

Article

A Comprehensive Survey of Cognitive Graphs: Techniques, Applications, Challenges

Meiling Chen ¹ , Ye Tian ² , Zhaorui Wang ³ , Hong Xu ¹  and Bo Jiang ^{1,*} 

¹ East China Institute of Computing Technology, Shanghai, 201808, China; cml13262283846@163.com(M.C.); frankxuh@126.com(H.X.)

² MeiTuan, Beijing 100102, China; peterytian@gmail.com

³ University of Shanghai for Science and Technology, Shanghai 200093, China; wzhr200806@163.com

* Correspondence: jiangbo@ecict.com

Abstract: The realization of the third-generation artificial intelligence (AI) requires the evolution from perceptual intelligence to cognitive intelligence, where knowledge graphs may not meet the practical needs anymore. Based on the dual channel theory, cognitive graphs are established and developed through coordinating the implicit extraction module and the explicit reasoning module as well as integrating knowledge graphs, cognitive reasoning and logical expressions, which have achieved successes in multi-hop question answering. It is desired for cognitive graphs to be widely used in advanced AI applications such as large-scale knowledge representations and intelligent responses, promoting the development of AI dramatically. This review discusses cognitive graphs systematically and elaborately, including basic concepts, generations, theories and technologies. Moreover, we try to predict the development of cognitive intelligence in the short-term future and further enlighten more researches and studies.

Keywords: Cognitive Graph; Knowledge Graph; Knowledge Reasoning; Natural Language Generating.

1. Introduction

The previous few decades have witnessed the dramatic development of artificial intelligence (AI). Broadly speaking, there have been three major stages during the evolution of AI [1], as can be seen in Figure 1.

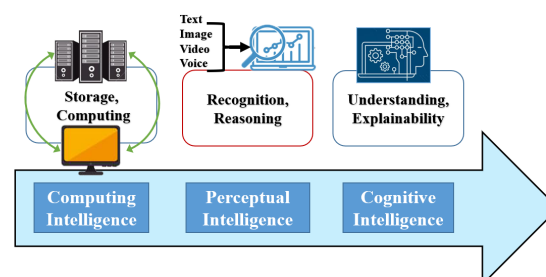


Figure 1. The development stages of AI.

The first stage is computational intelligence, which is owing to the fast computing and mass storage capacities of computers. With the maturity of technologies such as grid computing, distributed storage and quantum storage, the computing power of machines has far exceeded that of human beings and laid a solid foundation for the next stages. The second stage is perceptual intelligence, which is the current stage of AI. Perceptual

intelligence understands the world through auditory, visual and tactile sensing systems, so that machines can acquire abilities of "listening", "speaking" and "watching" [2–12].

Perceptual intelligence is mainly concerned with two schools, namely data-driven AI represented by deep learning [13] and knowledge-driven AI represented by knowledge graphs. With the support of big data, computing power and advanced algorithms, data-driven AI has approached or even surpassed humans in many tasks. For example, the error rate of image recognition on ImageNet has become lower than that of humans since 2015 [14]. The speech recognition in a single Chinese sentence also achieves a lower error rate [15]. However, the effect of neural networks heavily depends on the training data, where massive high-quality annotated data are usually required. Furthermore, the inference results are lack of explainability [16], poor in robustness [17,18], and easy to be disturbed [19–21]. The model is usually oriented to a specific task and difficult to migrate for different tasks. On the contrast, the symbolic knowledge-driven AI represents the relationships between entities through symbol transformation. Different from deep learning, knowledge graphs are highly interpretable and widely applicable to different tasks [22–27]. However, the incompleteness of knowledge graphs results in low accuracy of logical inference rules learned from sparse data, especially when the order of inference rules is increased. Symbol-based knowledge reasoning methods mostly use logic rules written manually or learned from the existing knowledge base. It is difficult for the manual method of writing logic rules to meet the demands of large-scale knowledge reasoning, with low reasoning coverage and low reasoning efficiency. Therefore, knowledge graphs face two fatal defects, which are high manually-built cost and low automatic-built precision. CYC [28] is one of the earliest knowledge projects, with each knowledge content cost \$5.71 manually. On the other hand, although another knowledge project NELL [29] applies machine learning to realize automatic knowledge graph building, its error rate is increased to 10 times correspondingly. Therefore, the two notable projects are at a standstill. Characteristics of data-driven AI and knowledge-driven AI are shown in Table 1.

Table 1: The characteristics of data-driven AI and knowledge-driven AI.

Ability performance	AI driven by data	AI driven by knowledge
Technology	Deep learning	Knowledge graph
Explainability	Weak	Strong
Robustness	Weak	Strong
Generalization ability	Weak	Strong
Input data requirements	Massive labeled-training data	Structured, noiseless data
Application scope	Specific tasks, difficult to migrate	Widely applicable to different tasks

To solve the problems mentioned above, some researchers proposed the concept of cognitive intelligence, which is considered as the third stage of AI. Bo Zhang unveiled a theoretical framework of the third-generation AI in 2018. The core goal of cognitive intelligence is to build up explainable and robust AI theories and methods, with secure, reliable, trusted and scalable AI technologies developed. To achieve this goal, one possible way is to combine knowledge-driven AI with data-driven AI and fully take advantages of both. As shown in Table 1, data-driven AI and knowledge-driven AI have apparent complementary relationships in inputs, outputs and functions. This situation is similar to that based on the dual-channel theory in human cognitive science.

Inspired by the dual-process theory in cognitive science, Ding et al. [30] built a cognitive graph iteratively by coordinating an implicit extraction module (System 1) and an explicit reasoning module (System 2). System 1 is conducted by an efficient knowledge-driven model called Bidirectional Encoder Representations from Transformers(BERT) [31], which can extract question-relevant entities and organize them into a cognitive graph. System 2 is processed by a data-driven neural network called Graph Neural Network (GNN) [32], which can carry out the reasoning procedure over the graph. The implementation based on BERT and GNN can not only reach accurate answers, but also provide explainable reasoning paths. The state-of-art results on the Hotpot QA dataset show the efficacy of cognitive graphs and indicate it is promising to develop cognitive graphs with the fusion of knowledge-driven AI and data-driven AI.

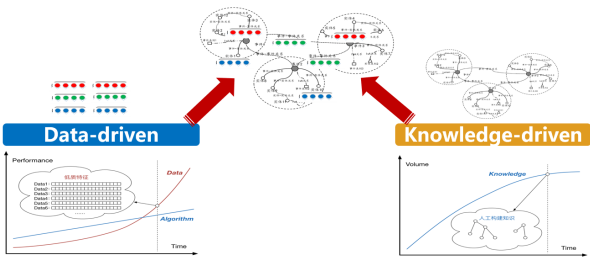


Figure 2. The foundation of cognitive graphs.

This paper aims to present a comprehensive overview of the cognitive graphs, to provide researchers with synthesis and pointers to relevant researches, and to introduce the field to researchers who are less familiar with cognitive graphs. In this paper, we systematically and elaborately review information about cognitive graphs. We construct this paper in a top-down structure, as shown in Figure 3. The first part mainly introduces the background and basic concepts of cognitive graphs. The second part mainly introduces the critical technology road maps of cognitive graphs, including knowledge graphs, knowledge reasoning and natural language generating. Lastly, we identify the challenges and point out the most promising technical research directions of cognitive graphs.

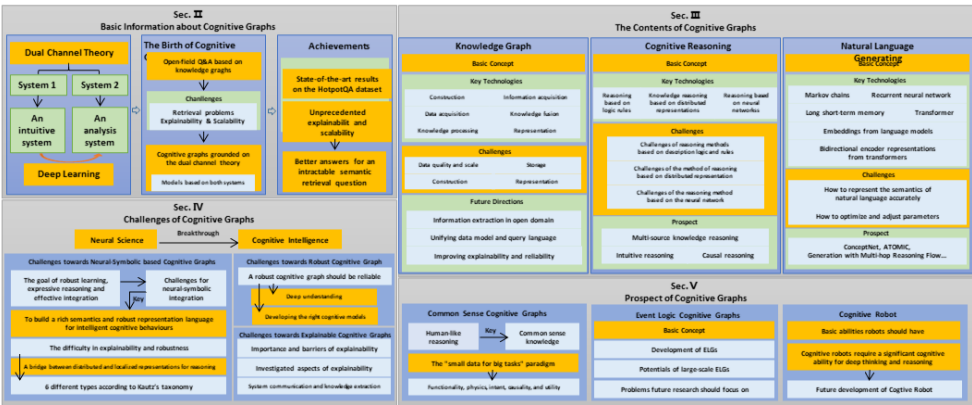


Figure 3. The overall structure of the paper.

2. Basic Information about Cognitive Graphs

2.1. Dual Channel Theory

The basic idea of cognitive graphs derives from the dual-channel theory in cognitive science, which classifies the cognitive systems of human brains according to functions of System 1 and System 2. As shown in Figure 4, System 1 is an intuitive system, which aims to find the answer through an intuitive matching of relevant information. So it

can only make intuitive and straightforward decisions, which actually fails to handle complex reasoning process. On the contrast, as an analysis system which finds answers through logic reasoning, System 2 improves the ability to handle complex reasoning.

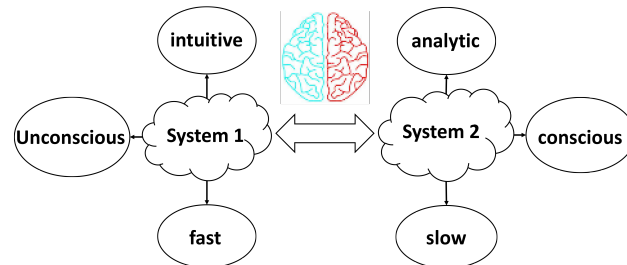


Figure 4. The cognitive system of human brains: System 1 and System 2.

Similar to the situation of System 1, current deep learning technologies perform well in the intuitive, fast, unconscious and habitual system, while perform poorly in the slow, logical, sequential, conscious system. Yoshua Bengio, ACM Turing Award winner, issued a statement in the special report of NeurIPS 2019 [33], that deep learning needs to develop from System 1 to System 2. How to overcome the limitations of previous deep learning and realize the cognitive intelligence is an urgent problem. Fortunately, the cognitive graph which follows the dual-channel theory in cognitive psychology is able to unify System 1 and System 2 and take full advantages of both, overcoming the defects of the deep learning system and knowledge graphs.

2.2. The Birth of Cognitive Graph

Ding et al. [30] first proposed the cognitive graph and leveraged it to deal with question&answer (Q&A) of reading comprehensions in the multi-hop field, where a classical question is "Who is the director of the 2003 film which has scenes in it filmed at the Quality Cafe in Los Angeles?". In traditional methods, open-field Q&A mainly relies on a large-scale knowledge graph. Previous work DrQA [34] proposed a straightforward framework to solve open-domain Q&A. The researchers first retrieved 5 most similar documents using a retriever model, and then turned the question into single-paragraph Q&A. However, in multi-hop questions, this method suffers from "short-sighted retrieval". This means that the relevance between the text of last few jumps and the question is very low, which is actually difficult to be directly retrieved, resulting in a poor effect. In addition to retrieval problems, there are also two challenges lying ahead, which are explainability and scalability.

Grounded on the dual process theory, an ideal cognitive graph can contribute to all the three challenges significantly. It is an iterative framework to build the cognitive graph step by step. As for the example of "Who is the director of the 2003 film which has scenes in it filmed at the Quality Cafe in Los Angeles?", the overview procedure of the cognitive graph is shown in Figure 5.

Models based on System 1 extract question-related entities from paragraphs to build the cognitive graph and generate semantic vectors for each node. Then the relevant paragraphs about new extracted entities are retrieved or just indexed from Wikipedia. Meanwhile, models based on System 2 carry out reasoning based on semantic vectors and compute clues to guide the extraction of System 1. After several iterations, System 2 selects a node as the predicted answer based on the reasoning results. Figure 6 shows the detailed procedure of cognitive graph.

System 1 and System 2 can be established by various types of models. Since the cognitive graph is initialized with entities extracted from questions, it is crucial to seek out a powerful module to extract useful entities and generate semantic vectors for each node. Recently, BERT [31] has been proved to be a successful language representation model. Therefore, BERT is designed to serve as System 1. The input of System 1 consists of three parts, including the question, the "clue" found in the previous paragraph and

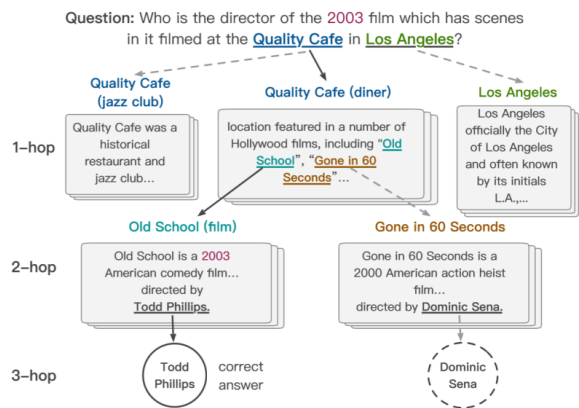


Figure 5. An example of the cognitive graph for multi-hop Q&A. Each hop node corresponds to an entity, followed by its introductory paragraph. The circles mean answer nodes, representing answer candidates to the question. The solid black edges are the correct reasoning paths [30].

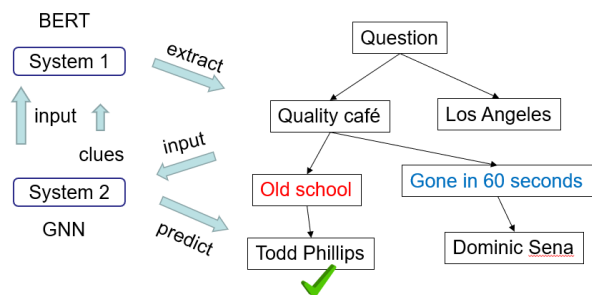


Figure 6. Procedure of cognitive graphs. System 1 extracts question-related entities from paragraphs and generates semantic vectors for each node. System 2 carries out reasoning based on semantic vectors and computes clues to guide the extraction of System 1. After several iterations, System 2 selects a node as the predicted answer based on the reasoning results [30].

the Wikipedia document about an entity x (for example, x is the movie “Old School”). The goal of System 1 is to extract the "next hop entity name" and "answer candidate" in the document. For example, as shown in Figure 5, from the "quality café" paragraph, "old school" and "gone in 60 seconds" are extracted as the entity names of the next jump. These extracted entities and answer candidates will be added to the cognitive graph as nodes. In addition, System 1 will calculate the semantic vector of current entity x , which will be used as the initial value of relational reasoning in System 2. Owing to the inductive bias of graph structure, GNN has presented remarkable performances on relational reasoning. Correspondingly a variant of GNN is designed to serve as System 2. GNN takes an semantic vector from System 1 as the initial hidden representation for each entity. At each step, hidden representations X for nodes are updated according to the propagation rules. A two-layer multilayer perceptron serves as the predictor, which predicts the final answer based on hidden representations X . To identify the correct answer node from the cognitive graph, the model in System 2 constructs a training sample for the reasoning task. Each training sample is the union of all correct reasoning paths and negative nodes. The losses of answer node prediction can not only optimize predictors and System 2, but also fine-tune System 1 through semantic vectors. The overview of cognitive graph implementation is illustrated in Figure 7.

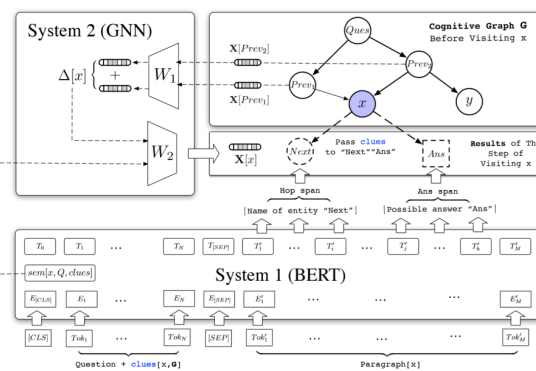


Figure 7. Overview of the cognitive graph implementation. When visiting the node x , System 1 generates new hop and answer nodes based on the $clues[x, G]$ discovered by System 2. It also creates the initial representation $sem[x, Q, clues]$, based on which the GNN in System 2 updates the hidden representations $X[x]$ [30].

2.3. The Advantages of Cognitive Graphs

The cognitive graph achieves state-of-art results on the HotpotQA dataset with outstanding performances on all the evaluation metrics, reaching a winning joint F1 score of 34.9, obviously higher when compared to 23.6 of the best competitor. More critically, the cognitive graph not only elevates the reasoning ability by solving the intractable short-sighted retrieval problem, but also attains unprecedented explainability and scalability. All the excellent performances mainly attribute to the superiority of the dual-channel-based cognitive graph framework over traditional retrieval-extraction methods. By iteratively expanding with clues and adding 1-hop entities spotted in the question, the cognitive graph improves the retrieval of multiple hops away from the question, which is often hard to be backtracked in traditional retrieval-extraction frameworks. Moreover, since the only operation purpose referred to all paragraphs is to acquire some specific paragraphs by their title indexes, the consumption of time does not have a significant growth along with the number of paragraphs. As a result, the cognitive graph can scale in nature. Benefiting from the explicit reasoning paths and collaborative reasoning processes, the cognitive graph enjoys marvelous explainability. Figure 8 shows its superiority in explainability.

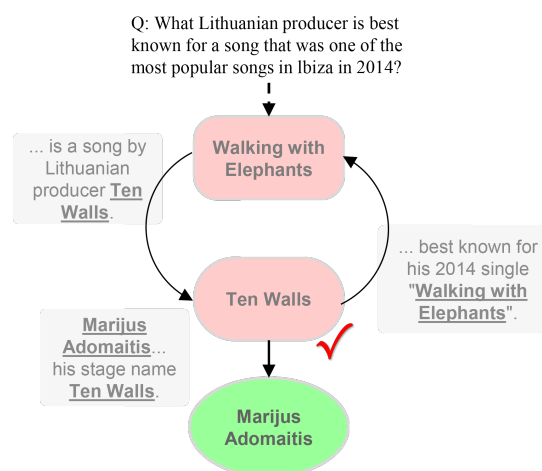


Figure 8. Case Study. Circles are candidate answer nodes while rounded rectangles are hop nodes. Green circles are the final answers given by the cognitive graph and check marks represent the annotated ground truth [30].

As shown in Figure 8, for an intractable semantic retrieval question without any entity mentioned, the cognitive graph finally gets the answer “Marijus Adomaitis”,

while the annotated standard answer is “Ten Walls”. However, considering that “Ten Walls” is the stage name of Marijus Adomaitis, the cognitive graph yields a more accurate result. When further tracing back to the reasoning process in the cognitive graph, it turns out that the model has already reached “Ten Walls” and finally answers with his real name, which seems to be more reasonable. Such explainable advantages are not enjoyed by black-box models. This explainability brings benefits to the users, that traditional black-box models do not have.

3. The Contents of Cognitive Graphs

Cognitive intelligence can reason the real world flexibly and dynamically, and transfer what it learn with the help of synthesized knowledge based on a large amount of data. It is well known that human cognition can successfully integrate the connectionist (brain-inspired) and symbolic (mind-inspired) paradigms, where the language is a compelling case in point. To build an intelligent cognitive graph, it is urgent and indispensable to develop a framework that can routinely acquire, represent, and manipulate knowledge, simultaneously using the knowledge in the service of reasoning logically like humans. Thus, as shown in Figure 9, three core technical supports are needed as prerequisites for building cognitive graphs: 1) large-scale knowledge graphs to support intuitive knowledge expansion; 2) reasoning mechanisms to conduct complex reasoning and make analytic decisions; 3) large-scale pre-trained natural language generating models to explain the inference process and express the reasoning results in a human-friendly way.

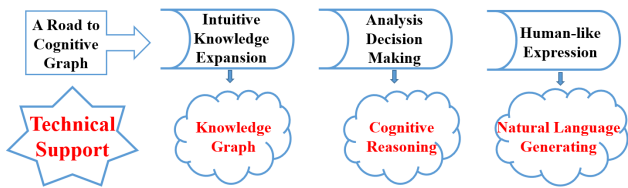


Figure 9. The Cornerstone of Cognitive Graph. On the way to build cognitive graphs, three core technical supports are needed as prerequisites for building a cognitive graph: large-scale knowledge graphs to support intuitive knowledge expansion; reasoning mechanisms to conduct complex reasoning and make analytic decisions; large-scale pre-trained natural language generating models to explain the inference process and express the reasoning results in a human-friendly way.

3.1. Knowledge Graph

The knowledge graph is regarded as an important cornerstone in the transformation from perceptual intelligence to cognitive intelligence. Edward Feigenbaum [35], Turing Award winner, the founder of knowledge engineering, once asserted that knowledge is the power in AI systems. Building large-scale and high-quality knowledge graphs is an essential link in the process of spanning from perceptual intelligence to cognitive intelligence.

3.1.1. Basic Concept of Knowledge Graph

A knowledge graph, also known as a semantic network, represents a network of real-world entities and illustrates the relationship between objects, events, situations and so on. The information is usually stored in a graph database and visualized as a graph structure, prompting the term knowledge graph. The knowledge graph is a multi-relational graph composed of entities (nodes) and relations (different types of edges). Each relational fact is represented as a triple of the form (head entity, relation, tail entity), indicating that two entities are connected by a specific relation. In essence, a

knowledge graph is a semantic network that reveals the relationship between entities, which can formally describe the real world and different relationships.

3.1.2. Key Technologies of Knowledge Graph

The techniques applied in knowledge graph building mainly include knowledge graph construction and knowledge graph representation.

The overview framework of knowledge graph construction is shown in Figure 10. As can be seen, the whole framework mainly consists of four parts, which are data acquisition, information acquisition, knowledge fusion and knowledge processing respectively.

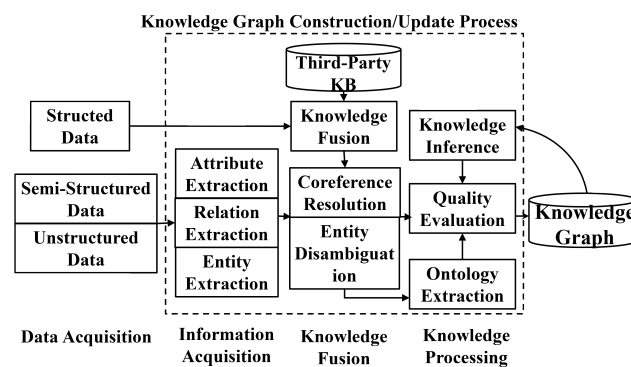


Figure 10. Overview of knowledge graph construction framework. The whole framework mainly consists of four parts: data acquisition, information acquisition, knowledge fusion, and knowledge processing [36].

Data Acquisition

Data Acquisition is the cornerstone of knowledge graph, whose goal is to extract structured data from unstructured or semi-structured data. Then the knowledge graph is built up based on these data. This step is mainly carried out with a series of automatic or semi-automatic techniques, where knowledge (actually a heap of entity relationships) is extracted from raw data and stored into the knowledge base.

Information Acquisition

Information acquisition aims to extract entity attributes and their interrelations from various data sources. This task can be divided into three categories: entity extraction, relation extraction and attribute extraction [37,38].

Entity extraction [39], also known as Named Entity Recognition (NER), is created to make automatic identification of proper nouns (organization name, place name, person name, time, etc.) or meaningful noun phrases from text corpus. It is the basis of knowledge acquisition. The accuracy of entity extraction directly affects the quality and efficiency of knowledge acquisition.

Attribute extraction [40] aims to achieve a complete description of entities. The attributes of entities can be regarded as a nominal relationship between entities and attribute values. Therefore, attribute extraction tasks can be transformed into relation extraction tasks.

Relation extraction [41] is applied to solve the problem of semantic links between entities. The relationship extraction is mainly to identify entity relations through the artificial construction of semantic rules and templates. Subsequently, the relationship model between entities gradually replaces the manual predefined syntax and rules.

Knowledge Fusion

After acquiring new knowledge, it is necessary to integrate it to remove contradictions and ambiguities, improve knowledge quality.

Knowledge fusion [42] was supposed to acquire multi-level, machine-processable and more complete innovative knowledge, and simultaneously remove uncertain and conflicting knowledge. So knowledge fusion is considered as an advanced stage of

knowledge integration. When multiple sources bring incomplete and inconsistent information, knowledge fusion is applied to produce new information (knowledge) that is complete and accurate, thus improve knowledge consistency and knowledge quality.

Knowledge Processing

For new fused knowledge, qualified parts can be added to the knowledge base to ensure the quality of the knowledge base after quality assessment. Knowledge processing mainly consist of ontology construction and quality evaluation.

a) Ontology Construction

Ontology is the semantic basis of communication between different subjects in the same domain, which has a tree structure. The most important feature of ontology is that it is shared, and the knowledge appearing in ontology is a kind of well-defined consensus. Ontology can be constructed by manual editing, with the help of ontology editing software. Or it can be automatically constructed by data-driven method, and then modified and confirmed by combining algorithm evaluation. The main research work of current ontology generating methods mainly focuses on the entity clustering method [43,44]. The main challenge is that the entity descriptions obtained by information extraction are very few and necessary context information is absent, which leads to the unavailability of most statistical models [45,46].

b) Quality Evaluation

The task of quality assessment of knowledge base is usually carried out together with the entity alignment task. Its significance is that the credibility of knowledge can be quantified, and the quality of knowledge can be effectively guaranteed by retaining the higher reliability and abandoning the lower confidence.

Knowledge Graph Representation

The knowledge graph representation is also called knowledge graph embedding. The key idea is to embed entities and relationships into a low-dimensional continuous vector space, so as to simplify operation and retain the inherent structure of the knowledge graph. A classical knowledge graph embedding technique is usually made up of three steps: (i) representing entities and relations, (ii) defining a scoring function, (iii) learning entity and relation representations. The existing knowledge graph embedding technologies are mainly divided into two categories: translational distance models and semantic matching models [47].

a) Translational distance model

In a translational distance model, the knowledge is well-preserved via capturing and measuring monolingual relations among entities with a distance-based scoring function.

In 2013, Mikolov et al. proposed Word2vec [48], a word representation model, and they discovered the translation-invariant phenomenon in the word representation space. Inspired by the translation-invariant phenomenon [48], Antoine Bordes et al. proposed TransE [49] to embed knowledge graphs into low-dimensional space via relational translation, which models the entity & relation representation as well as the monolingual relations. Since the TransE model has fewer parameters, the computational complexity is significantly reduced, and it also has better performances on the large-scale sparse knowledge graph. However, the TransE model has trouble dealing with complex relationships (such as 1-to-N, N-to-1, N-to-N). In response to the above problems of TransE model, an effective strategy called TransH was proposed [50]. It allows an entity to have distinct representations under different relationships by introducing the hyper-plane. Both TransE and TransH assume that entities and relationships are vectors in the same semantic space, so that similar entities can be located at similar positions in the space. However, each entity can have many aspects, while different relationships focus on different aspects of the entity. Therefore, the TransR model [51] establishes separate relationship spaces for different relationships. The TransR model is more complex than the TransE and TransH models because it introduces a projection matrix for each relationship. Thus, TransR is computationally intensive. The TransD model

[52] has made improvements to solving the above problems. It represents each entity or relationship with two vectors, one for semantics and another for building a mapping matrix. The process of mapping matrix and space transformation can also be replaced by vector multiplication, thus avoiding time-consuming matrix multiplication. Due to the uncertainty of semantics of entities and relationships in the knowledge graph, which has been ignored in previous research, TransG [53] proposed a KG2E model, using the Gaussian distribution to represent entities and relationships. The covariance of the Gaussian distribution indicates the uncertainty of the entity or relationship. TransG also models Gaussian entities, that is, TransG applies the Gaussian mixture model to describe the relationship between head and tail entities. By considering different semantics of relation, the TransG model could form multiple Gaussian distributions to distinguish correct and incorrect entities.

b) Semantic matching model

Semantic matching models exploit similarity-based scoring functions to measure plausibility of facts by matching latent semantics of entities and relations contained in their vector space representations.

One of the most representative semantic matching models is RESCAL [54]. In this model, entities are represented as vectors and relationships are represented as matrices. This relational matrix is applied to model the paired interaction between potential factors, and the scoring function is defined by a bilinear function. DistMult [55] simplifies RESCAL by restricting the matrix associated with the relation to diagonal matrices. However, the over-simplified model can only deal with symmetric relationships, which is obviously not fully applicable to general knowledge graphs.

By taking advantage of the expressiveness of RESCAL and the simplicity of DistMult, the Holographic Embeddings (Hole) model [56] represents both entities and relations as vectors. Therefore, it can model asymmetric relations as RESCAL does but is more efficient than RESCAL.

Inspired by the success of deep learning, some models attempt to integrate deep architectures into knowledge embedding, such as Semantic Matching Energy (SME) [57], Multilayer perceptron (MLP) [58], Neural Tensor Network (NTN) [59], and Neural Association Model (NAM) [60].

3.1.3. Challenges of Knowledge Graph

Although some neural networks have achieved impressive performance in knowledge graph representation and construction, they still have limitations in terms of transparency and explainability. Considering the basic requirements of knowledge graph based cognitive intelligence, we summarize the fundamental problems and corresponding challenges as follows:

A) Challenges of data quality and scale

Massive amounts of data are generated every day in the era of big data. Crawling, downloading, and indexing these web-page data requires a lot of storage and network resources. Traditional information extraction algorithms face the bottleneck of efficiency and it is difficult to cope with the dynamic changes of large-scale data. Meanwhile, web data also contain a lot of noise data, such as inconsistent data formats, diverse data types, and fuzzy naming, which bring great challenges to the information extraction technology.

B) Challenges of knowledge graph representation

The knowledge graph is a graphical representation of knowledge. It is necessary to establish a unified semantic space for knowledge so that the semantics can be calculated, so as to conduct prediction, reasoning and recommendation. Therefore, it has remained an urgent requirement to design more powerful knowledge graph representation learning algorithms to incorporate more ontology features and analyze the relationship between representation learning and ontology reasoning.

C) Challenges of knowledge graph construction

Building a knowledge graph requires extracting knowledge of entities, relationships, and attributes from different types of data. The difficulty of extracting knowledge is up to the type of data. For example, text data involve a large amount of natural language understanding issues. Image data involve a large amount of vision recognition-related issues. Voice data involve a large amount of voice recognition-related issues. Multimodal noisy perceptual data brings burdensome challenges to the construction of knowledge graph because more cross-modal relations bring accompanying cross-modal disambiguation.

D) Challenges of knowledge graph storage

The current knowledge graphs are usually based on multiple data models, such as resource description framework (RDF) graphs and attribute graphs. Although the RDF graph model has a stronger expression ability than the attribute graph model, its excessive theoretical information affects its promotion in the industry. The attribute graph, in contrast, is not perfect in terms of its theoretical foundation. So far, it has not formed a uniformly recognized strict mathematical definition. Although the attribute graph has been applied in multiple graph databases, it has not yet formed an industry standard. In addition, for different data models, there are different query languages in different graph databases, including SPARQL, Cypher, PGOL and so on. The inconsistency of data models and query languages not only increases the cost of database development and maintenance but also brings difficulties to users' learning.

3.1.4. Future Directions of Knowledge Graph

New challenges in knowledge graphs (e.g., knowledge representation, construction, and storage) also provide potential opportunities, and we explore a series of future directions as follows:

Information Extraction in Open Domain

In the stage of knowledge graph construction, information extraction is an important part. Current relation [61] extraction algorithms are mainly divided into two classes: open-domain and closed-domain. While closed-domain algorithms mainly apply supervised methods [62,63] or weakly supervised methods [64,65] to identify a finite and fixed set of relations, the open-domain algorithms explore relations without predefined schema, thus resulting in a series of challenges. Corpus heterogeneity is an evident obstacle for practicable tools. In order to expand the scale and coverage of the knowledge graph, researchers have paid attentions to information extraction methods for open fields. Automatically extracting knowledge from unlimited data resources is a potential hopeful way to build large-scale knowledge graphs, and it is also the underlying support for other high-level technologies such as knowledge reasoning. Therefore, open-domain information extraction technology is the hot-spot of future researches in the construction of knowledge graphs.

Unifying Data Model and Query Language

In terms of knowledge graph storage, the data model and query language are not unified yet. The unified data model and query language not only reduce the research and development cost of the database management systems, but also reduce the cost of users designing, building, managing and maintaining the database, and reduce the learning difficulty for new users. Therefore, for a unified knowledge graph data model, the development of a unified knowledge graph query language, and the definition of precise syntax and semantics are important research directions for the future of knowledge graph storage.

Improving Explainability and Reliability

Although some neural networks have achieved impressive performance in knowledge graph representation and construction, they still have limitations in terms of transparency and explainability. Some methods try to combine the black-box neural network models with symbolic reasoning, and increase explainability by introducing logical rules. Because only the realization of explainability can convince human to

believe in the prediction results, researchers need to do more in terms of explainability and improving the credibility of prediction results.

3.2. Cognitive Reasoning

Cognitive reasoning is an essential part of cognitive intelligence. Combined with the reasoning process of human brains, cognitive reasoning can further solve the complex problem of reading comprehension and the inference problem of knowledge graph with fewer samples, cooperative structured reasoning process and unstructured semantic understanding. Cognitive reasoning can help machines acquire multimode perception abilities. Knowledge graphs provide strong supports for cognitive reasoning. The bottom realization of cognitive reasoning is knowledge reasoning. Therefore, from the perspective of knowledge reasoning based on knowledge graphs, this section introduces the concept, development process, key technologies, problems, challenges and future research directions.

3.2.1. Basic Concept of Reasoning

The basic concept of reasoning is generally defined as a process in which people analyze, synthesize and make decisions on various things [66]. Kompridis [67] points out that knowledge reasoning is to obtain new knowledge from existing knowledge based on specific rules and constraints. Tari [68] defines reasoning as a general term for a series of capabilities, including the abilities to consciously understand things, to establish and verify facts, and to infer new knowledge based on the existing facts and logic rules.

3.2.2. Key Technologies of Reasoning

Early traditional reasoning methods are divided into deductive reasoning, inductive reasoning and default reasoning. Traditional knowledge reasoning is mainly based on logic and rules, such as first-order logic and predicate logic [69].

With the explosive growth of Internet data scale, traditional methods cannot adapt to the need to mine a large amount of knowledge in the era of big data. Recently, reasoning over knowledge graphs has become a hot research topic since it can obtain new knowledge and conclusions from existing data. With the emergence of distributed representations and neural network technologies, reasoning over knowledge graphs has attracted extensive attention. The reasoning methods are divided into three categories: rule-based reasoning, distributed representation-based reasoning and neural network-based reasoning.

Reasoning based on logic rules

Reasoning methods based on logic rules include logic reasoning, ontology reasoning and random walk reasoning. In the early stage, first-order predicate logic rules are exploited for reasoning, with typical high-accurate knowledge graph-oriented applications such as First-Order Inductive Learner (FOIL) [70]. Nevertheless, this universal protocol is suppressed by the complexity and diversity of large-scale knowledge graphs. On the contrast, balanced protocols such as the description logic [71], a decidable subset of predicate logical rules, are found to be more suitable. Subsequently, the well-defined semantics and powerful reasoning tools lay a crucial foundation for ontology reasoning.

Classical ontology reasoning methods could be based on tableaux, logical programming, first-order queries, production rules and so on. For instance, the tableaux-based method [72] only supports reasoning with predefined ontology axioms. In contrast, the Datalog-oriented rewriting method based on logical programming can meet most user-defined reasoning needs. As supplementary, another rewriting method based on first-order queries, which is associated with different query languages, can efficiently combine data sources with different data formats.

In 2001, a unique algorithm based on random walk reasoning, named Path Ranking Algorithm (PRA) [73], was introduced by Lao et al. to deal with related reasoning issues.

PRA is a typical technique for performing reasoning in a graph. To learn the inference model for a particular edge type in a knowledge base, PRA finds sequences of edge types that link nodes and then uses those types as features in a logistic regression model to predict missing edges in the graph. PRA dramatically improves the efficiency and robustness by replacing exhaustive searches with random walks and using unique paths as features in a per-target-relation binary classifier.

Reasoning based on distributed representations

Owing to the development of embedding technology, knowledge reasoning based on distributed representations has made great progress, using tensor decomposition, distance and semantic matching models. Its motivation is to project the entity and relation in the triple facts (head entity, relation and tail entity) into a low-dimensional vector form and perform inference predictions based on vector representations.

The tensor decomposition model treats the fact triple of knowledge graph as an element to construct tensors, where unknown knowledge is inferred by tensor decomposition. In this process, high-dimensional arrays are decomposed into multiple low-dimensional matrices. The RESCAL model [54] is a typical tensor decomposition model, which can reduce the data dimension and retain characteristics of the original data by decomposing multi-relational and high-dimensional data into a third-order tensor. Advanced similar models include TRESKAL [74], PRESKAL [75] and a novel model combining matrix decomposition with tensor decomposition.

The purpose of the distance model is to design a score function based on the transfer hypothesis to measure the validity score of the fact triple, where a higher score means better validity. TransE is one of the most widely-used distance models, in which potential feature representations are simply and efficiently translated by a relation-specific offset. However, due to the disability to quickly deal with 1-to-N, N-to-1 and N-to-N relations, TransE has been replaced by TransH in these cases. The latter realizes those relation treatments through introducing a relation-specific hyperplane.

The original semantic matching model uses two separate matrices to project head and tail entities for each relationship, which actually cannot effectively represent the essential relationship. In order to solve this problem, a novel method, named Semantic Matching Energy (SME) [76], is put forward with both linear and bilinear forms of semantic matching energy functions well defined.

Reasoning based on neural networks

By imitating the human brain for perception and cognition, neural networks have a great ability to extract features, making it potential to knowledge graphs (as well as cognitive graphs) reasoning. Reasoning based on neural networks can be classified according to detailed models such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Reinforcement Learning (RL).

At the early stage, a single layer model based on CNN is proposed to connect the entity vectors implicitly, which is soon replaced by an advanced model named Neural Tensor Network (NTN). With the assistance of a bilinear tensor layer, NTN can improve the reasoning performance efficiently. Other models based on CNN include ProjE [77], DKRL [78] and so on.

As for models based on RNN, a new approach to reason about conjunctions of multi-hop relations is proposed by Rajarshi Das et al. , which is called Path RNN [79]. In Path RNN, a path-sorting algorithm is exploited to find a different path for each relationship type, and the embedded representation of the binary relationship is treated as an input vector. Considering drawbacks of Path RNN such as the disability in downstream tasks, Single Model [80] is then put forward, which shares the relation type representation and the composition matrices of RNN across all target relations, enabling the same training data to be represented by a reduced number of parameters. Single Model significantly increases the accuracy and practicality of RNN based reasoning on Horn clause chains in large scale knowledge bases.

Models based on RL can work well because the process of inferring unknown answers over incomplete knowledge bases can be modeled as a serialized decision problem. For example, with a related framework called DeepPath [81], it is the first time for RL methods to be applied to multi-hop reasoning problems. Afterward, a neural RL method called MINERVA [82] is proposed, which learns how to find the predictive path according to the input query and avoids the requirement of modeling. In MINERVA, the large-action-space shortage of Deep Path has been well solved.

3.2.3. Challenges of Reasoning

Reasoning methods based on description logic and rules can capture semantic information hidden in a knowledge graph, significantly improving the accuracy of knowledge reasoning. Moreover, it can imitate human reasoning capabilities, making it possible to use prior knowledge to assist reasoning. However, the manual-defined logic rules have the limitations of high cost and low coverage, and the inference rules automatically mined by algorithms are likely to contain noise. In addition, the currently gained rules are relatively simple and have poor generalization ability.

The method of reasoning based on distributed representation is to map entities and relationships to low-dimensional space vectors, and to use semantic expressions for reasoning. The advantage is that it fully utilizes the structural information in the knowledge graph, and the method is convenient to extend for large-scale knowledge graphs. The disadvantage is that this kind of approach does not consider prior knowledge when establishing the reasoning model. It only considers satisfying the constraint conditions of fact triples in the knowledge graph, leading to the lack of deeper component information, which limits the reasoning ability.

The reasoning method based on the neural network directly models the fact triples, which has strong learning, reasoning and generalization abilities. However, the model is complicated and poor at explainability. Moreover, it is deficient in acquiring high-quality data to build a large-scale knowledge graph. Although on the internet, publicly available data is large in scale, it has the characteristics of fragmentation, diversity, irregularity, and noise. Besides the imperfect knowledge extraction technology, it is difficult to guarantee the quality of knowledge in the knowledge graph. Reasoning technologies based on rules, distributed representations, and neural networks are more sensitive to noisy data. Ambiguous data will lead to learning deviations and reduce the accuracy of inference results.

3.2.4. Prospect of Reasoning

Following with the fundamental challenges in knowledge graph reasoning, we further discuss and explore a series of possible future research directions as follows:

Multi-source knowledge reasoning

Knowledge reasoning based on multi-source information may reduce the connectivity and sparsity of knowledge graph by combining text corpus or other knowledge graphs so as to carry out effective reasoning. Knowledge reasoning based on multi-source information can improve reasoning performance by mixing different methods at a deeper level. In addition, in view of the outstanding performance of neural networks in various fields, including knowledge graph, the fusion of neural networks and other complementary methods will become the main research focus in the future, combining the vital learning and generalization ability of neural network with the high accuracy and explainability of rule method, excellent reasoning results are obtained. For the entity prediction task, the proposed method can be used to predict entities, the proportion of effective entities ranked in the top 10 reached 99.8% and 91.6%, respectively. How to fuse multi-source information and multiple methods to further improve reasoning performance will also become a major research direction in the future. Among them, the fusion mode, that is, how to fuse, is a major difficulty.

Intuitive reasoning

Intuitive reasoning [83] is the ability of human beings to quickly judge the answer to a problem based on their own perception without step-by-step analysis and without being constrained by certain fixed logical rules. Intuitive reasoning can quickly find an effective initial iteration position, which will also determine whether the final iteration result is globally optimal. In addition, traditional machine algorithm will fall into local minimum value when solving complex problems [84], while intuitive reasoning inspired by human brain can make decisions through strategy optimization, action feedback, reward and punishment mechanism, so as to avoid such problems and improve the generalization ability of cognitive intelligence. A famous and successful leading card of machine intuitive reasoning is AlphaGo [85]. With the ongoing optimization of AlphaGo, AlphaZero [86] and Muzero [87] were born.

AlphaZero merged the value network and the strategy network into a network on the basis of AlphaGo, which could randomly select an initial state under the condition of only given the game rules. Muzero can surpass the previous performance through the continuous reinforcement learning of self-playing games. Instead of learning the rules of relevant games, AlphaZero uses predictive strategies, value functions, and immediate rewards to calculate and plan future behavior, and the intellectual body can create rules to achieve the most accurate planning.

A series of achievements such as AlphaGo, AlphaZero, and AlphaZero show the importance of intuitive reasoning for solving complex problems. Although the realization of machine intuitive reasoning can endow the machine with the ability to predict, judge and make decisions creatively and quickly, and adapt to the complex problems in the real environment, there are still many problems, such as how to combine the intuitive sense of with the explicit knowledge of game rules, so that the machine has the ability of associative memory. Secondly, it is difficult for the machine to obtain the appropriate return function under the current environment when the system applies reinforcement learning to realize intuitive reasoning. In addition, the problems of gradient fluctuation, convergence speed and exploration efficiency also need to be solved.

Causal reasoning

Causal reasoning is crucial to survival for human beings because learning about causes contributes to imposing order and then making sense of the world. The winner of the Turing Award Pearl [88] divides the causality into three levels, namely association, intervention, and counterfactual reasoning. The current AI has merely reached the first level, while humans can reach the counterfactual level through imagination. Pearl believes that the ultimate fatal defects is that the existing AI models are mainly driven by associations, so it is difficult for machines to distinguish between causal association reasoning and false associations in data. The key to solving the problem is to apply 'causal reasoning' instead of 'reasoning for associations'.

Causal reasoning can be combined with natural language processing to extract the causal relationship between terms or phrases from a large corpus of texts for capturing and understanding the causal relationship between events and actions. Luo et al. [89] proposed a data-driven method to solve the common sense causal reasoning problem among short texts, and came up with a framework which can automatically collect causal relationships from large-scale network corpus and can correctly model the strength of causal relationships among various items.

Research on the combination of computer vision and causal reasoning has increased and has led to discovering causal signals in images [90] and generating unbiased scene graph from biased training [91]. The problem of visual reasoning has been extensively studied. Although these studies cover the complexity and diversity of vision, the basic logic, temporal and causal structures behind reasoning have rarely been explored. As moving forward, causal reasoning need further exploration to discover other potential causalities hidden in conversational visual dialog tasks and more extensive areas.

3.3. Natural Language Generating (NLG)

To make a machine have cognitive intelligence, it not only requires the machine to understand the data, process the data, and make decisions through cognitive reasoning, but more importantly, the machine is supposed to have the ability to express the reasoning results in a way that humans can understand. Therefore, how to make the machine generate natural language in line with human understanding is a crucial aspect of cognitive intelligence.

Along the way to cognitive intelligence, NLG plays a crucial role: it is responsible for converts a cognitive system action into a human-understandable response. Therefore, the response is supposed to be fluent, adequate. NLG has significant influence on users' experience.

In this section, we comprehensively review the concept, key technologies, problems and challenges and future research directions of NLG.

3.3.1. Basic Concept of NLG

As a crucial component in cognitive graphs, Natural Language Generation (NLG) technology is a hot topic which aims make computers have human-like expression, that is, automatically generate a high-quality natural language based on some essential information.

In the most widely-cited surveys of NLG methods [92,93], NLG is characterized as the subfield of AI and computational linguistics that is concerned with the construction of computer systems that can produce understandable texts or other human languages from some underlying non-linguistic representation of information.

It has been pointed out that precisely defining NLG is rather difficult [94]: everybody seems to agree on what the output of an NLG system should be text, but what the exact input is vary substantially [95]. Examples include flat semantic representations, numerical data and structured knowledge bases. More recently, generation from visual input such as image or video has become an important direction [96–98], which extend the concept of NLG to non-linguistic data.

3.3.2. Key Technologies of NLG

In this section, we systematic review the key technologies of NLG, including Markov Chains [99], RNN [100], LSTM [101], Transformer [102], ELMo [103], BERT [31].

Markov chains

Markov chains are the earliest algorithms for language generating. It predicts the next word in a sentence from the current word. For example, the model is trained in the following two sentences: "I drink coffee in the morning" and "I eat sandwiches with tea". The probability of "coffee" after "drink" is 100%, and the probability of "eat" and "drink" after "I" is 50%. When calculating the probability of the next word, Markov chain takes into account the relationship between each word. The model was first used to provide suggestions for the next word generating of smartphone input sentences.

Recurrent Neural Network (RNN)

Inspired by the working principle of the human brain, the neural network provides a new method for computing by modelling the nonlinear relationship between input and output data, which is called neural language modelling. Recurrent neural network (RNN) [100] is a kind of neural network which can capture the sequence characteristics of input data. As is shown in Figure 11, each item in the sequence is fed into a feedforward network, and the output of the model is taken as the next item of the sequence. This process can help to store the information of each previous step. Such "memory" makes RNN perform well in language generating because remembering past information can help better predict the future. Significantly different from the Markov chain, RNN focuses not only on the current word, but also on the processed word. But there is a big problem with RNN: the catastrophic forgetting [104]. With the increase of sequence length, RNNs can not store the words encountered long ago, so they can only predict

based on the latest words. This makes RNNs unable to be used to generate coherent long sentences.

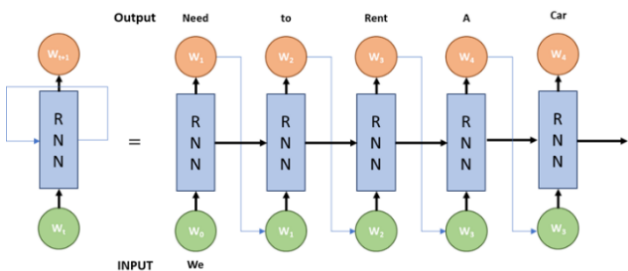


Figure 11. Overview architecture of RNN [100].

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) [101] is a variant of RNNs, which is more suitable for processing long sequences than RNNs. LSTM is widely used, and its structure is similar to that of RNNs. The difference is that RNN has only one simple layer structure, while LSTM has four layers. An LSTM consists of four parts: cell, input gate, output gate and forgetting gate. For example, when input is “I am from Spain. I am fluent in ____”. In order to correctly predict the next word “Spanish”, LSTM will pay more attention to “span” in the previous sentence and use cell to memorize it. As the sequence is processed, the cell stores the acquired information, which is used to predict the next word. When a period is encountered, the forgetting gate will realize that the context of the sentence has changed and ignored the state information stored in the current cell. In other words, the function of the forgetting gate is to make the recurrent neural network “forget” the information not used before. LSTM and its variants can solve the problem of gradient disappearance and generate coherent sentences. However, LSTM also has its limitations: high computational requirements and difficult to train.

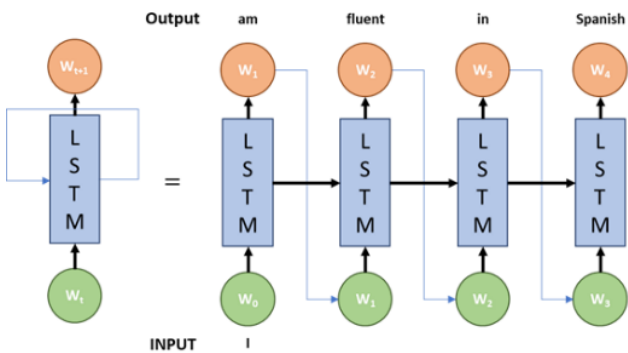


Figure 12. Overview architecture of LSTM [101].

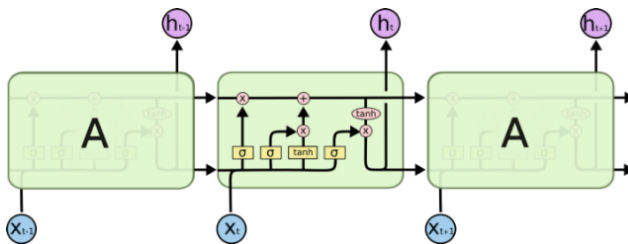


Figure 13. The repeated module in LSTM containing four layers of interactive neural network layer [101].

Transformer

Transformer [102] was first proposed by the Google team in the paper "Attention Is All You Need" in 2017, which involves a new method called self attention mechanism. Transformer is widely used to solve NLP problems, such as language modeling, machine translation, and text generating. The Transformer model consists of a group of encoders and a group of decoders. The former is responsible for processing any length of the input, and the latter is responsible for outputting the generated sentences.

The encoder processes the input sentence and generates a representation for it. The decoder uses representation to generate sentences for output. The initial representation or embedding of each word is represented by a hollow circle. Afterward, the transformer model uses SAM to obtain the relationship between all other words and generate a new representation of each word, such as a single circle in Figure 14. Repeat this step for each word to generate new representations continuously. Similarly, the decoder generates words from left to right.

Different from LSTM, Transformer needs fewer steps. The application of SAM can directly capture the relationship between all words in a sentence without considering the positions of them.

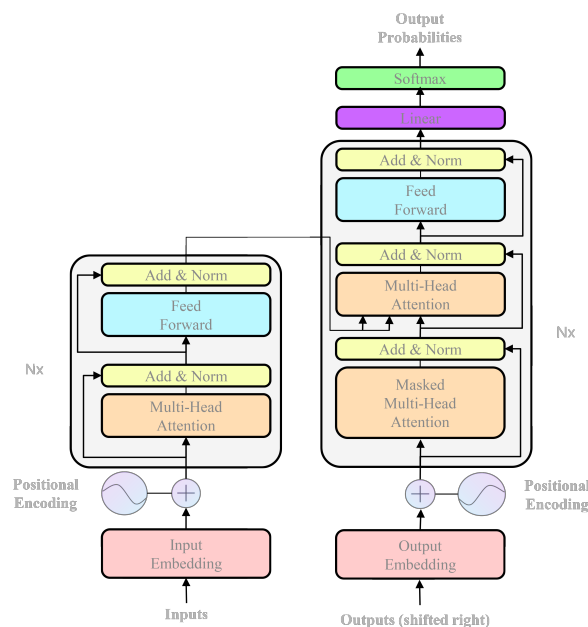


Figure 14. Overview architecture of Transformer [102].

Transformer model architectures have garnered immense interest lately due to their effectiveness. In the field of natural language generating, for example, Transformers have become an indispensable staple in the modern deep learning stack. Recently, a dizzying number of "X-former" models have been proposed - Reformer [105], Linformer [106], Performer [107], Longformer [108], to name a few - which improve upon the original Transformer architecture, many of which make improvements around computational and memory efficiency.

In 2018, OpenAI proposed the GPT model [109], a language model based Transformer, which can be migrated to a variety of NLP tasks. The basic idea is to apply the pre-trained language model without changing the model structure as much as possible. It can capture a more extended range of information compared with RNN, and is faster than RNN in computing speed. However, it is not scalable, and the structure of input data needs to be adjusted for some types of tasks.

Embeddings from Language Models(ELMo)

ELMo is a new type of method deeply contextualized word representation [103], which can not only model the complex syntax and semantics features but also change

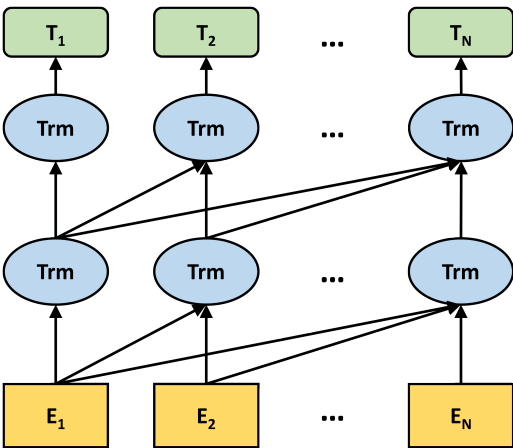


Figure 15. Overview architecture of GPT [109].

with linguistic context. ELMo adopts a two-layer bi-directional LSTM language model to understand the meaning in context. The structure of the model is shown in Figure 16.

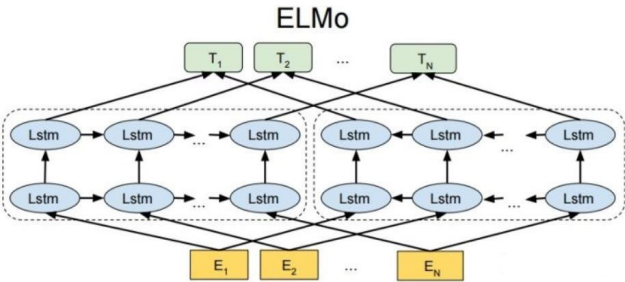


Figure 16. Overview architecture of ELMo [103].

The ELMo model is first trained on a large predictive library, and the vector of each word is represented by the function of the internal state of BiLSTM, thus, the internal state is mixed with all the semantics. The text of the downstream task is trained in a model where each word has a specific context, and each word is represented by a linear combination of internal states. The BiLSTM layer in the ELMo model can effectively encode different types of syntax and semantics in the context. A lot of studies have proved that Transformer is much better at extracting features than LSTM, so the ELMo would have a more dramatic impact if it adapted the Transformer as a feature extractor. In addition, the ability to adopt the fusion feature of bidirectional splicing may be weaker than that of Bert integration.

The new ELMo faced some knotty problems to adopt the fusion feature of bidirectional splicing, so new methods are on the horizon.

Bidirectional Encoder Representations from Transformers(BERT)

In 2018, Google introduced a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. As shown in Figure 17, BERT is designed to pre-train deep bidirectional representations from unlabeled texts by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

The BERT model has three innovations: bidirectional transformer, masked language model, sentence-level relation.

Bidirectional transformer: When understanding a language, pre-order words can be helpful in understanding post-order words, and in the same way, subsequent words

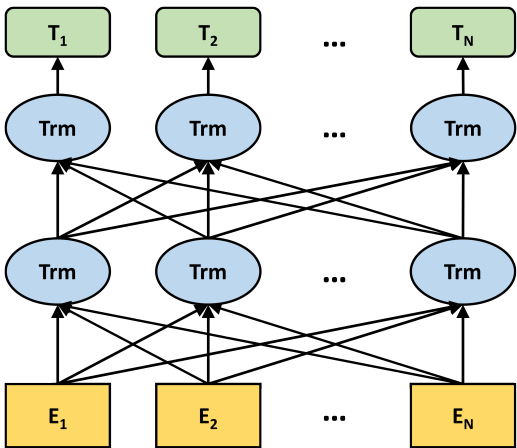


Figure 17. Overview architecture of BERT [31].

can be helpful in understanding forward words. Early language models could be trained from left to right or right-to-left, but the two could not be conducted at the same time.

Masked language model: Humans understand language with contexts in mind. BERT cleverly utilized the idea of filling in the blanks, put forward the masked language model to achieve a two-way transformer.

Sentence-level relation: Learn the relationship between sentences by predicting whether sentence B is the next sentence of sentence A.

The BERT model extends the pre-training supervised fine-tuning model to a deeper two-way structure, where long-distance semantic information can be obtained, context semantics can be left and right, and can be performed in parallel. The disadvantages of the BERT model mainly come from the masked language model: the "Mask" tag does not appear in the actual prediction, too much use of the "Mask" during training affects model performance, and only 15% of each batch is predicted, so the BERT converges more slowly than the left-to-right model.

Table 2: Key technologies of NLG.

Technology	Reference	Mechanism	Advantage	Disadvantage
Markov Chains	[99]	Probability calculation of the next word	Next-word prediction in a sentence	Only focusing on the current word
RNN	[100]	Neural Network	Consideration of both the current word and the processed word	Catastrophic forgetting
LSTM	[101]	A four-layer structure	Long sequence suitability	High computational requirements and training difficulties
ELMo	[103]	Two-layer bi-directional LSTM	Changeability with linguistic context	Weaker feature extractor
BERT	[31]	Bidirectional transformer, masked language model, sentence-level relation	Easier to create state-of-the-art models for a wide range of tasks	Converging more slowly

3.3.3. Challenges of NLG

Despite the recent success of pre-trained language models such as GPT-3 [110] on various language generation tasks, these models are still struggling on generation tasks that require reasoning over commonsense knowledge which is not explicitly stated in the context. A significant bottleneck of NLG is how to represent the semantics of natural language accurately. For example, in the process of human-computer interaction, the users' intention must firstly be understood. At present, there are two methods commonly used in the industry: semantic analysis methods based on knowledge or semantic rules and semantic analysis methods based on statistics. Although both types of methods can derive natural language semantics to a certain extent and determine the association between information, the method based on knowledge and semantics rules cannot cover all language phenomena, the reasoning process is complex, and it cannot handle uncertain events. There are defects and limitations in the compatibility of rules and application levels. The establishment of knowledge and semantic rules is the bottleneck problem. The statistics-based methods rely too much on the support of large-scale corpora, and they are susceptible to data sparseness and data noise.

With the increase of user needs, traditional template generation methods based on syntactic-semantic rules cannot meet the requirements of social development. With the continuous development and maturity of deep learning technology, deep learning research for natural language generation has achieved certain results, but there is no breakthrough. In the current existing deep learning models, the difficulty lies in the optimization and adjustment of parameters in the model construction process, mainly including the number of deep network layers, regularization problems and network learning rate. Possible solutions include the use of multi-core machines to improve the network training speed, select suitable optimization algorithms for different applications. Combining the two methods and learning from each other is the future research direction in NLG.

3.3.4. Prospect of NLG

For human-like cognitive interactive devices, a system should take various kinds of knowledge into consideration, decide what knowledge will be utilized in specific reasoning contexts, and make the final decision by taking all valuable knowledge into consideration.

Various language generation models have achieved remarkably effective results. These models remain stuck in generation tasks that need to reason over commonsense knowledge that is not clearly stated in the context. Although some models have been proposed to enhance commonsense reasoning ability by elaboratively learning some relational patterns from large-scale corpora, they do not fully utilize the commonsense knowledge which can provide more explicit knowledge grounding [111]. Several attempts have been widely explored to deal with this defect [112–114]. By utilizing triple information in commonsense knowledge bases such as ConceptNet [115] and ATOMIC [116], transferring commonsense knowledge into pre-trained language models is a promising direction.

However, these methods have several potential drawbacks. Firstly, they ignore the rich structural relevance of the concepts in the knowledge graphs. Secondly, recovering knowledge triples at the post-training stage is hardly conducive to exploit the encoded knowledge in generation tasks. Therefore, a straightforward way of making use of external commonsense knowledge is to take advantage of both structural and semantic information of the knowledge graph and reason over multi-hop relational paths, so multiple connected triples can provide chains of evidence for grounded text generation.

To facilitate explicit commonsense reasoning in text generation, Haozhe Ji [117] proposed Generation with Multi-hop Reasoning Flow, a novel generation model that conduct dynamic multi-hop reasoning for knowledge-enriched language generation. This model leverages the structural and semantic information of the external knowledge

and aggregates evidence along relational paths which is beneficial for the generation of some critical concepts.

In recent years, with the development of deep learning technology, the accumulation of large-scale data corpus and the actual needs of the industry, the related research and engineering implementation of natural language generation have received a lot of attention.

The existing neural network-based natural language generation models are far from enough for the application of automatic NLG. It is clear that a full end-to-end goal-driven dialogue system should not only output a final sentence to respond to an input sentence, but also keep and update fruitful internal representations or memories for dialogues. At the same time, because of various factors such as fuzzy language boundaries, diverse ambiguities, irregular expressions, limited learning corpus scale, and complex application scenarios, NLG is not mature in practical applications, and there are still many problems and challenges in related research. Natural language generation technology is currently mainly based on rule-based template generation technology in practical applications. Although it can realize the automatic text generation function with a less complex text structure, the result is rigid, and the later expansion depends heavily on the initial planning. The specific manifestations are: (1) The format of the automatically generated text is fixed, the structure is not flexible, and it is difficult to modify. (2) The expression of generated sentences is not flexible and does not meet the characteristics of the diversity of language expression.

The generation technology based on deep learning does not require a manual design of rule templates and grammar planning process, can automatically learn grammar and semantic rules, and can directly extract features from the text. In the automatically generated text, the expression of sentences is more flexible, and it is more in line with the diversity of language expression.

4. Challenges of Cognitive Graphs

Current advancements in AI have yielded tremendous improvements across academic and industry research communities. AI researchers still face considerable challenges in developing truly intelligent systems. Concerns about poor robustness and lack of explainability are raised by some influential thinkers. The cognitive graph which is based on neural-symbolic integration offers an opportunity to treat with these challenges. However, previous researches have revealed that the realization of a symbolical understanding neural system is hard, requiring a great deal of work related to refined theories and engineering. Furthermore, at present it is quite unclear how symbolic processing emerges from neural activities of complex neural network. In contrast, human brains are impressive examples of the admirable neural system that have effectively descended upon humans. At the same time, human brains are able to deal with symbolic tasks successfully. Therefore, it is necessary to take inspiration from neural science as a breakthrough for cognitive intelligence.

4.1. Challenges towards Neural-Symbolic based Cognitive Graphs

Recently, there has been a clear tendency towards integrating symbolic reasoning and deep learning. Neural AI is continuous, distributed and good at dealing with large-scale perceptual data. Usually, neural AI is implemented efficiently using propagation of activation and units of tensor processing. Symbolic AI is generally localized and discrete, which has the ability of sophisticated reasoning, including epistemic, analogic and temporal reasoning. Recent researches have revealed that non-classical logics provide an applicable language for describing neural networks [118,119]. In [120], Marcus made his point clear by stating the necessity of hybrid systems, and attempted to identify what makes a system hybrid. At this point, all attempts to create such a bridge between System 1 and System 2 are beneficial and should be commended given our lack of understanding of how human brains work.

The challenges for neural-symbolic integration emerge from the goal of robust learning, expressive reasoning and effective integration.

Turing award winner Leslie Valiant pointed out that the key challenge is to build a rich semantics and robust representation language for intelligent cognitive behaviours [121]. Avila Garcez [122] identified representation as the bottleneck and argued for the importance of focusing on representation since representation precedes learning as well as reasoning. At the same time, the difficulty of neural networks in explainability and robustness also leads to the need for a bridge between distributed and localized representations for reasoning. For several years neural-symbolic computing has been seeking to establish such a bridge [123–125], leading to the emergence of several methods. According to Kautz's taxonomy [126], there are roughly six different types. TYPE 1 is standard deep learning with symbolic input and output. TYPE 2 is a kind of hybrid system where the core neural network is loosely-coupled with a symbolic problem solver such as the Monte Carlo tree search. One typical example of TYPE 2 is DeepMind's AlphaGo. TYPE 3 is a hybrid system where a neural network focusing on one task (e.g. object segmentation) interacts via its input and output with a symbolic system, subsequently followed by a commentary task (e.g. query answering). Recent efforts in this direction include Neuro-symbolic Concept Learner [127] and DeepProbLog [128]. TYPE 4 refers to a learning architecture that extrapolates to harder symbolic reasoning problems [129,130]. Moreover, TYPE 4 includes tightly-coupled but localized neural-symbolic systems, where the key point is to build a one-to-one correspondence between neurons and elements of logical formulas [119]. TYPE 5 mainly refers to those tightly-coupled but distributed neural-symbolic systems in which a logic symbolic rule is mapped into the form of an embedding. Classical examples include Tensor Product Representations [131]. Finally, a TYPE 6 system is supposed to have the ability of true symbolic reasoning inside a neural engine by using an attention schema to achieve combinatorial reasoning. Recent researches include [132–134], and further researches on TYPE 6 systems come closer to the essence of neural-symbolic computing.

In summary, by paying attention to the developments of neural-symbolic integration, we are getting closer to the ultimate goal of promoting the faster development of cognitive science with expressive reasoning and robust learning capabilities.

To develop neural network models with a symbolic interpretation, the key is to learn representations neurally and make them available for use symbolically. An adequate language for describing knowledge encoded in neural networks is an important ingredient. As a part of the interplay between learning and reasoning, constraint satisfaction is therefore another ingredient. In addition to the above ingredients, there are a few tough challenges for neuro-symbolic AI. So far, there has been no proper way to achieve the combination of language and structure. Therefore, the neural-symbolic cognitive graph has a long way to go inevitably.

4.2. Challenges towards Robust Cognitive Graph

In a certain sense, considering the current technological development level, a hopefully achievable robust cognitive graph is not necessarily a superhuman, but at least can be counted on in a reliable way, to apply what it knows to a wide range of problems.

Ernie Davis and Gary Marcus [120] put forward a view that we have no hope of achieving robust cognitive graphs without first developing systems with deep understanding, which equips with the ability to identify subtle patterns in complex data sets and address questions like a journalist. Although GPT-3 [110] can produce stories and generate compelling examples, it often falls apart under close inspection and fails to grasp details [135]. The reality is that GPT-3 only has restricted representations and it fails to reflect a deep understanding.

Achieving robustness is not just about developing the right cognitive prerequisites. Actually, it is about developing the right cognitive models. DeepMd's Atari game system, DQN, for example, almost lacks cognitive models entirely. Fortunately, in

the reinforcement learning, there is some sort of internal model that is self-generated, pointing to the internal system that to some extent correlate with cognitive states, where a classical example is MuZero [136]. Model-free reinforcement learning often works remarkably well. However, this does not mean the model-free reinforcement learning is a good general solution to cognitive intelligence, because they lack rich cognitive models of the environments, where they operate and they require considerable retraining.

4.3. Challenges towards Explainable Cognitive Graphs

Explainability is the ability to provide explanations in understandable terms to a human and provide human-understandable justifications for their actions. The importance of explainability in systems that should be highly-reliable, ethical or legal requirements has already been emphasized by many papers [137,138]. Explainability is crucial for some systems that are required to be highly reliable such as the medical system and the financial system, because an error may cause catastrophic results. Explainability can make potential failures easier to detect, avoiding severe consequences. Moreover, it can help engineers to find out the root cause and provide a fix accordingly. Knowledge extraction is an effective ingredient towards explainable AI systems. The main barrier is the efficient extraction of correct, compact and complete knowledge. In the case of local explanations, a knowledge-base is certainly more explainable than a neural network because it offers a trace showing how an outcome was obtained.

Many aspects of the explainable cognitive graphs are being investigated at present. Some pendent research lines include: Is an explanation intended set to improve system performances and reduce bias fairness, or just to understand the decision process? Is an explanation required because one does not trust the system and would like to create a different outcome, or just want to doubt the normative system? Early efforts on these questions mainly focused on fidelity: a measure of the accuracy of the extracted knowledge in relation to the neural network. Without high fidelity results, an apparently good explanation is likely not to be an expected explanation of the underlying system. D'Avila Garcez [139] proposed a way of measuring fidelity with local methods, which should be adopted and promoted vigorously. Since soundness is normally interconnected with exponential complexity, a measure of fidelity may be available in practice and knowledge extraction should have communications between users and cognitive systems. Communications with the system mean being capable of asking questions (querying the system) and checking one's understanding (obtaining a rationale for the result). The user can either agree or disagree with the outcome, then provide useful feedback or direct instruction to change the outcomes.

The existing AI-based decision support systems deal with large scale data in a timely pattern. Thus, with a so-called human-in-the-loop approach where experts or end-users may be accountable to the decisions, domain experts or end-users may soon feel less capable of over-riding recommendations which are based on too much more data than they could handle with. The current reality is that in order to function well with Big Data, the current cognitive system must execute a form of triage of the data to be presented to the expert. Without the abilities of system communication and knowledge extraction, the decision maker will be out of control. Lastly, the simple extraction of rules is insufficient. We need to extract confidence values to rank extracted rules. In this way, a system will know when it does not know. The adversarial approaches and knowledge extraction for robustness have made a contribution here. In a nutshell, for many reasons discussed above, a measurable form of knowledge extraction is a fundamental part towards explainable cognitive graphs.

5. Prospect of Cognitive Graphs

In this second-to-last section of our survey, we give an overview of what are considered the most potentially theoretical challenges on the way towards robust and

explainable AI currently. In this section, the future directions and practical questions are discussed.

It has been argued eloquently that to build a semantical, explainable and ultimately trustworthy AI system, one needs to pay attention to a lot of aspects, such as integrated neural-symbolic approaches, extraction of symbolic knowledge from trained neural networks, integrated neural-symbolic approaches to logical reasoning, cognitive and biologically-inspired neural-symbolic agents, and applications in robotics, fraud prevention, semantic web, software engineering, fault diagnosis, verification and validation, bioinformatics.

New developments should investigate the previously unconsidered properties of symbolic and neural formalisms, which might shed light on offering explanations for the performances and the empirical applicability.

In essence, the crucial innovation of cognitive graphs is to reduce the information loss during the construction of the graphs, transfer the pressure of information processing to retrieval and natural language understanding algorithms, and retain the graph structures for explainable relational reasoning.

In the future, it is necessary to focus on how to capture structural information and learn rule knowledge at the same time, so as to improve the performance of cognitive graph reasoning. In the big data era, large-scale, diverse forms, scattered distribution, dynamic changes and low-quality data features bring new challenges to AI technologies. It is necessary not only to learn the distribution representation of data from the perspective of perception but also to interpret the semantics of data from the perspective of cognition. The research and development of cognitive graphs that integrate core technologies such as common sense knowledge graphs, cognitive reasoning and logical expression will become the key to the breakthrough of the next generation of AI technologies. Given the fast pace at which developments occur both in industry and academia, we feel it is helpful to point to potential future directions.

5.1. Common Sense Cognitive Graphs

Common sense knowledge has long been a key component in human-like reasoning as a fundamental part of AI. The significance of having common sense knowledge will emerge when it comes to action, interpretation and decision-making. If you do not know what happens when a bottle breaks or what can cause a fire, it is difficult to make reasoning about what will happen surrounding you. Obviously, you cannot reliably make plans.

As mentioned previously, deep learning is essentially based on a "big data for small tasks" paradigm, which has a demand for massive amounts of data in a single narrow task. Yixin Zhu [140] proposed "small data for big tasks" paradigm which is capable of solving a wide range of tasks with a few data. The new paradigm consider functionality, physics, intent, causality and utility (FPICU) as core ingredients of cognitive AI. Fig 18 is an example of in-depth understanding through cognitive reasoning and joint parsing. Reasoning with FPICU relies on the combination of top-down reasoning based on abstract knowledge and bottom-up reasoning based on visual patterns. This means the systems would continue to make reasoning from the observation of visible, pixel-represented scene aspects, as they can do today, and make reasoning with human-like common senses based on FPICU understanding.

These processes can feed on each other, boosting the performance of the overall system. The breakthroughs in human vision fields demonstrate the promising potential power of the FPICU. Commonsense understanding is pivotal in the development of cognitive intelligence. More efforts should be made to realize the acquisition and understanding of common senses.

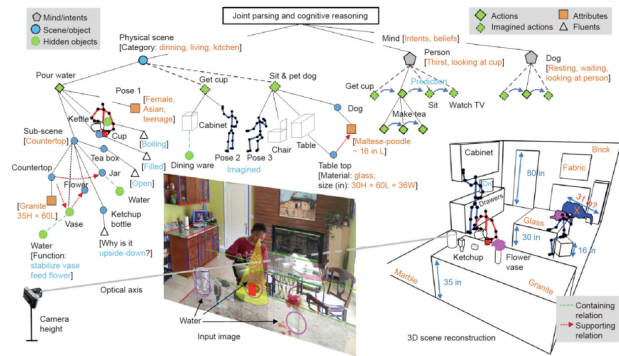


Figure 18. An example of in-depth understanding through cognitive reasoning and joint parsing. Reasoning with FPICU depends on the integration of bottom-up reasoning based on visual patterns and top-down reasoning based on abstract knowledge [140].

5.2. Event Logic Cognitive Graphs

The existing cognitive graphs are still generally based on "concept and their relations between concepts", which failed to mine "event evolutionary logic". In practical applications, the evolutionary rules and patterns between events are valuable common sense knowledge. It is of great significance to explore this kind of knowledge in order to understand the laws of human behaviors and social developments.

The concept of Event Logic was proposed by X Ding et al. [141]. To reveal evolutionary patterns and development patterns of real world events, J P Wang et al. [142] proposed a directed cyclic graph called Event Logic Graph (ELG). Essentially, ELG is an event evolutionary logic knowledge base, where nodes are events, and edges stand for the causal, sequential, conditional or hypernym-hyponym relations between events. The edges of ELG represent abundant logical rules, including concatenation relationships, causality relationships and conditional relationships. The concepts, key techniques and applications of ELG is shown in Figure19.

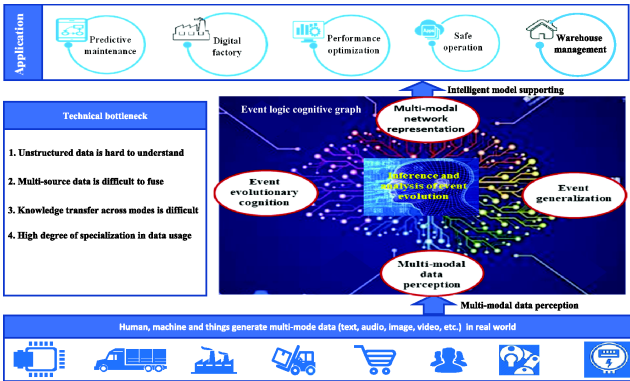


Figure 19. The event cognitive vault concepts, key techniques and applications [142].

Large-scale ELGs have tremendous potentials and can be applied to plenty of downstream tasks, including event prediction, commonsense reasoning, question answering, dialogue generation, consumption intention mining and so on.

As a new way of knowledge organization, representation and management, the ELG is an important breakthrough of cognitive intelligence. Future research should focus on organizing the universal event evolutionary principles and patterns into a knowledge graph based on the extracted temporal and causal event pairs.

5.3. Cognitive Robots

The emergence of industrial robots and emotional escort robots demonstrates that cognitive AI is a tendency for AI to move from narrow weak intelligence to general

strong intelligence. The human-robot interaction aims to build robotic systems that can collaborate with humans. To support social interaction, robots need the ability to generate and understand a variety of verbal expression and nonverbal behaviors of humans. There are a series of technical challenges, including understanding context, expressing tasks and domain knowledge and optimizing generated content.

Recent researches show that cognitive robots require a significant cognitive ability for deep thinking and reasoning based on the development of cognitive systems and brain-inspired systems. The emergence of language is considered as a fruitful accomplishment in multi-agent decentralized collaborations. Being able to communicate and collaborate with other agents is a crucial component of cognitive AI. In classic AI, a multi-agent communication strategy is modeled using a pre-defined rule-based system (e.g., adaptive learning of communication strategies in MAS [143–145]). To scale up from rule-based systems, decentralized partially observable Markov decision processes were devised to model multi agent interactions, with communications considered as a special type of actions [146,147]. With the success of RL in single agent games [148], generalizing Q-learning [149] and actor-critic-based methods [143,150] from single-agent systems to MAS have been booming topics in recent years.

It is conceivable that in the future, combined with psychological considerations on embodied interactions and low-level representations of the agent's sensing, cognitive modelling is created as output of the ground layer of the envisioned architecture. This output then is fed into a second layer which performs an extended form of anchoring, not only grounding symbols referring to perceived physical objects, but also dynamically adapting and repairing acquired mappings between environment and internal representation.

6. Conclusion

AI has achieved remarkable results with the support of computing power, big data and deep learning. To develop a more explicable and robust AI, we need a system that can routinely acquire, represent and manipulate common senses and abstract knowledge, subsequently applying that knowledge in the service of reasoning over complex problems.

The advanced development stage of AI is supposed to be a hybrid, knowledge-driven reasoning-based approach that could provide the substrate for the more explicable and robust AI. The cognitive graph, which is based on the dual-channel theory and the neuro-symbolic architecture, meets the requirement for critical cognitive prerequisites. Taken together, progress towards these prerequisites could provide a substrate for richer, more intelligent systems.

In this paper, we highlight the key ideas and principles of the cognitive graph and illustrate the main methodological approaches for the integration of effective neural learning with symbolic-based, knowledge representation and reasoning methods.

In summary, by paying attention to the developments of cognitive graph, we are getting closer to the true artificial intelligence, or at least promoting the faster development of cognitive sciences and AI.

Acknowledgment

References

1. Ji, Q. Analysis on E-commerce Course Reform under the Background of Artificial Intelligence. *Journal of Physics: Conference Series* **2020**, 1533, 032088. doi:10.1088/1742-6596/1533/3/032088.
2. Turk, M.; Pentland, A. Eigenfaces for Recognition. *Journal of cognitive neuroscience* **1991**, 3, 71–86. doi:10.1162/jocn.1991.3.1.71.
3. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2009**, 31, 210–227. doi:10.1109/TPAMI.2008.79.

4. Winograd, T. Understanding natural language. *Cognitive Psychology* **1972**, *3*, 1 – 191. doi: [https://doi.org/10.1016/0010-0285\(72\)90002-3](https://doi.org/10.1016/0010-0285(72)90002-3).
5. Vinyals, O.; Le, Q. A Neural Conversational Model. *ICML Deep Learning Workshop, 2015* **2015**.
6. McAulay, R.; Quatieri, T. Speech analysis/Synthesis based on a sinusoidal representation. *Acoustics, Speech and Signal Processing, IEEE Transactions on* **1986**, *ASSP-34*, 744 – 754. doi: 10.1109/TASSP.1986.1164910.
7. Moulines, E.; Charpentier, F. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication* **1990**, *9*, 453 – 467. Neurospeech '89, doi:[https://doi.org/10.1016/0167-6393\(90\)90021-Z](https://doi.org/10.1016/0167-6393(90)90021-Z).
8. Rabiner, L. A Tutorial on Hidden Markov Models and Selected Applications. *Proceedings of The IEEE - PIEEE* **1989**, *77*.
9. Shannon, R.V.; Zeng, F.G. Speech recognition with primarily temporal cues. *Science* **1995**, *270*, 303.
10. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556* **2014**.
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
12. Wu, Y.; Liu, B.; Ma, X. Building an "Artificial Intelligence + Education" Ecosystem. *Journal of Distance Education (in Chinese)*. **2017**, *35*, 27–39.
13. Wienrich, C.; Latoschik, M.E. eXtended Artificial Intelligence: New Prospects of Human-AI Interaction Research **2021**.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026–1034. doi:10.1109/ICCV.2015.123.
15. Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; Chen, J.; Chen, J.; Chen, Z.; Chrzanowski, M.; Coates, A.; Diamos, G.; Ding, K.; Du, N.; Elsen, E.; Zhu, Z. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *ArXiv* **2016**, *abs/1512.02595*.
16. Tramer, F.; Behrmann, J.; Carlini, N.; Papernot, N.; Jacobsen, J.H. Fundamental Tradeoffs between Invariance and Sensitivity to Adversarial Perturbations. *ArXiv* **2020**, *abs/2002.04599*.
17. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. *CoRR* **2014**, *abs/1312.6199*.
18. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Laskov, P.; Giacinto, G.; Roli, F. Evasion Attacks against Machine Learning at Test Time. *ArXiv* **2013**, *abs/1708.06131*, 387–402. doi: 10.1007/978-3-642-40994-3_25.
19. Tramèr, F.; Kurakin, A.; Papernot, N.; Boneh, D.; McDaniel, P. Ensemble Adversarial Training: Attacks and Defenses. *ArXiv* **2018**, *abs/1705.07204*.
20. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal Adversarial Perturbations. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) **2017**, pp. 86–94.
21. Wu, D.; Wang, Y.; Xia, S.T.; Bailey, J.; Ma, X. Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets. *ArXiv* **2020**, *abs/2002.05990*.
22. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; Bizer, C. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* **2015**, *6*, 167–195. doi:10.3233/SW-140134.
23. Vrandečić, D.; Krötzsch, M. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. doi:10.1145/2629489.
24. Gini, G. Industry-scale knowledge graphs: lessons and challenges. *Computing reviews* **2020**, *61*, 186–186.
25. Bollacker, K.; Tufts, P.; Pierce, T.; Cook, R. A platform for scalable, collaborative, structured information integration. In *IIWeb*, 2007.
26. Hoffart, J.; Suchanek, F.M.; Berberich, K.; Lewis-Kelham, E.; de Melo, G.; Weikum, G. YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. Proceedings of the 20th International Conference Companion on World Wide Web; Association for Computing Machinery: New York, NY, USA, 2011; WWW '11, p. 229–232. doi:10.1145/1963192.1963296.

27. Henson, C.; Schmid, S.; Tran, A.T.; Karatzoglou, A. Using a Knowledge Graph of Scenes to Enable Search of Autonomous Driving Data. ISWC Satellites; Suárez-Figueroa, M.C.; Cheng, G.; Gentile, A.L.; Guéret, C.; Keet, C.M.; Bernstein, A., Eds. CEUR-WS.org, 2019, Vol. 2456, *CEUR Workshop Proceedings*, pp. 313–314.
28. Matuszek, C.; Witbrock, M.; Kahlert, R.C.; Cabral, J.; Schneider, D.; Shah, P.; Lenat, D. Searching for Common Sense: Populating CycTM from the Web. Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3. AAAI Press, 2005, AAAI'05, p. 1430–1435.
29. Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.; Nakashole, N.; Platanios, E.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R.; Wijaya, D.; Gupta, A.; Chen, X.; Saparov, A.; Greaves, M.; Welling, J. Never-Ending Learning. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), 2015.
30. Ding, M.; Zhou, C.; Chen, Q.; Yang, H.; Tang, J. Cognitive Graph for Multi-Hop Reading Comprehension at Scale. 2019. doi:10.18653/v1/P19-1259.
31. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding **2018**.
32. Battaglia, P.; Hamrick, J.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; Gulcehre, C.; Song, F.; Ballard, A.; Gilmer, J.; Dahl, G.; Vaswani, A.; Allen, K.; Nash, C.; Langston, V.; Pascanu, R. Relational inductive biases, deep learning, and graph networks. *ArXiv* **2018**, *abs/1806.01261*.
33. Bengio, Y. FROM SYSTEM 1 DEEP LEARNING TO SYSTEM 2 DEEP LEARNING YOSHUA BENGIO. *NeurIPS'2019 Keynote* **2019**.
34. Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. 2017, pp. 1870–1879. doi:10.18653/v1/P17-1171.
35. Feigenbaum, E.A. How the “What” Becomes the “How”. *Commun. ACM* **1996**, *39*, 97–104. doi:10.1145/229459.229471.
36. Qiao, L.; Yang, L.; Hong, D.; Yao, L.; Zhiguang, Q. Knowledge Graph Construction Techniques. *Journal of Computer Research and Development* **2016**, *53*, 582–600. doi:10.7544/jssn1000-1239.2016.20148228.
37. Huffman, S.B. Learning information extraction patterns from examples. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*; Wermter, S.; Riloff, E.; Scheler, G., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 1996; pp. 246–260.
38. Jim.; Cowie.; Wendy.; Lehnert. Information extraction. *Communications of the ACM* **1996**.
39. Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Linguisticae Investigationes* **2007**, *30*, 3–26. doi:https://doi.org/10.1075/li.30.1.03nad.
40. Ying, W.; Guangli, L.; Shengli, B.; Zhimin, Y.; Ying, W.; Guangli, L.; Shengli, B.; Zhimin, Y. Attribute Extraction, 2012.
41. Meiji, C.; Li, L.; Wang, Z.; You, M. *A Survey on Relation Extraction; Knowledge Graph and Semantic Computing*. Language, Knowledge, and Intelligence, 2017; pp. 50–58. doi: 10.1007/978-981-10-7359-5_6.
42. Smirnov, A.; Levashova, T. Knowledge fusion patterns: A survey. *Information Fusion* **2019**, *52*, 31–40. doi:https://doi.org/10.1016/j.inffus.2018.11.007.
43. Jain, A.; Pennacchiotti, M. Open Entity Extraction from Web Search Query Logs. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010); Coling 2010 Organizing Committee: Beijing, China, 2010; pp. 510–518.
44. Liu, X.; Song, Y.; Liu, S.; Wang, H. Automatic Taxonomy Construction from Keywords. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Association for Computing Machinery: New York, NY, USA, 2012; KDD '12, p. 1433–1441. doi:10.1145/2339530.2339754.
45. Whitelaw, C.; Kehlenbeck, A.; Petrovic, N.; Ungar, L. Web-scale named entity recognition. 2008, pp. 123–132. doi:10.1145/1458082.1458102.
46. Wang, C.; Danilevsky, M.; Desai, N.; Zhang, Y.; Nguyen, P.; Taula, T.; Han, J. A Phrase Mining Framework for Recursive Construction of a Topical Hierarchy; Association for Computing Machinery: New York, NY, USA, 2013; KDD '13, p. 437–445. doi:10.1145/2487575.2487631.
47. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* **2017**, *29*, 2724–2743. doi:10.1109/TKDE.2017.2754499.

48. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *Computer Science* **2013**.
49. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. 2013, pp. 2787–2795.
50. Feng, J. Knowledge Graph Embedding by Translating on Hyperplanes. 2014.
51. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. *Proceedings of AAAI* **2015**, pp. 2181–2187.
52. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. Meeting of the Association for Computational Linguistics & the International Joint Conference on Natural Language Processing, 2015.
53. Xiao, H.; Huang, M.; Zhu, X. TransG : A Generative Model for Knowledge Graph Embedding. 2016, pp. 2316–2325. doi:10.18653/v1/P16-1219.
54. Nickel, M.; Tresp, V. A Three-Way Model for Collective Learning on Multi-Relational Data.
55. Yang, B.; Yih, W.t.; He, X.; Gao, J.; Deng, I. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *Machine Learning* **2014**.
56. Nickel, M.; Rosasco, L.; Poggio, T. Holographic Embeddings of Knowledge Graphs **2015**.
57. Bordes, A.; Glorot, X.; Weston, J.; Bengio, Y. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Machine Learning* **2014**, *94*, 233–259.
58. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmman, T.; Sun, S.; Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **2014**. doi:10.1145/2623330.2623623.
59. Socher, R.; Chen, D.; Manning, C.; Ng, A. Reasoning With Neural Tensor Networks for Knowledge Base Completion. *Advances in Neural Information Processing Systems*, 2013, pp. 926–934.
60. Liu, Q.; Jiang, H.; Ling, Z.H.; Wei, S.; Hu, Y. Probabilistic Reasoning via Deep Learning: Neural Association Models **2016**.
61. Niklaus, C.; Cetto, M.; Freitas, A.; Handschuh, S. A Survey on Open Information Extraction. *arXiv e-prints* **2018**, p. arXiv:1806.05599, [arXiv:cs.CL/1806.05599].
62. Miwa, M.; Bansal, M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. *arXiv e-prints* **2016**, p. arXiv:1601.00770, [arXiv:cs.CL/1601.00770].
63. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. *arXiv e-prints* **2017**, p. arXiv:1706.05075, [arXiv:cs.CL/1706.05075].
64. Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. Neural Relation Extraction with Selective Attention over Instances. 2016, pp. 2124–2133. doi:10.18653/v1/P16-1200.
65. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant supervision for relation extraction without labeled data. 2009, Vol. 2, pp. 1003–1011. doi:10.3115/1690219.1690287.
66. Ball, G.A. Basic assumptions and application of the theory of problem solving. **1984**.
67. Kompridis, N. So we need something else for reason to mean (Richard Rorty, non-rational semantic and cultural change) **2000**.
68. Tari, L., Knowledge Inference. In *Encyclopedia of Systems Biology*; Dubitzky, W.; Wolkenhauer, O.; Cho, K.H.; Yokota, H., Eds.; Springer New York: New York, NY, 2013; pp. 1074–1078. doi: 10.1007/978-1-4419-9863-7_166.
69. Fei, W.; Yahong, H.; Xi, L.; Qinghua, Z.; Xilin, C. Reasoning in artificial intelligence: advances and challenges. *Bulletin of National Natural Science Foundation of China* **2018**.
70. Schoenmackers, S.; Davis, J.; Etzioni, O.; Weld, D. Learning First-Order Horn Clauses from Web Text. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Cambridge, MA, 2010; pp. 1088–1098.
71. Tsarkov, D.; Horrocks, I. FaCT++ Description Logic Reasoner: System Description. *Proceedings of the Third International Joint Conference on Automated Reasoning*; Springer-Verlag: Berlin, Heidelberg, 2006; IJCAR'06, p. 292–297. doi:10.1007/11814771_26.
72. Li, Y.; Xu, B.; Lu, J.; Kang, D. Discrete tableau algorithms for FSHI. In *Proceedings of the International Workshop on Description Logics (DL 2006)*, Lake District, 2006.
73. Lao, N.; Cohen, W. Relational Retrieval Using a Combination of Path-Constrained Random Walks. *Machine Learning* **2010**, *81*, 53–67. doi:10.1007/s10994-010-5205-8.
74. Chang, K.W.; Yih, W.t.; Yang, B.; Meek, C. Typed Tensor Decomposition of Knowledge Bases for Relation Extraction. *EMNLP*, 2014. doi:10.3115/v1/D14-1165.

75. Wu, Y.; Zhu, D.; Liao, X.; Zhang, D.; Lin, K. Knowledge Graph Reasoning Based on Paths of Tensor Factorization. *Moshi Shibie yu Rengong Zhineng/Pattern Recognition and Artificial Intelligence* **2017**, *30*, 473–480. doi:10.16451/j.cnki.issn1003-6059.201705010.
76. Bordes, A.; Glorot, X.; Weston, J. Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. *International Conference on Artificial Intelligence and Statistics* **2012**.
77. Shi, B.; Wenginger, T. ProjE: Embedding Projection for Knowledge Graph Completion **2016**.
78. Xie, R.; Liu, Z.; Jia, J.; Luan, H.; Sun, M. Representation Learning of Knowledge Graphs with Entity Descriptions. AAAI; Schuurmans, D.; Wellman, M.P., Eds. AAAI Press, 2016, pp. 2659–2665.
79. Das, R.; Neelakantan, A.; Belanger, D.; McCallum, A. Chains of Reasoning over Entities, Relations, and Text using Recurrent Neural Networks. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 132–141.
80. Tagasovska, N.; Lopez-Paz, D. Single-Model Uncertainties for Deep Learning. *arXiv e-prints* **2018**, p. arXiv:1811.00908, [arXiv:stat.ML/1811.00908].
81. Xiong, W.; Hoang, T.; Wang, W. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning **2017**.
82. Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; McCallum, A. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning **2017**.
83. Li, Q.; Huang, S.; Hong, Y.; Chen, Y.; Nian Wu, Y.; Zhu, S.C. Closed Loop Neural-Symbolic Learning via Integrating Neural Perception, Grammar Parsing, and Symbolic Reasoning. *arXiv e-prints* **2020**, p. arXiv:2006.06649, [arXiv:stat.ML/2006.06649].
84. Janis, I.; Mann, L. Decision Making: A Psychological Analysis of Conflict, choice, and commitment. *The Academy of Management Review* **1980**, *5*. doi:10.2307/257816.
85. Silver, D.; Huang, A.; Maddison, C.; Guez, A.; Sifre, L.; Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. doi:10.1038/nature16961.
86. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; Hassabis, D. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv e-prints* **2017**, p. arXiv:1712.01815, [arXiv:cs.AI/1712.01815].
87. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; Lillicrap, T.; Silver, D. Mastering Atari, Go, chess and shogi by planning with a learned model **2020**. 588, 604–609, [arXiv:cs.LG/1911.08265]. doi: 10.1038/s41586-020-03051-4.
88. Pearl, J.; Mackenzie, D. The Book of Why: The New Science of Cause and Effect. *Science* **2018**, *361*, 855–855, [https://science.sciencemag.org/content/361/6405/855.2.full.pdf]. doi: 10.1126/science.aau9731.
89. Luo, Z. Commonsense Causal Reasoning between Short Texts. AAAI Press **2016**.
90. Qi, J.; Niu, Y.; Huang, J.; Zhang, H. Two Causal Principles for Improving Visual Dialog **2019**.
91. Tang, K.; Niu, Y.; Huang, J.; Shi, J.; Zhang, H. Unbiased Scene Graph Generation from Biased Training **2020**.
92. Reiter, E.; Dale, R. Building Natural-Language Generation Systems. *Natural Language Engineering* **1997**, *PP*, 57–87.
93. E. Reiter and, R. Dale. *Building Natural Language Generation Systems (Studies in Natural Language Processing)*; Cambridge University Press, Cambridge, UK, 2000.
94. Evans, R.; Piwek, P.; Cahill, L. What is NLG? Proceedings of the International Natural Language Generation Conference; Association for Computational Linguistics: Harriman, New York, USA, 2002; pp. 144–151.
95. McDonald, D. Issues in the choice of a source for natural language generation. *Computational Linguistics* **1993**, *19*, 191–197.
96. Mitchell, M.; Han, X.; Dodge, J.; Mensch, A.; Goyal, A.; Berg, A.; Yamaguchi, K.; Berg, T.; Stratos, K.; Daumé, I. Midge: generating image descriptions from computer vision detections. **2012**, pp. 747–756.

97. Kulkarni, G.; Premraj, V.; Ordonez, V.; Dhar, S.; Li, S.; Yejin, C.; Berg, A.; Berg, T. BabyTalk: Understanding and Generating Simple Image Descriptions. *IEEE transactions on pattern analysis and machine intelligence* **2012**, *35*. doi:10.1109/TPAMI.2012.162.
98. Thomason, J.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; Mooney, R. Integrating Language and Vision to Generate Natural Language Descriptions of Videos in the Wild. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers; Dublin City University and Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 1218–1227.
99. Markov, A.A. An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains. *Science in Context* **2006**, *19*, 591–600. doi: 10.1017/S0269889706001074.
100. Tran, V.K.; Nguyen, L. Natural Language Generation for Spoken Dialogue System using RNN Encoder-Decoder Networks. 2017. doi:10.18653/v1/K17-1044.
101. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.
102. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention Is All You Need **2017**.
103. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. 2018, pp. 2227–2237. doi:10.18653/v1/N18-1202.
104. Coop, R.; Arel, I. Mitigation of catastrophic forgetting in recurrent neural networks using a Fixed Expansion Layer. 2013, pp. 1–7. doi:10.1109/IJCNN.2013.6707047.
105. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. International Conference on Learning Representations, 2020.
106. Wang, S.; Li, B.Z.; Khabza, M.; Fang, H.; Ma, H. Linformer: Self-Attention with Linear Complexity. *ArXiv* **2020**, *abs/2006.04768*.
107. Choromanski, K.; Likhoshesterov, V.; Dohan, D.; Song, X.; Davis, J.; Sarlos, T.; Belanger, D.; Colwell, L.J.; Weller, A. Masked Language Modeling for Proteins via Linearly Scalable Long-Context Transformers. *ArXiv* **2020**, *abs/2006.03555*.
108. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *ArXiv* **2020**, *abs/2004.05150*.
109. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training **2018**.
110. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Amodei, D. Language Models are Few-Shot Learners **2020**.
111. Trinh, T.H.; Le, Q.V. A Simple Method for Commonsense Reasoning. *arXiv e-prints* **2018**, p. arXiv:1806.02847, [arXiv:cs.AI/1806.02847].
112. Yuchen Lin, B.; Chen, X.; Chen, J.; Ren, X. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. *arXiv e-prints* **2019**, p. arXiv:1909.02151, [arXiv:cs.CL/1909.02151].
113. Lv, S.; Guo, D.; Xu, J.; Tang, D.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; Cao, G.; Hu, S. Graph-Based Reasoning over Heterogeneous External Knowledge for Commonsense Question Answering. *arXiv e-prints* **2019**, p. arXiv:1909.05311, [arXiv:cs.CL/1909.05311].
114. Ye, Z.X.; Chen, Q.; Wang, W.; Ling, Z.H. Align, Mask and Select: A Simple Method for Incorporating Commonsense Knowledge into Language Representation Models. *arXiv e-prints* **2019**, p. arXiv:1908.06725, [arXiv:cs.CL/1908.06725].
115. Speer, R.; Havasi, C. Representing general relational knowledge in ConceptNet 5. *Proc. of LREC* **2012**, pp. 3679–3686.
116. Sap, M.; LeBras, R.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N.A.; Choi, Y. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. *arXiv e-prints* **2018**, p. arXiv:1811.00146, [arXiv:cs.CL/1811.00146].
117. Ji, H.; Ke, P.; Huang, S.; Wei, F.; Zhu, X.; Huang, M. Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); Association for Computational Linguistics: Online, 2020; pp. 725–736. doi:10.18653/v1/2020.emnlp-main.54.
118. Serafini, L.; Garcez, A. Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge **2016**.

119. Riegel, R.; Gray, A.; Luus, F.; Khan, N.; Makondo, N.; Akhalwaya, I.; Qian, H.; Fagin, R.; Barahona, F.; Sharma, U.; Ikbal, S.; Karanam, H.; Neelam, S.; Likhyan, A.; Srivastava, S. *Logical Neural Networks* **2020**.
120. G, M. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence **2020**.
121. Valiant, L. Three problems in computer science. *J. ACM* **2003**, *50*, 96–99. doi: 10.1145/602382.602410.
122. Garcez, A.S.d.; Lamb, L.C.; Gabbay, D.M. *Neural-Symbolic Cognitive Reasoning*, 1 ed.; Springer Publishing Company, Incorporated, 2009. doi:10.1007/978-3-540-73246-4.
123. Serafini, L.; D'Avila Garcez, A.S. Learning and Reasoning with Logic Tensor Networks. *Proceedings of the XV International Conference of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence - Volume 10037*; Springer-Verlag: Berlin, Heidelberg, 2016; AI*IA 2016, p. 334–348. doi:10.1007/978-3-319-49130-1_25.
124. Valiant, L. Knowledge Infusion. 2006, Vol. 2.
125. Evans, R.; Grefenstette, E. Learning Explanatory Rules from Noisy Data. *J. Artif. Int. Res.* **2018**, *61*, 1–64.
126. Lample, G.; Charton, F. Deep Learning for Symbolic Mathematics **2019**.
127. Mao, J.; Gan, C.; Kohli, P.; Tenenbaum, J.; Wu, J. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision **2019**.
128. Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; De Raedt, L. DeepProbLog: Neural Probabilistic Logic Programming **2019**.
129. Arabshahi, F.; Lu, Z.; Singh, S.; Anandkumar, A. Memory Augmented Recursive Neural Networks **2019**.
130. Arabshahi, F.; Singh, S.; Anandkumar, A. Combining Symbolic and Function Evaluation Expressions In Neural Programs **2018**.
131. Garcez, A.; Gori, M.; Lamb, L.; Serafini, L.; Spranger, M.; Tran, S. Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning **2019**.
132. Cameron, C.; Chen, R.; Hartford, J.S.; Brown, K.L. Predicting Propositional Satisfiability via End-to-End Learning. *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 3324–3331.
133. Lamb, L.; Garcez, A.; Gori, M.; Prates, M.; Avelar, P.; Vardi, M. Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective **2020**. pp. 4810–4817. doi: 10.24963/ijcai.2020/670.
134. Prates, M.; Avelar, P.; Lemos, H.; Lamb, L.; Vardi, M. Learning to Solve NP-Complete Problems - A Graph Neural Network for Decision TSP **2018**.
135. Marcus, G. GPT-2 and the Nature of Intelligence. *The Gradient* **2020**.
136. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; Lillicrap, T.; Silver, D. Mastering Atari, Go, chess and shogi by planning with a learned model **2020**. 588, 604–609, [arXiv:cs.LG/1911.08265]. doi: 10.1038/s41586-020-03051-4.
137. Lipton, Z. The Mythos of Model Interpretability. *Communications of the ACM* **2016**, *61*. doi: 10.1145/3233231.
138. Barredo Arrieta, A.; Diaz Rodriguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado González, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, V.R.; Chatila, R.; Herrera, F. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion* **2019**, *58*. doi:10.1016/j.inffus.2019.12.012.
139. White, A.; d'Avila Garcez, A. Measurable Counterfactual Local Explanations for Any Classifier. *arXiv e-prints* **2019**, p. arXiv:1908.03020, [arXiv:cs.AI/1908.03020].
140. Zhu, Y.; Gao, T.; Fan, L.; Huang, S.; Edmonds, M.; Liu, H.; Gao, F.; Zhang, C.; Qi, S.; Wu, Y.; Tenenbaum, J.; Zhu, S. Dark, Beyond Deep: A Paradigm Shift to Cognitive AI with Humanlike Common Sense. *Engineering* **2020**, *6*, 310–345.
141. Ding, X.; Li, Z.; Liu, T.; Liao, K. ELG: An Event Logic Graph. *arXiv e-prints* **2019**, p. arXiv:1907.08015, [arXiv:cs.AI/1907.08015].
142. WANG, J.; ZHANG, W.; WANG, Y.; SUN, Z. Constructing and inferring event logic cognitive graph in the field of big data. *Scientia Sinica(Informationis)* **2020**, *50*, 988–1002. doi:10.1360/SSI-2019-0273.

-
143. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments **2017**.
 144. Foerster, J.; Assael, Y.; Freitas, N.; Whiteson, S. Learning to Communicate with Deep Multi-Agent Reinforcement Learning **2016**.
 145. Foerster, J.; Nardelli, N.; Farquhar, G.; Torr, P.; Kohli, P.; Whiteson, S. Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning **2017**.
 146. Goldman, C.; Zilberstein, S. Optimizing Information Exchange in Cooperative Multi-Agent Systems. *Aamas'* **2003**, 50. doi:10.1145/860575.860598.
 147. Bernstein, D.S.; Zilberstein, S.; Immerman, N. The Complexity of Decentralized Control of Markov Decision Processes. *arXiv e-prints* **2013**, p. arXiv:1301.3836, [[arXiv:cs.AI/1301.3836](https://arxiv.org/abs/1301.3836)].
 148. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *Computer Science* **2013**.
 149. Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; Vicente, R. Multiagent Cooperation and Competition with Deep Reinforcement Learning. *arXiv e-prints* **2015**, p. arXiv:1511.08779, [[arXiv:cs.AI/1511.08779](https://arxiv.org/abs/1511.08779)].
 150. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual Multi-Agent Policy Gradients **2017**.