singh, S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, Denver, USA, pp. 1119–1129, 2015. DOI: 10.3115/v1/N15-1118.

[7] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[8] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, ACL, Minneapolis, USA, pp. 4171–4186, 2019. DOI: 10.18653/v1/N19-1423.

[9] E. F. Tjong Kim Sang, F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, ACL, Edmonton, Canada, pp. 142–147, 2003.

[10] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, R. Weischedel. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, Association for Computational Linguistics, New York City, USA, pp. 57–60, 2006.

[11] J. P. C. Chiu, E. Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016. DOI: 10.1162/tacl_a_00104.

[12] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa. Natural language processing (Almost) from scratch. *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[13] A. Passos, V. Kumar, A. McCallum. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the 18th Conference on Computational Natural Language Learning*, ACL, Ann Arbor, USA, pp. 78–86, 2014. DOI: 10.3115/v1/W14-1609.

[14] D. E. Appelt, J. R. Hobbs, J. Bear, D. J. Israel, M. Tyson. FASTUS: A finite-state processor for information extraction from real-world text. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, Chambery, France, pp. 1172–1178, 1993.

[15] T. Eftimov, B. K. Seljak, P. Korošec. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PLoS One*, vol. 12, no. 6, Article number e0179488, 2017. DOI: 10.1371/journal.pone.0179488.

[16] H. Isozaki, H. Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th International Conference on Computational Linguistics*, ACL, Taipei, China, pp. 1–7, 2002. DOI: 10.3115/1072228.1072282.

[17] J. D. Lafferty, A. McCallum, F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, USA, pp. 282–289, 2001.

[18] A. McCallum, W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, ACL, Edmonton, Canada, pp. 188–191, 2003. DOI: 10.3115/1119176.1119206.

[19] Z. H. Huang, W. Xu, K. Yu. Bidirectional LSTM-CRF models for sequence tagging. [Online], Avaiable: https://arxiv.org/abs/1508.01991, 2015.

[20] X. Z. Ma, E. Hovy. End-to-end sequence labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, Berlin, Germany, pp. 1064–1074, 2016. DOI: 10.18653/v1/P16-1101.

[21] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer. Neural architectures for named entity recognition. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, San Diego, USA, pp. 260–270, 2016. DOI: 10.18653/v1/N16-1030.

[22] J. Hammerton. Named entity recognition with long short-term memory. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, ACL, Edmonton, Canada, pp. 172–175, 2003. DOI: 10.3115/1119176.1119202.

[23] A. Akbik, D. Blythe, R. Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, ACL, Santa Fe, USA, pp. 1638–1649, 2018.

[24] A. Akbik, T. Bergmann, R. Vollgraf. Pooled contextualized embeddings for named entity recognition. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, ACL, Minneapolis, USA, pp. 724–728, 2019. DOI: 10.18653/v1/N19-1078.

[25] K. Liu, Y. Fu, C. Q. Tan, M. S. Chen, N. Y. Zhang, S. F. Huang, S. Gao. Noisy-labeled NER with confidence estimation. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, pp. 3437–3445, 2021. DOI: 10.18653/v1/2021.naacl-main.269.

[26] D. J. Zeng, K. Liu, Y. B. Chen, J. Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Lisbon, Portugal, pp. 1753–1762, 2015. DOI: 10.18653/v1/

D15-1203.

[27] E. Sandhaus. The New York Times Annotated Corpus LDC2008T19. Philadelphia, USA, 2008. DOI: 10.35111/77ba-9x74.

[28] Y. H. Zhang, V. Zhong, D. Q. Chen, G. Angeli, C. D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 35–45, 2017. DOI: 10.18653/v1/D17-1004.

[29] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, J. Ellis. Overview of the TAC knowledge base population track. In *Proceedings of Text Analysis Conference*, 2010.

[30] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, R. M. Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, European Language Resources Association, Lisbon, Portugal, pp. 837–840, 2004.

[31] S. M. Strassel, M. A. Przybocki, K. Peterson, Z. Y. Song, K. Maeda. Linguistic resources and evaluation techniques for evaluation of cross-document automatic content extraction. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, European Language Resources Association, Marrakech, USA, pp. 2706–2709, 2008.

[32] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz. SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, Uppsala, Sweden, pp. 33–38, 2010.

[33] M. Banko, O. Etzioni. The tradeoffs between open and traditional relation extraction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, ACL, Columbus, USA, pp. 28–36, 2008.

[34] R. C. Bunescu, R. J. Mooney. Subsequence kernels for relation extraction. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, MIT Press, Vancouver, Canada, pp. 171–178, 2005.

[35] G. D. Zhou, J. Su, J. Zhang, M. Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL, Ann Arbor, USA, pp. 427–434, 2005. DOI: 10.3115/1219840.1219893.

[36] A. Culotta, J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL, Barcelona, Spain, pp. 423–429, 2004. DOI: 10.3115/1218955.1219009.

[37] Q. Li, H. Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, Baltimore, USA, pp. 402–412, 2014.

[38] M. Miwa, M. Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, Berlin, Germany, pp. 1105–1116, 2016. DOI: 10.18653/v1/P16-1105.

[39] B. W. Yu, Z. Y. Zhang, X. B. Shu, T. W. Liu, Y. B. Wang, B. Wang, S. J. Li. Joint extraction of entities and relations based on a novel decomposition strategy. In *Proceedings of the 24th European Conference on Artificial Intelligence*, Santiago de Compostela, Spain, pp. 2282–2289, 2020.

[40] T. J. Fu, P. H. Li, W. Y. Ma. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp. 1409–1418, 2019. DOI: 10.18653/v1/P19-1136.

[41] X. Y. Li, F. Yin, Z. J. Sun, X. Y. Li, A. Yuan, D. Chai, M. X. Zhou, J. W. Li. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp. 1340–1350, 2019. DOI: 10.18653/v1/P19-1129.

[42] I. Beltagy, K. Lo, A. Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, Hong Kong, China, pp. 3615–3620, 2019. DOI: 10.18653/v1/D19-1371.

[43] H. Y. Zheng, R. Wen, X. Chen, Y. F. Yang, Y. Y. Zhang, Z. H. Zhang, N. Y. Zhang, B. Qin, X. Ming, Y. F. Zheng. PRGC: Potential relation and global correspondence based joint relational triple extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, ACL, pp. 6225–6235, 2021. DOI: 10.18653/v1/2021.acl-long.486.

[44] T. Lai, H. Ji, C. X. Zhai, Q. H. Tran. Joint biomedical entity and relation extraction with knowledge-enhanced collective inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, ACL, pp. 6248–6260, 2021. DOI: 10.18653/v1/2021.acl-long.488.

[45] J. Wang, W. Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 1706–1721, 2020. DOI: 10.18653/v1/2020.emnlp-main.133.

[46] M. Mintz, S. Bills, R. Snow, D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL, Suntec, Singapore, pp. 1003–1011, 2009.

[47] C. J. Xiao, Y. Yao, R. B. Xie, X. Han, Z. Y. Liu, M. S. Sun, F. Lin, L. Y. Lin. Denoising relation extraction from document-level distant supervision. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 3683–3688, 2020. DOI: 10.18653/v1/2020.emnlp-main.300.

[48] S. Riedel, L. M. Yao, A. McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Berlin, Germany, pp. 148–163, 2010. DOI: 10.1007/978-3-642-15939-8_10.

[49] T. G. Dietterich, R. H. Lathrop, T. Lozano-Pérez. Solv-

ing the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997. DOI: 10.1016/S0004-3702(96)00034-3.

[50] G. L. Ji, K. Liu, S. Z. He, J. Zhao. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI, San Francisco, USA, pp. 3060–3066, 2017.

[51] Y. K. Lin, S. Q. Shen, Z. Y. Liu, H. B. Luan, M. S. Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, Berlin, Germany, pp. 2124–2133, 2016. DOI: 10.18653/v1/P16-1200.

[52] Z. X. Ye, Z. H. Ling. Distant supervision relation extraction with intra-bag and inter-bag attentions. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, ACL, Minneapolis, USA, pp. 2810–2819, 2019. DOI: 10.18653/v1/N19-1288.

[53] G. Y. Wang, W. Zhang, R. X. Wang, Y. L. Zhou, X. Chen, W. Zhang, H. Zhu, H. J. Chen. Label-free distant supervision for relation extraction via knowledge graph embedding. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Brussels, Belgium, pp. 2246–2255, 2018. DOI: 10.18653/v1/D18-1248.

[54] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems* Lake Tahoe, USA, pp. 2787–2795, 2013.

[55] Z. Wang, J. W. Zhang, J. L. Feng, Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, AAAI, Quebec City, Canada, pp. 1112–1119, 2014.

[56] Y. K. Lin, Z. Y. Liu, M. S. Sun, Y. Liu, X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI, Austin, USA, pp. 2181–2187, 2015.

[57] T. Hasegawa, S. Sekine, R. Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, ACL, Barcelona, Spain, pp. 415–422, 2004. DOI: 10.3115/1218955.1219008.

[58] B. Rosenfeld, R. Feldman. Clustering for unsupervised relation identification. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, Association for Computing Machinery, Lisbon, Portugal, pp. 411–418, 2007. DOI: 10.1145/1321440.1321499.

[59] L. M. Yao, A. Haghighi, S. Riedel, A. McCallum. Structured relation discovery using generative models. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Edinburgh, UK, pp. 1456–1466, 2011.

[60] L. M. Yao, S. Riedel, A. McCallum. Unsupervised Relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, Jeju Island, Korea, pp. 712–720, 2012.

[61] B. N. Min, S. M. Shi, R. Grishman, C. Y. Lin. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, ACL, Jeju Island, Korea, pp. 1027–1037, 2012.

[62] D. Marcheggiani, I. Titov. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 231–244, 2016. DOI: 10.1162/tacl_a_00095.

[63] T. T. Tran, P. Le, S. Ananiadou. Revisiting unsupervised relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 7498–7505, 2020. DOI: 10.18653/v1/2020.acl-main.669.

[64] L. B. Soares, N. FitzGerald, J. Ling, T. Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp. 2895–2905, 2019. DOI: 10.18653/v1/P19-1279.

[65] X. Han, T. Y. Gao, Y. K. Lin, H. Peng, Y. L. Yang, C. J. Xiao, Z. Y. Liu, P. Li, J. Zhou, M. S. Sun. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, ACL, Suzhou, China, pp. 745–758, 2020.

[66] A. Yates, M. Banko, M. Broadhead, M. Cafarella, O. Etzioni, S. Soderland. TextRunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ACL, Rochester, USA, pp. 25–26, 2007.

[67] A. Fader, S. Soderland, O. Etzioni. Identifying relations for open information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, ACL, Edinburgh, UK, pp. 1535–1545, 2011.

[68] L. Del Corro, R. Gemulla. ClausIE: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, Association for Computing Machinery, Rio de Janeiro, Brazil, pp. 355–366, 2013. DOI: 10.1145/2488388.2488420.

[69] G. Stanovsky, J. Michael, L. Zettlemoyer, I. Dagan. Supervised open information extraction. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, ACL, New Orleans, USA, pp. 885–895, 2018. DOI: 10.18653/v1/N18-1081.

[70] Y. Ro, Y. Lee, P. Kang. Multi.2OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Proceedings of Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, pp. 1107–1117, 2020. DOI: 10.18653/v1/2020.findings-emnlp.99.

[71] C. G. Wang, X. Liu, Z. Chen, H. Y. Hong, J. Tang, D. Song. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 1225–1238, 2021.

DOI: 10.18653/v1/2021.emnlp-main.94.

[72] R. D. Wu, Y. Yao, X. Han, R. B. Xie, Z. Y. Liu, F. Lin, L. Y. Lin, M. S. Sun. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, ACL, Hong Kong, China, pp. 219–228, 2019. DOI: 10.18653/v1/D19-1021.

[73] Y. L. Shen, X. Y. Ma, Z. Q. Tan, S. Zhang, W. Wang, W. M. Lu. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, ACL, pp. 2782–2794, 2021. DOI: 10.18653/v1/2021.acl-long.216.

[74] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems* Lake Tahoe, USA, pp. 3111–3119, 2013.

[75] J. Pennington, R. Socher, C. D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ACL, Doha, USA, pp. 1532–1543, 2014. DOI: 10.3115/v1/D14-1162.

[76] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever. Improving Language Understanding by Generative Pre-Training, [Online], Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, 2021.

[77] R. Colin, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Q. Zhou, W. Li. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[78] W. Cui, X. Chen. Open rule induction. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

[79] J. W. Han, M. Kamber, J. Pei. *Data Mining: Concepts and Techniques*, 3rd ed., Berlin, Germany: Morgan Kaufmann Publishers, 2011.

[80] J. Leskovec, A. Rajaraman, J. D. Ullman. Mining of Massive Datasets, [Online], Available: http://infolab.stanford.edu/~ullman/mmds/book.pdf, 2021.

[81] J. W. Han, Y. Z. Sun, X. F. Yan, P. S. Yu. Mining knowledge from data: An information network analysis approach. In *Proceedings of the IEEE 28th International Conference on Data Engineering*, IEEE, Arlington, USA, pp. 1214–1217, 2012. DOI: 10.1109/ICDE.2012.145.

[82] P. N. Tan, M. Steinbach, A. Karpatne, V. Kumar. *Introduction to Data Mining*, 2nd ed., USA: Pearson, 2018.

[83] R. Agrawal, R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, pp. 487–499, 1994.

[84] J. W. Han, J. Pei, Y. W. Yin. Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, vol. 29, no. 2, pp. 1–12, 2000. DOI: 10.1145/335191.335372.

[85] M. J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372–390, 2000. DOI: 10.1109/69.846291.

[86] M. C. Liu, J. F. Qu. Mining high utility itemsets without candidate generation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, Association for Computing Machinery, Maui, USA, pp. 55–64, 2012. DOI: 10.1145/2396761.2396773.

[87] Z. H. Deng, S. L. Lv. PrePost$^+$: An efficient N-lists-based algorithm for mining frequent itemsets via Children-parent equivalence pruning. *Expert Systems with Applications*, vol. 42, no. 13, pp. 5424–5432, 2015. DOI: 10.1016/j.eswa.2015.03.004.

[88] J. F. Qu, B. Hang, Z. Wu, Z. B. Wu, Q. Gu, B. Tang. Efficient mining of frequent itemsets using only one dynamic prefix tree. *IEEE Access*, vol. 8, pp. 183722–183735, 2020. DOI: 10.1109/ACCESS.2020.3029302.

[89] R. Srikant, R. Agrawal. Mining generalized association rules. In *Proceedings of the 21th International Conference on Very Large Data Bases*, Zurich, Switzerland, pp. 407–419, 1995.

[90] J. W. Han, Y. J. Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, Zurich, Swizerland, pp. 420–431, 1995.

[91] J. W. Han, J. Pei, Y. W. Yin, R. Y. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, 2004. DOI: 10.1023/B:DAMI.0000005258.31418.83.

[92] R. Agrawal, R. Srikant. Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering*, IEEE, Taipei, China, pp. 3–14, 1995. DOI: 10.1109/ICDE.1995.380415.

[93] R. Srikant, R. Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, ACM, Montreal, Canada, pp. 1–12, 1996. DOI: 10.1145/233269.233311.

[94] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, Ed., Menlo Park, USA: American Association for Artificial Intelligence, pp. 307–328, 1996.

[95] T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In *Proceedings of ACM SIGMOD Conference on Management of Data*, Association for Computing Machinery, Montreal, Canada, pp. 13–23, 1996.

[96] B. Lent, A. Swami, J. Widom. Clustering association rules. In *Proceedings of the 13th International Conference on Data Engineering*, IEEE, Birmingham, UK, pp. 220–231, 1997. DOI: 10.1109/ICDE.1997.581756.

[97] K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama. Computing optimized rectilinear regions for association rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAI, Newport Beach, USA, pp. 96–103, 1997.

[98] M. Kamber, J. W. Han, J. Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAI,

Newport Beach, USA, pp. 207−210, 1997.

[99] Y. Aumann, Y. Lindell. A statistical theory for quantitative association rules. *Journal of Intelligent Information Systems*, vol. 20, no. 3, pp. 255–283, 2003. DOI: 10.1023/A:1022812808206.

[100] L. Q. Geng, H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, vol. 38, no. 3, Article number 9, 2006. DOI: 10.1145/1132960.1132963.

[101] J. Blanchard, F. Guillet, R. Gras, H. Briand. Using information-theoretic measures to assess association rule interestingness. In *Proceedings of the 5th IEEE International Conference on Data Mining*, IEEE, Houston, USA, pp. 66−73, 2005. DOI: 10.1109/ICDM.2005.149.

[102] J. W. Han, H. Cheng, D. Xin, X. F. Yan. Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007. DOI: 10.1007/s10618-006-0059-1.

[103] P. Fournier-Viger, J. C. W. Lin, B. Vo, T. T. Chi, J. Zhang, H. B. Le. A survey of itemset mining. *WIREs Data Mining and Knowledge Discovery*, vol. 7, no. 4, Article number e1207, 2017. DOI: 10.1002/widm.1207.

[104] C. C. Aggarwal, Y. Li, J. Y. Wang, J. Wang. Frequent pattern mining with uncertain data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Paris, France, pp. 29–38, 2009. DOI: 10.1145/1557019.1557030.

[105] W. S. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, P. S. Yu. HUOPM: High-utility occupancy pattern mining. *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 1195–1208, 2020. DOI: 10.1109/TCYB.2019.2896267.

[106] C. M. Chen, L. L. Chen, W. S. Gan, L. N. Qiu, W. P. Ding. Discovering high utility-occupancy patterns from uncertain data. *Information Sciences*, vol. 546, pp. 1208–1229, 2021. DOI: 10.1016/j.ins.2020.10.001.

[107] B. Vo, L. T. T. Nguyen, N. Bui, T. D. D. Nguyen, V. N. Huynh, T. P. Hong. An efficient method for mining closed potential high-utility itemsets. *IEEE Access*, vol. 8, pp. 31813–31822, 2020. DOI: 10.1109/ACCESS.2020.2974104.

[108] R. Andrews, J. Diederich, A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, vol. 8, no. 6, pp. 373–389, 1995. DOI: 10.1016/0950-7051(96)81920-4.

[109] V. I. S. Carmona. Experimental Analysis of Representation Learning Systems, Ph. D. dissertation. University College London, UK, 2018.

[110] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, vol. 51, no. 5, Article number 93, 2019. DOI: 10.1145/3236009.

[111] S. Thrun. Extracting rules from artificial neural networks with distributed representations. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, Denver, USA, pp. 505−512, 1994.

[112] M. W. Craven, J. W. Shavlik. Extracting tree-structured representations of trained networks. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, Denver, USA, pp. 24−30, 1995.

[113] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, vol. 51, no. 1, pp. 141–154, 2011. DOI: 10.1016/j.dss.2010.12.003.

[114] A. A. Freitas. Comprehensible classification models: A position paper. *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 1, pp. 1–10, 2013. DOI: 10.1145/2594473.2594475.

[115] H. Jacobsson. Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Computation*, vol. 17, no. 6, pp. 1223–1263, 2005. DOI: 10.1162/0899766053630350.

[116] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. *Classification and Regression Trees*, New York, USA: Wadsworth Int. Group, 1984.

[117] M. T. Ribeiro, S. Singh, C. Guestrin. "Why Should I Trust You?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, San Francisco, USA, pp. 1135−1144, 2016. DOI: 10.1145/2939672.2939778.

[118] P. Domingos. Knowledge discovery via multiple models. *Intelligent Data Analysis*, vol. 2, no. 3, pp. 187–202, 1998. DOI: 10.3233/IDA-1998-2303.

[119] I. Sánchez, T. Rocktaschel, S. Riedel, S. Singh. Towards extracting faithful and descriptive representations of latent variable models. In *Proceedings of AAAI Spring Syposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches*, AAAI, Stanford, California, USA, pp. 35−38, 2015.

[120] I. Sanchez Carmona, S. Riedel. Extracting interpretable models from matrix factorization models. In *Proceedings of International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches*, Montreal, Canada, pp. 78−84, 2015.

[121] G. Peake, J. Wang. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, London, United Kingdom, pp. 2060−2069, 2018. DOI: 10.1145/3219819.3220072.

[122] A. C. Gusmão, A. H. C. Correia, G. De Bona, F. G. Cozman. Interpreting embedding models of knowledge bases: A pedagogical approach. In *Proceedings of ICML Workshop on Human Interpretability in Machine Learning*, Stockholm, Sweden, pp. 79−86, 2018.

[123] S. B. Thrun. *Extracting Provably Correct Rules from Artificial Neural Networks*, Bonn, University of Bonn, Germany, 1993.

[124] J. R. Zilke, E. L. Mencía, F. Janssen. DeepRED – rule extraction from deep neural networks. In *Proceedings of the 19th International Conference on Discovery Science*, Springer, Bari, Italy, pp. 457−473, 2016. DOI: 10.1007/978-3-319-46307-0_29.

[125] M. Sato, H. Tsukimoto. Rule extraction from neural networks via decision tree induction. In *Proceedings of International Joint Conference on Neural Networks*, IEEE, Washington, USA, pp. 1870−1875, 2001. DOI: 10.1109/IJCNN.2001.938448.

[126] R. Setiono, H. Liu. Understanding neural networks via rule extraction. In *Proceedings of the 14th International*

Joint Conference on Artificial Intelligence, Montreal, Canada, pp. 480–485, 1995.

[127] B. S. Yang, W. T. Yih, X. D. He, J. F. Gao, L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA, 2015.

[128] W. J. Murdoch, A. Szlam. Automatic rule extraction from long short term memory networks. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[129] S. M. Lundberg, S. I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 4768–4777, 2017.

[130] S. Bang, P. T. Xie, H. Lee, W. Wu, E. Xing. Explaining a black-box by using a deep variational information bottleneck approach. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, AAAI, pp. 11396–11404, 2021.

[131] M. Pourvali, Y. C. Jin, C. Sheng, Y. Meng, L. Wang, M. S. Gorkovenko, C. J. Hu. Path-based visual explanation. In *Proceedings of the 9th CCF International Conference on Natural Language Processing and Chinese Computing*, Springer, Zhengzhou, China, pp. 454–466, 2020. DOI: 10.1007/978-3-030-60457-8_37.

[132] A. Jacovi, Y. Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 4198–4205, 2020. DOI: 10.18653/v1/2020.acl-main.386.

[133] J. Sippy, G. Bansal, D. S. Weld. Data staining: A method for comparing faithfulness of explainers. In *Proceedings of ICML Workshop on Human Interpretability in Machine Learning*, 2020.

[134] J. Bastings, S. Ebert, P. Zablotskaia, A. Sandholm, K. Filippova. "Will you find these shortcuts?" A protocol for evaluating the faithfulness of input salience methods for text classification, [Online], Available: https://arxiv.org/pdf/2111.07367.pdf, 2021.

[135] J. Yuan, O. Nov, E. Bertini. An exploration and validation of visual factors in understanding classification rule sets. In *Proceedings of IEEE Visualization Conference*, IEEE, New Orleans, USA, pp. 6–10, 2021. DOI: 10.1109/VIS49827.2021.9623303.

[136] I. Lage, E. Chen, J. He, M. Narayanan, B. Kim, S. J. Gershman, F. Doshi-Velez. Human evaluation of models built for interpretability. In *Proceedings of the 7th AAAI Conference on Human Computation and Crowdsourcing*, AAAI Press, Stevenson, USA, pp. 59–67, 2019.

[137] A. McCallum, D. Jensen. A Note on the Unification of Information Extraction and Data Mining using Conditional-Probability, Relational Models, [Online], Available: https://scholarworks.umass.edu/cs_faculty_pubs/42/, 2021.

[138] R. J. Mooney, R. Bunescu. Mining knowledge from text using information extraction. *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 1, pp. 3–10, 2005. DOI: 10.1145/1089815.1089817.

[139] Q. Z. Xie, X. Z. Ma, Z. H. Dai, E. Hovy. An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, Vancouver, Canada, pp. 950–962, 2017. DOI: 10.18653/v1/P17-1088.

[140] L. A. Galárraga, C. Teflioudi, K. Hose, F. Suchanek. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web*, Association for Computing Machinery, Rio de Janeiro, Brazil, pp. 413–422, 2013. DOI: 10.1145/2488388.2488425.

[141] S. Riedel, S. Singh, G. Bouchard, T. Rocktäschel, I. Sanchez. Towards two-way interaction with reading machines. In *Proceedings of the 3rd International Conference on Statistical Language and Speech Processing*, Springer, Budapest, Hungary, pp. 1–7, 2015. DOI: 10.1007/978-3-319-25789-1_1.

[142] K. A. Kaufman, R. S. Michalski. From data mining to knowledge mining. *Data Mining and Data Visualization*, C. R. Rao, E. J. Wegman, J. L. Solka, Eds., Amsterdam, Netherlands: Elsevier, pp. 47–75, 2005. DOI: 10.1016/S0169-7161(04)24002-0.

Yong Rui received the B.Sc. degree in electrical engineering from Southeast University, China in 1991, the M. Sc. degree in electrical engineering from Tsinghua University, China in 1994, and the Ph.D. degree in electrical and computer engineering from University of Illinois at Urbana-Champaign (UIUC), USA in 1999. He is currently the Chief Technology Officer and Senior Vice President of Lenovo Group, China. He is a Fellow of ACM, IEEE, IAPR, China SPIE, CCF and CAAI, and a Foreign Member of Academia Europaea. He holds 70 patents, and is the recipient of the prestigious 2018 ACM SIGMM Technical Achievement Award and 2016 IEEE Computer Society Edward J. McCluskey Technical Achievement Award.

His research interests include multimedia, artificial intelligence, big data and knowledge mining.

E-mail: yongrui@lenovo.com (Corresponding author)

Vicente Ivan Sanchez Carmona received the B.Eng. and M.Eng. degree in computer engineering from National Autonomous University of Mexico, Mexico in 2008 and 2011, and the Ph.D. degree in computer science from University College London, UK in 2018. He is currently a researcher in Lenovo's AI Lab, China. He has served as a reviewer in different conferences such as AAAI, ACL, CoNLL, COLING, among others.

His research interests include artificial intelligence, behavioral science, cognitive science and human-computer interaction.

E-mail: vcarmona@lenovo.com

Mohsen Pourvali received the Ph.D. degree in computer science from Ca′ Foscari University of Venice, ltaly in 2017. During the Ph.D. period, he was working on text summarization and document enrichment. Currently, he is an advisory researcher at AI Lab in Lenovo. He is an experienced lecturer with a demonstrated history of teaching in universities.

His research interests include explainable artificial intelligence and knowledge graph, especially in domain adaptive information extraction.

E-mail: mpourvali@lenovo.com

ORCID iD: 0000-0003-2653-9613

Yun Xing received the B.Sc. degree in optical information science and technology from Beijing Institute of Technology, China in 2012, and the M.Eng. degree in electronics and optics from Polytech Orleans, France in 2016. Currently, he is a NLP researcher in AI Lab at Lenovo Research, China.

His research interests include natural language processing in machine learning and deep learning.

E-mail: xingyun44@hotmail.com

Wei-Wen Yi received the B. Eng. and M. Eng. degrees in information and communication engineering from Beijing University of Posts and Telecommunications, China in 2017 and 2020, respectively. Currently, she is a natural language processing researcher at Lenovo Research, China. She received the Best Paper Award of the EAI International Conference on Communications and Networking in China, China in 2018. She is a member of EAI and IEEE.

Her research interests include named entity recognition, relation extraction and entity linking.

E-mail: yiww1@lenovo.com

Hui-Bin Ruan received the M.Sc. degrees in computer technology from Soochow University, China in 2020. Currently, she is a researcher in natural language process in Lenovo, China.

Her research interests include discourse parsing, text classification and entity linking.

E-mail: ruanhb2@lenovo.com

Yu Zhang received the B.Eng. degree in human factors, B.Sc. (Minor) degree in applied mathematics and M.Sc. degree in engineering physics from Beihang University, China in 2008, 2008 and 2011. He is currently the technical assistant to chief technology officer at Lenovo, China, and a Ph.D. degree candidate in computer science at Southeast University, China.

His research interests include human computer interaction and human-centered AI.

E-mail: zhangyu29@lenovo.com