PrimeNet: A Framework for Commonsense Knowledge Representation and Reasoning Based on Conceptual Primitives

Qian Liu[a], Sooji Han[b], Erik Cambria[c], Yang Li[d] and Kenneth Kwok[e]

[a]School of Computer Science, The University of Auckland, New Zealand
[b]Intapp, Berlin, Germany
[c]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[d]School of Automation, Northwestern Polytechnical University, China
[e]Institute of High Performance Computing, A*STAR, Singapore

ARTICLE INFO

Keywords:
commonsense acquisition
knowledge representation
conceptual primitives

ABSTRACT

Commonsense knowledge acquisition and representation is a core topic in artificial intelligence (AI), which is crucial for building more sophisticated and human-like AI systems. However, existing commonsense knowledge bases organize facts in an isolated manner like *bag of facts*, lacking the cognitive-level connections that humans possess. Humans possess the ability to efficiently organize vast amounts of knowledge. On the one hand, individuals summarize concrete *entities* into *concepts* based on observations to express their commonalities. On the other hand, they establish connections between *concepts* and engage in reasoning based on these connections, and maintain a small core set of primitives to build cognitive blocks. This is indicated as the *conceptual primitives* theory, which reveals that humans organize knowledge by establishing a set of primitives and designing reasoning strategies based on them. Inspired by this theory, in this work, we design a new commonsense knowledge base named PrimeNet. It is constructed in a three-layer structure, i.e., primitive, concept, and entity. It is constructed by comprising a small core of primitives, linked to a much more extensive base of factual knowledge instances. First, we collect commonsense knowledge and employ a gradual expansion strategy for knowledge integration. After refinement, PrimeNet contains 6 million edges between 2 million nodes, with 34 different types of relations. Then, we design a new conceptualization method by leveraging a probabilistic taxonomy, to build the concept layer of PrimeNet. Finally, we conduct primitive detection to build the primitive layer, where a lexical substitution task is used to identify related concepts, and large language models are employed to generate a rational primitive to label each concept cluster as well as verify the primitive detection process. To verify the usefulness of PrimeNet, we utilize the knowledge in PrimeNet to improve the model performance on two downstream tasks, i.e., semantic similarity and neuro-symbolic commonsense question answering. All the data, codes, APIs, and tools that are used to leverage PrimeNet are available at https://github.com/senticnet/primenet.

1. Introduction

Commonsense knowledge refers to the information about everyday life that humans are expected to know, such as *painters use pencils* and *animals don't drive cars*. This kind of knowledge is usually taken for granted in human communication and reasoning, even though it may not be explicitly stated [1]. However, machines lack access to this innate commonsense knowledge, which often results in their inferior performance in simple reasoning tasks. As mentioned by Oren Etzioni, commonsense is *"the dark matter"* of AI: it shapes so much of what we do and what we need to do, and yet it's ineffable. To address this limitation, researchers have dedicated significant effort to construct diverse commonsense knowledge bases like Cyc [2], FrameNet [3], ConceptNet [4], TransOMCS [5], ATOMIC [6, 7], CSKG [8], and VoCSK [9]. These knowledge bases are compiled from diverse sources (e.g., encyclopedias, crowdsourcing, and expert annotations), aiming to empower machines with access to commonsense knowledge and enhance the reasoning abilities of AI systems. Despite advancements in existing knowledge bases, the reasoning capabilities of AI systems remain unsatisfactory. One notable limitation is that current knowledge bases often organize facts in a manner resembling a *"millions of facts"*, lacking the cognitive-level connections inherent in human understanding.

liu.qian@auckland.ac.nz (Q. Liu); sooji.han@intapp.com (S. Han); cambria@ntu.edu.sg (E. Cambria); liyangnpu@nwpu.edu.cn (Y. Li); kenkwok@ihpc.a-star.edu.sg (K. Kwok)
ORCID(s):

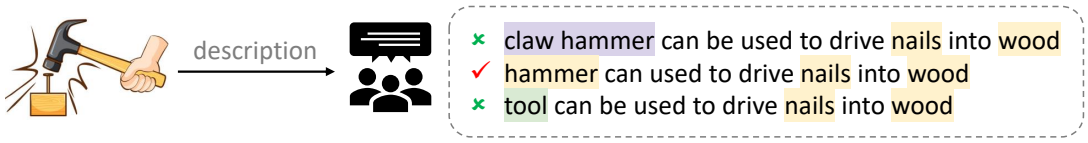


Figure 1: Example of the description of commonsense knowledge with concepts (e.g., *hammer*, *nail*, and *wood*), instead of specific entities (e.g., *claw_hammer*) or abstract primitives (e.g., `tool`).

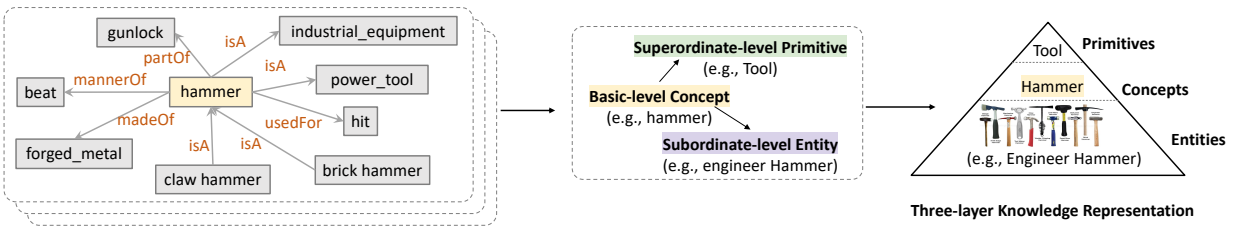


Figure 2: Illustration of three-layer structure in PrimeNet. Given the factual knowledge, a *concept* layer is generated as the basic level, comprising widely recognized mental representations associated with various categories or classes of objects. Its subordinate layer is termed as *entity* layer, which consists of specific entities, and its superordinate layer is defined as *primitive* level, encapsulating overarching and fundamental primitives.

Humans, on the other hand, exhibit the ability to efficiently organize extensive amounts of knowledge. This capability goes beyond mere accumulation of facts and involves the intricate weaving of cognitive-level connections, enabling a deeper and more nuanced comprehension of the information at hand. We have two observations for human-like knowledge organization. First, individuals are able to function well in most real-world situations using a much smaller set of *concepts*, as opposed to dealing with an exhaustive list of specific *entities*. For example, humans generally describe commonsense knowledge like *hammer can be used to drive nails into wood*, as illustrated in Fig. 1. In this example, the more general concepts such as *hammer*, *nail*, and *wood* are used for the description, rather than getting into overly specific terms like *engineering hammer* or *rubber hammer*. From estimates of effective vocabulary, the number of words that people need in order to understand 95% of everyday texts is around 3000 words, and the average size of American freshman college students' vocabulary has been estimated at about 12,000 words [10]. This underscores the human ability to distill extensive information into manageable concepts, facilitating a more streamlined expression and understanding of daily experiences.

Second, in the cognitive level, human cognition relies on a small set of fundamental and innate building blocks called *primitives*. In the *conceptual primitive* theory, the primitives serve as elemental units of information and actions, like `color`, `shape`, `size`, `increase`, and `decrease`, and forms the foundation for humans to make generalizations, inferences, and predictions, ultimately facilitating efficient reasoning and understanding in a wide range of real-world situations. For example, we generalize concepts with relevant higher-level primitives. Verb concepts such as *eat*, *slurp*, and *munch* could be related to a primitive `EAT`. Noun concepts like *pasta*, *bread*, and *milk* can be associated with the primitive `FOOD`. Therefore, *eat pasta* or *slurp milk* can be generalized into a primitive-level description, i.e., `EAT FOOD`. Hierarchical concept representations have significant applications in diverse domains, e.g., conceptual metaphor understanding [11, 12] and cognitive analysis [13].

In history, some efforts have been devoted to building knowledge bases more in line with human cognition. For example, VoCSK [9] is designed to exploit concept-level knowledge representation for implicit verb-oriented commonsense knowledge (e.g., *person eats food* instead of *John eats bread*). SenticNet [14] is developed for organizing sentiment knowledge with a core set of primitives. ASER [15] (short for Activities, States, Events, and their Relations) is built to extend the traditional definition of selectional preference to higher-order selectional preference over eventualities. These methods share a common goal of conceptualizing diverse types of commonsense knowledge, mapping them to higher-level cognition, and moving beyond the explicit representation of knowledge as discrete facts. Following this line, we take a further step by constructing a new framework for representing the intricate commonsense knowledge based on the conceptual primitive theory.

Table 1
Examples of verb primitives in PrimeNet. Given the input string, we illustrate the detected verb primitives, and its primitive-level representation and explanation. Primitives are marked in green.

Input String	Verb Primitives	Primitive-level Representation and Explanation	
turn off light	turn off → DEACTIVATE	DEACTIVATE(light)	light.STATE=ON → light.STATE = OFF
add salary	add → INCREASE	INCREASE(salary)	salary → salary++
cut budget	cut → DECREASE	DECREASE(budget)	budget → budget−−
drive car	drive → ACCELERATE	ACCELERATE(car)	INCREASE(car.SPEED) := car.SPEED++
build house	build → GENERATE	GENERATE(house)	∄ house →∃ house
butcher chickens	butcher → KILL	KILL(chicken)	TERMINATE(LIFE(chicken))
revise the manuscript	revise →FIX	Fix(manuscript)	manuscript.STATE=BAD → manuscript.STATE=GOOD
illuminate the idea	illuminate →SIMPLIFY	SIMPLIFY(idea)	idea.STATE=DIFFICULT → idea.STATE=EASY

In this work, we propose a new framework for commonsense knowledge representation and reasoning based on conceptual primitives, named PrimeNet. By mimicking the way human organizing knowledge, we design a new framework which consists of three layers, as illustrated in Fig. 2:

- **Primitive**: The primitive layer comprises fundamental and universal elements that act as the building blocks of cognition. These primitives form the foundation upon which the entire knowledge representation is constructed. Examples of basic primitives include `color`, `shape`, `size`, `object`, `tool`, `increase`, `decrease`, and others. These primitives are essential for understanding and reasoning about the world.
- **Concept**: The concept layer is commonly used mental representations of categories or classes of objects, ideas, or events that share common features or characteristics. For example, concepts like *hammer* and *nail* fall into this layer. They allow for efficient information organization and grouping based on shared attributes.
- **Entity**: The entity layer represents specific instances or examples of *concepts*. For example, given the concept *hammer*, specific entities include *brick_hammer*, *rubber_hammer*, and *engineer_hammer*. This layer enables a more specific representation of knowledge, capturing individual objects or instances in the real world.

PrimeNet is built upon the three-layer structure to systematically organize commonsense knowledge. We begin by gathering extensive commonsense knowledge from diverse sources and integrate it to form a graph. Unlike a simple aggregation of facts, we adopt a gradual expansion approach. Initially, we construct the graph with core concepts and relation types, systematically expanding it by adding more specific entities and incorporating diverse relation types. In the next stage, we establish the conceptual layer of PrimeNet, by assessing the abstractness of all nodes using a new scoring function tailored for conceptualization. We leverage the probabilistic taxonomy Probase [16] to identify the abstract concepts, and our scoring method centers around core words rather than the peripheral leaves [9, 17]. Then, we perform primitive detection on the concepts to build the primitive layer of PrimeNet. Formulating a thorough primitive set demands considerable time and effort. To address this, we design a lexical substitution task to discover the set of primitives. This is grounded in the assumption that within a shared context, the associated concepts under a primitive can be seamlessly interchanged, resulting in grammatically accurate sentences upon substitution. To allocate a representative primitive to each concept cluster, we leverage large language models (LLMs) to generate the primitive and employ an LLM-based verifier to validate the assignment of the primitive to concepts. Moreover, we manually check the primitives, refine the hierarchy structure of the primitives, and generate the explanation of primitives. For example, `DEACTIVATE` is defined as *change the status from on to off*, i.e., `STATE=ON` → `STATE=OFF`. In Table. 1, we present several cases of verb primitives used in PrimeNet. This strategy of constructing a primitive layer balances the need for human hand-coding for accuracy with that for crowd-sourcing and machine-based knowledge extraction for coverage.

The contribution of this work is summarized as follows.

1. **Representation of commonsense knowledge based on conceptual primitives.** We propose a multi-layer commonsense knowledge base based on *conceptual primitives* under the hypothesize that commonsense reasoning could depend on a concise core of concepts. To our best knowledge, this is the first work incorporating the idea of conceptual primitives into a general-purpose commonsense knowledge base to provide a generalizable, effective representation of commonsense knowledge for AI tasks.

2. **Construction of a new commonsense knowledge base PrimeNet.** Based on the designed multi-layer structure, we construct a brand new commonsense knowledge base. We first collect commonsense knowledge from various sources and perform knowledge integration to build a knowledge graph.
3. **Conceptualization for PrimeNet.** We design a new scoring method to measure the abstractness of a term for conceptualization, according to the conditional probability and connections to core words. Compared with previous methods, our method centers around core words rather than the peripheral leaves, which is effective in measuring the abstractness of concepts.
4. **Primitive Detection for PrimeNet.** We design a new primitive detection method to build the primitive layer. We employ a lexical substitution task to discover related concepts under the assumption that they share a similar context. For the clusters of related concepts, we leverage LLMs to label their primitives and verify the detection process.

The rest of the paper is organized as follows: Section 2 introduces conceptual primitive theory and the challenges of building commonsense knowledge bases; Section 4 details the construction of the graph of PrimeNet; Section 5 introduces the conceptualization for building the concept layer of PrimeNet; Section 6 introduces the primitive detection to build the primitive layer of PrimeNet; Section 7 reports experiments; Section 8 surveys existing commonsense knowledge bases, briefly introduces their features, and describes conceptual primitives to provide a motivation behind the utilization of such an approach for PrimeNet; finally, Section 9 provides concluding remarks.

2. Background

2.1. Theory of Conceptual Primitive

In linguistics and cognitive science, *conceptual primitive* commonly refers to a basic, irreducible concept or idea that serves as a foundation for understanding more complex concepts. Conceptual primitives are fundamental elements that are not further defined in terms of other concepts but are instead used to define other, more complex ideas. They are often considered to be the building blocks of thought and language. The exploration of conceptual primitives has a rich history within linguistics. In the 1950s, Chomsky [18] introduced the universal grammar theory, positing innate linguistic structures as foundational conceptual primitives. According to this theory, humans inherently possess the capacity to acquire language, with universal linguistic structures serving as fundamental building blocks shared across all languages. The conceptual dependency theory, put forth by Schank [19], suggested that the basis of natural language is conceptual, forming an interlingual foundation composed of shared concepts and relationships across languages. Jackendoff [20] delved into explanatory semantic representation, asserting the existence of semantic primitives common to all languages, enabling humans to express a diverse range of semantic information. Wierzbicka [21] emphasized that "conceptual primitives and semantic universals are the cornerstones of a semantic theory", asserting that this limited set of primitives can determine interpretations for all lexical and grammatical meanings in natural language. These theories collectively aim to identify a core set of fundamental primitives for language, facilitating the description of lexicalized concepts.

In the realm of cognitive science, theoretical studies on commonsense knowledge representation align with similar insights. Jackendoff et al. [22] highlighted a strong correlation between semantic primitives and cognitive representation. According to Pesina and Solonchak [23], the primitives studied in linguistics form the basis for the formation of a person's conceptual system, which is both unique and universal in many aspects. In this view, language emerges as a central tool for cognitive functions, including conceptualization and categorization. In the development of knowledge representation theories in cognitive science, many have been based on the idea that humans possess a core set of knowledge connecting a vast array of specific knowledge. In the early stages, Minsky [24] studied the framework for knowledge representation and introduced the concept of "frames" as a structured way to organize information about situations or objects. He proposed that humans when encountering new situations, retrieve typical knowledge from their minds. Piaget et al. [25] introduced the term "schema", representing both the category of knowledge and the process of acquiring that knowledge. The knowledge representation based on schema has also been further researched by Rumelhart and Ortony [26], Winograd [27], Bobrow and Norman [28], Johnson [29] and others. Spelke and Kinzler [30] introduced the core knowledge theory, suggesting that infants are born with "core knowledge systems" supporting basic intuitions about the world. West [31] introduced a data modeling structure divided into primitive and derived concepts, with primitive concepts serving as building blocks for other concepts. These theories collectively underscore that the core primitive set constitutes the fundamental structure of human cognition and provides guidance for knowledge representation.

2.2. Challenge

In modern large-scale commonsense knowledge bases, there have been relatively few attempts to build a knowledge base in a way incorporating core primitives based on the conceptual primitive theory and linking a vast amount of facts. Cambria et al. [14] has developed a sentiment analysis system based on primitives such as `DECREASE` and `INCREASE`. Wachowiak and Gromann [32] proposed to build on large multilingual pre-trained language models and a small dataset of examples from image schema literature to train a supervised classifier that classifies natural language expressions of varying lengths into image schemas. Liu et al. [9] designed conceptualization for verbs and built a knowledge base with conceptual verb-oriented knowledge to represent various instances, e.g., *"John eat apple"* and *"Helen eat bread"* are represented as `"people eat food"`.

The primary challenge hindering progress in this field stems from the complexity of constructing a comprehensive set of core primitives to encompass extensive knowledge across diverse domains. On the one hand, managing large-scale factual data makes manual editing and maintenance of a core primitive set impractical. While it is possible to manually craft a small, high-quality core primitive set, this approach becomes intricate when using primitives to interpret other specific concepts, and its coverage of specific knowledge is limited. On the other hand, primitives are not fixed but rather flexible and adaptable. The core primitives are deeply embedded in the human conceptual system, which is both unique and universal in many aspects. The proposed number of semantic primitives varies significantly, ranging from a few units in some studies [21, 22] to several dozens [21] or even hundreds [14] in others. Pesina and Solonchak [23] stated that the main concepts of human society remain relatively stable, but their overall volume changes over time.

3. Framework

In this section, we first introduce the task definition. Then, we introduce the solution of constructing PrimeNet and the key ideas of each module.

3.1. Task Definition

PrimeNet is a hybrid graph $\mathcal{H}$ combining a traditional graph $\mathcal{G}$ where each edge is built among nodes to represent commonsense knowledge in triplets, and a hypergraph $\mathcal{G}^*$ where each edge is built over the nodes to linked their concepts and primitives. For example, in the graph $\mathcal{G}$, its edge is represented as a triplet like (*corgi*, *isA*, *dog*), where *dog* and *corgi* are nodes, and *isA* is a relation type. In the hypergraph $\mathcal{G}^*$, *corgi* is linked to *dog* in the concept layer, and *dog* is linked to `ANIMAL` in the primitive layer. We devise the formal definition of PrimeNet as below.

Definition 1 (PrimeNet). *PrimeNet is a hybrid graph $\mathcal{H}$ of a knowledge graph $\mathcal{G}$ and a hypergraph $\mathcal{G}^*$. The knowledge graph is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{R}\}$ where $\mathcal{V}$ is a node set, $\mathcal{E}$ is an edge set connecting pairs of nodes, and $\mathcal{R}$ is the set of distinct relation types associated with the edges in $\mathcal{E}$. Each node $v \in \mathcal{V}$ is a term. Each edge $e \in \mathcal{E}$ is a triplet (v_i, r, v_j) where v_i and v_j are the connected nodes, and $r \in \mathcal{R}$ is the relation type. The hypergraph is denoted as $\mathcal{G}^* = \{\mathcal{V}, \mathcal{C}, \mathcal{P}, \mathcal{M}\}$, where $\mathcal{V}$ represents the set of entities, $\mathcal{C}$ represents the set of concepts, and $\mathcal{P}$ represents the set of primitives. The hyperedge set $\mathcal{M} = \{\mathcal{M}_{v\to c}, \mathcal{M}_{c\to p}\}$ contains two types of hyperedges. The hyperedge $(v, c) \in \mathcal{M}_{v\to c}$ links the entity $v \in \mathcal{V}$ to its concept $c \in \mathcal{C}$, and the hyperedge $(c, p) \in \mathcal{M}_{c\to p}$ links the concept $c \in \mathcal{C}$ to its primitive $p \in \mathcal{C}$. Overall, we have the PrimeNet $\mathcal{H} = \{\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{C}, \mathcal{P}, \mathcal{M}\}$.*

3.2. PrimeNet Construction

The solution of PrimeNet mainly consists of three modules. The first is to construct the knowledge graph $\mathcal{G}$ to organize the large-scale commonsense knowledge. The second is a conceptualization module to identify the concept set $\mathcal{C}$ and build the hyperedges $\mathcal{M}_{v\to c}$ to link entities to concepts. The third is a primitive detection module to build the core primitive set $\mathcal{P}$, and build the hyperedges $\mathcal{M}_{c\to p}$ to link the concepts to their primitives.

Module-1: Knowledge Graph Construction. Over the course of many years, a vast reservoir of factual knowledge has accumulated, taking on various forms and originating from diverse sources. In order to systematically organize this wealth of knowledge, we have undertaken the construction of a knowledge graph. Drawing inspiration from the theory of cognitive development put forth by Piaget et al. [25], which posits that human cognitive development occurs in stages, we have adopted a gradual expansion strategy to build our knowledge repository. Rather than merging disparate sources abruptly, our approach is to delicately expand the knowledge base.

The fundamental idea underlying our strategy is that human knowledge acquisition follows a pattern of continuous expansion, rooted in commonly shared and widely accepted information. To illustrate, individuals typically begin by learning that a "hammer" is a "tool" used for driving "nails," and subsequently delve into more intricate details, such as discerning the differences among various types of hammers, such as the "engineer hammer" or "brick hammer". To emulate this cognitive process, we initially construct a basic graph consisting of widely used concepts and relations. Subsequently, we systematically enlarge the graph by incorporating a multitude of facts from diverse sources. This method allows for the gradual incorporation of information, mirroring the incremental nature of human knowledge acquisition. We detail this module in Section 4.

Module-2: Concept Detection. To construct the concept layer over the knowledge graph $\mathcal{G}$, this module focuses on identifying a suitable concept set $\mathcal{C}$ from the node set $\mathcal{V}$ and establishing hyperedges in the set $\mathcal{M}_{v\to c}$ to link entities with their respective concepts. Within PrimeNet, this concept layer encapsulates commonly used mental representations of categories, classes, or ideas that share common features or characteristics. Consequently, we initialize the concept set layer using Core WordNet[1], a compilation of approximately 5000 of the most commonly used words meticulously curated by experts. Then, we design a concept detection method to discover new concepts and expand the concept set, leveraging a large-scale probabilistic taxonomy, i.e., Probase [16], and build the edges to link entities to the detected concepts.

Specifically, Probase encompasses 33.4 million *isA* triples between 2.7 million concepts, automatically extracted from 1.68 billion web pages, with each triplet associated with a frequency score. Our observation underscores that, for a concept, its hyponyms tend to establish robust connections with diverse concepts in a probabilistic taxonomy, whereas a specific entity is more concentrated in its connection to concepts. To capture this regularity, we introduce a novel scoring function designed to identify whether a term qualifies as a concept. In contrast to alternative conceptualization methods, our approach stands out by centering around core words rather than initiating from the leaves of an extensive taxonomy for concept detection. The pre-defined core words enhance diversity and accuracy, distinguishing our strategy as effective in steering clear of misleading information stemming from isolated graphs or incorrect circles within the large-scale taxonomy.

Module-3: Primitive Discovery. This module is dedicated to constructing the primitive layer of PrimeNet, involving the establishment of a core primitive set $\mathcal{P}$ and the creation of the hyperedge set $\mathcal{M}_{c\to \mathrm{p}}$ to connect concepts with their higher-level primitives. For instance, the primitive `INCREASE` is associated with concepts like *ramp up, go up, broaden, step up, elevate, supplement, redouble, pile up, upward spiral, distend*, and more. The manual definition of the primitive set and linking of primitives to their lower-level concepts is impractical. In our approach, an automated method is designed, utilizing concept clustering and subsequent labeling of their primitives using large language models, followed by error checking to refine both the primitives and concept clusters.

Specifically, it is observed that concepts under the same primitive often share a similar meaning and context. For instance, *elongate* and *stretch* fall under the same primitive `GROW` and share a similar context. Although intuitive, lexical substitution tends to overlook crucial differences between concepts. For example, verbs such as *stretch* and *compress* belong to opposite primitives, `GROW` and `SHRINK` respectively, yet can be identified within similar lexical contexts. To address this issue, we leverage powerful LLMs to filter out incorrect concepts within each cluster, generating a primitive that accurately describes the concept cluster. Manual checks are also employed to ensure the quality of primitives in building the primitive layer. This strategy strikes a balance between human hand-coding for accuracy and crowd-sourcing and machine-based knowledge extraction for comprehensive coverage.

4. Knowledge Graph Construction

In this section, we detail the construction of the knowledge graph ($\mathcal{G}$) of PrimeNet. It mainly contains four stages. First, *commonsense knowledge acquisition* is to collect high-quality knowledge from diverse sources which are created through manually annotated or crowd-sourcing. Then, *knowledge integration* is to map the nodes and relations among different sources. Next, the *graph construction* is to organize the knowledge in a graph. Finally, *exploration* is to define functions to leverage the knowledge graph in the downstream tasks. We detail each stage as follows.

[1] Please find more details from `https://wordnet.princeton.edu/`. Core WordNet is available in `https://wordnetcode.princeton.edu/glosstag.shtml`.

Table 2
Sources of commonsense knowledge for building the knowledge graph of PrimeNet. *Creation* denotes the construction methods, *Relations* denotes the number of relation types, and *Size* denotes the graph scale.

Source	Creation	# Relation Types	Size	Example
WordNet	manual	10	155K words, 176K synsets	(denied, morphy, deny)
FrameNet	manual	10	1.2K frames, 12K roles, 1.9K edges	(Criminal_process, Subframe, Arrest)
Roget	manual	2	72k words, 1.4M edges	(explore, Synonym, investigate)
ConceptNet	crowd-sourcing	34	8M nodes, 21M edges	(keyboard, part of, computer)
Wikidata	crowd-sourcing	6.7K	75M objects, 900M edges	(George Washington, isInstanceOf, human)
DBpedia	crowd-sourcing	53.1K	4.8M nodes, 62M edges	(Applied_Artificial_Intelligence, discipline, Artificial_intelligence)
ATOMIC	crowd-sourcing	9	300K nodes, 877K edges	(PersonX bakes bread, Before X needed to, buy the ingredients)
Visual Genome	crowd-sourcing	42.4K	3.8M nodes, 2.3M edges, 2.8M attributes	(man, sit on, bench)

4.1. Commonsense Knowledge Acquisition

In constructing a commonsense knowledge base, the acquisition of knowledge stands out as a pivotal initial phase. Collecting commonsense knowledge is a challenging task due to its sheer volume, implicit nature, and diverse forms of expression. With decades of human efforts, a wealth of commonsense knowledge has been amassed and stored in various knowledge bases. To ensure quality, in this work, we extract knowledge from expert-crafted databases and crowd-sourced repositories, as summarized in Table 2, including:

- Lexical knowledge extracted from WordNet [33], FrameNet [3], and Roget [34].
- Factual knowledge extracted from ConceptNet[2] [4], which is a commonsense knowledge that represents general knowledge and commonsense relationships between concepts.
- Structured information in Wikidata and DBpedia. For DBpedia[3] [35], we extract knowledge from *InfoBoxes* which provide information about a wide variety of topics, e.g., people, places, and organizations, as well as knowledge from *InstanceTypes* which contains instances of 438 types, e.g., book, company, city, and plant.
- Task-specific knowledge, such as inferential knowledge extracted from ATOMIC [6, 7] which is organized as typed "if-then" relations with variables, and visual knowledge extracted from Visual Genome [36].

4.2. Knowledge Integration

In commonsense knowledge graph construction, multiple sources can provide complementary knowledge of different types. However, the integration of knowledge from diverse sources is impeded by the varying representation formats. It is noted that many databases provide mappings to other databases, e.g., ConceptNet contains mappings to DBpedia, WordNet, Wikidata, and FrameNet. Yet, these mappings may be incomplete. Recent research endeavors to create high-quality mappings among different knowledge bases, offering a pathway for knowledge integration. For example, CommonSense Knowledge Graph (CSKG) [8] construct mappings across seven knowledge bases (i.e., ATOMIC, ConceptNet, FrameNet, Roget, Visual Genome, Wikidata, and WordNet). We conduct knowledge integration to build a knowledge graph of PrimeNet using these high-quality mappings, as well as lexical-level and semantic-level matching methods. Table 3 summarizes the details of our integration process.

First, we process the individual sources. More specifically, we keep the initial sets of nodes, edges, and relations in ConceptNet and ATOMIC. For other sources, we extract their nodes and edges and convert their relations to the format of relations in ConceptNet, as detailed in Table 3. Then, we conduct mappings between sources for node resolution. On the one hand, we leverage mappings released by Ilievski et al. [8][4] to map nodes from different sources. On the other hand, we represent each node using its label and use exact lexical matching to establish the mappings of nodes from different sources. Moreover, we conduct semantic-level matching to identify the same nodes with different labels.

[2]We use the ConceptNet version 5.7.0, which is available at https://github.com/commonsense/conceptnet5/wiki/Downloads.
[3]We use the DBpedia version 2022.09.01, which is available at https://www.dbpedia.org/resources/.
[4]The project description and mappings are available on https://github.com/usc-isi-i2/cskg. Please refer to Ilievski et al. [8] for more details on processing individual sources, performing node resolution, and constructing mappings.

Table 3
Details of knowledge integration for individual sources and mapping between sources. Relation types are in italics. * denotes the processed nodes, edges, or mappings released by Ilievski et al. [8].

Step 1. Individual Sources	
ConceptNet	Initial nodes and edges are used, and 34 relations are mapped to PrimeNet relations (e.g.,*/r/IsA* is converted to *isA*, */r/UsedFor* is converted to *usedFor*).
ATOMIC	Initial nodes, edges, and 9 relations.
WordNet	*Hyponym* and *hypernym* are converted to *isA*, *part holonymy* is converted to *partOf*, *substance meronymy* is converted to *madeOf*.
FrameNet*	Four types of nodes are used (i.e., frames, frame elements, lexical units, and semantic types) and 19 relations are mapped to PrimeNet relations (e.g., *is_causative_of* is converted to *cause*).
Roget	Two relations are used, i.e., *synonyms* and *antonyms* are mapped to the PrimeNet relations *synonym* and *antonym*, respectively.
Visual Genome*	The image objects are converted to WordNet synsets. The relationships between objects are mapped to the relation *locatedNear*. Object attributes are represented by different relations, conditioned on their part-of-speech, i.e., *capableOf* for verbs and *mayHaveProperty* for adjective attributes.
Wikidata*	101K statements in Wikidata-CS subset are used, and the relations are manually mapped to 15 relations.
DBpedia	The instance-types subset and infobox-properties subset are used, and *#type* relation is converted to PrimeNet relation *instanceOf*.
Step 2. Mapping Between Sources	
WordNet-WordNet*	Align ConceptNet and Visual Genome using WordNet InterLingual Index (ILI) generating 117,097 mappings.
WordNet-Wikidata*	Generate links between WordNet synsets and Wikidata nodes using pre-trained XLNet model for embeddings. Manual validation with 17 students. Keep 57,145 validated edges.
FrameNet-ConceptNet*	Link FrameNet lexical units to ConceptNet nodes through Predicate Matrix (3,016 edges). Use 200k hand-labeled sentences from FrameNet corpus for additional linking.
Lexical Matching*	Establish links between nodes in ATOMIC, ConceptNet, and Roget through exact lexical matching of labels.
Semantic Matching	Establish links between nodes in ConceptNet, Wikidata, and DBpedia through semantic matching of labels.

We convert all labels of nodes to embeddings using pre-trained Sentence-BERT [37][5]. Subsequently, we employ the labels of nodes from another source as queries and perform embedding-based semantic search. The cosine similarity metric is employed to measure semantic similarities between two nodes. We establish a link between the query and its top-1 similar node if they share the same representation after lexical tokenization using NLTK[6].

4.3. Graph Construction

Confronted with an extensive dataset of knowledge triplets, creating a graph by incorporating all of them directly is a blunt method. Humans develop core conceptual primitives grounded in the most frequently utilized knowledge. For example, in the realm of *geography*, individuals effortlessly understand fundamental concepts like *country*, *continent*, and *ocean*, forming a foundational understanding without the need to memorize every specific detail, including aspects like the area and visual representation of each country available in DBpedia and Visual Genome, respectively. This insight guides our approach to graph construction through a gradual expansion strategy. We illustrate the construction process in Fig. 3. Initially, we start from core nodes and relations to construct a new knowledge graph. For core nodes, Core WordNet[7], which contains the most frequently used 5,000 words, i.e., 3,300 nouns, 1,000 verbs, and 700 adjectives. We mainly consider knowledge from WordNet and ConceptNet, with a set of core relations: isA, madeOf, partOf, mannerOf, usedFor, and capableOf. We denote this graph as a basic graph, which contains 488,216 nodes and 962,228 edges. Then, we extract instanceOf and isA relations from DBpedia to expand the core graph with more specific nodes. In this step, we employ an embedding-based semantic similarity method using pre-trained Sentence-BERT for mapping. After integration, the graph is expanded to 1.4M nodes and 3M edges.

[5]Used version: https://huggingface.co/sentence-transformers/all-mpnet-base-v2.
[6]https://www.nltk.org/
[7]https://wordnetcode.princeton.edu/standoff-files/core-wordnet.txt

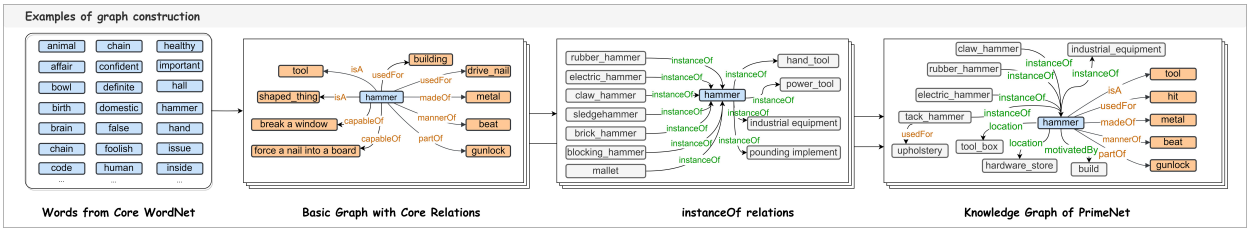


Figure 3: Illustration of graph construction of PrimeNet. Starting with Core WordNet, we first construct a basic graph with core words and relations from WordNet and ConceptNet. Then, we add instanceOf knowledge from DBpedia and Wikipedia. Next, diverse types of knowledge from other knowledge bases are incorporated into the graph of PrimeNet.

Table 4
Relations defined in the basic graph of PrimeNet, and their description, example, and mappings to WordNet and ConceptNet.

Relations	Description	Example	Mapping to WordNet	Mapping to ConceptNet
isA	A is a specific instance of B	(car, isA, machine)	hyponym, hypernym	/r/IsA, /r/InstanceOf
madeOf	A is made of B	(car, madeOf, metal)	meronymy	/r/MadeOf
usedFor	A is used for B; the purpose of A is B	(hammer, usedFor, hit)	-	/r/UsedFor
partOf	A is a part of B	(gunlock, partOf, gun)	holonymy	/r/PartOf
mannerOf	A is a specific way of B	(screw, mannerOf, revolve)	-	/r/MannerOf
capableOf	Something that A can typically do is B	(bowl, capableOf, hold_water)	-	/r/CapableOf

Finally, we integrate commonsense knowledge from diverse sources into our graph, ensuring a wide-ranging and diverse coverage. To map nodes from other sources to our graph, we employ the mappings developed by CSKG for integration. Moreover, to merge nodes, we use the embedding-based similarity method to identify nodes with the same meaning, and then use the tokenization-based method for verification. After integration, the nodes in PrimeNet are enriched with different kinds of commonsense knowledge, with 2.04M nodes and 6.03M edges.

4.4. Exploration

Then, we design multiple functions for exploring the graph that are capable of:

- Exploring graph structure of PrimeNet. For example, *nodes* and *edges* functions are designed to generate all concepts and relations in PrimeNet, respectively, and *get_number_of_nodes* and *get_number_of_edges* are designed to count the number of nodes and edges in the knowledge graph.
- Exploring commonsense knowledge for specific concepts. For example, given a concept, *what_is* function is designed to get all its relations, *get_polarity* function is used to get its sentiment polarity, and *find_path* function is designed to find a specific path in PrimeNet given a pair of concepts.
- Integrating new knowledge into PrimeNet. For example, the *add_node* and *add_edge* functions are designed to add new concepts and relations into PrimeNet, and the *add_primenet_new* function is able to incorporate a new knowledge base into PrimeNet.

We detail all the designed functions in Table 5, including their input, output, and description. These functions make it easy to apply PrimeNet in downstream tasks, as well as update PrimeNet with new commonsense knowledge or domain-specific knowledge.

5. Concept Detection

To create the concept-level of PrimeNet, we conduct concept detection to identify concepts that represent categories or classes of objects, ideas, or events based on shared features or characteristics. An intuitive approach is to use the *isA* relation to establish mappings between concepts and entities. For example, *(dog, isA, animal)*, *(cat, isA, animal)*, and *(lion, isA, animal)* indicate that *animal* is a concept, and *dog*, *cat*, and *lion* are entities falling under that concept.

Table 5
Functions designed for exploring PrimeNet. For each function, we introduce its input, output, and description.

Function	Input	Output	Description
nodes	-	a list of nodes	Return all nodes in PrimeNet.
edges	-	a list of edges	Return all edges in PrimeNet.
get_number_of_nodes	-	an int number	Return the number of nodes in PrimeNet.
get_number_of_edges	-	an int number	Return the number of edges in PrimeNet.
relation_types	a node	a list of relation types	Return all relation types that the node involved.
what_is	a node	a path of the node	Return the first edge of a node.
what_can_be	a node	a list of edges	Return all edges of a node.
relation_exist	a node and a relation type	True or False	If a relation type exists in the node return True, else False.
get_node_with_relation	a node and a relation type	a node	Given a node A and a relation R, return node B if there is an edge (A, R, B).
explain	a node and a relation type	a chain of this node	Return the chain of a node and a relation type.
generalize	a node	a list of edges	Return the root node of each of its relationships.
get_similarity	two nodes	a float score	Return a score that denotes how similar two nodes are, based on the path similarity computed by SequenceMatcher.
get_polarity	a node	Positive or Negative	Return the sentiment polarity of a node.
get_path	start_node and end_node	a path	Return a path from the start_node to the end_node .
find_last_nodes	a node	a list of paths	Return all edges where the end_node is the given node.
find_all_paths	start_node and end_node	a list of paths	Return all paths from start_node to end_node.
get_node_degree	a node	a number	Return the number of edges which connect with the given node.
get_phonetic	a concept	the phonetic information	Return the phonetic information of a concept.
add_node	a node	-	Add a node to PrimeNet if it does not exist in PrimeNet.
add_edge	an edge	-	Add an edge to PrimeNet.
add_primenet_new	a new knowledge graph	-	Add a new knowledge graph to PrimeNet.
print_to_file	a knowledge graph	-	Save a knowledge graph to a file.

Though simple, in practice, it is sub-optional to identify concepts by checking whether exist entities fall under them. For example, *animal*, *dog*, and *cargi* have specific entities. However, only *animal* and *dog* are widely-used as concepts in human daily reasoning, *cargi* are too specific. In this section, we study how to conduct concept detection with appropriate abstractions.

5.1. Preliminaries

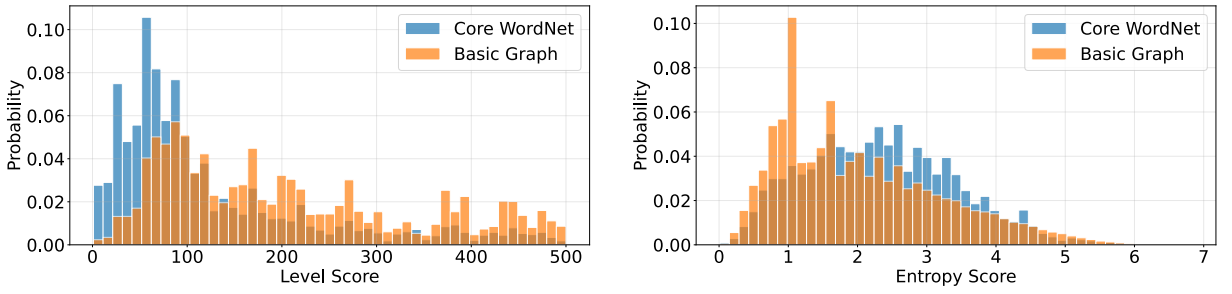
When considering the conceptualization, it is important to measure the abstractness of a term. For example, *person* is a more abstract concept compared with *student*. Given a graph with *isA* relation, it is observed that abstract terms are usually located at the higher levels in a graph, while the specific terms tend to be positioned at the lower levels Liu et al. [9]. Specifically, the leaf nodes are regarded as the most specific terms, and they are considered as first level. The level of non-leaf nodes defined as the length of the longest path from the leaf nodes to itself. Formally, the *level* of a term is defined as following.

Definition 2 (Level Score). *Given a term c, the level score of c is defined as:*

$$level(c) = \begin{cases} \max\limits_{c' \in hypo(c)} level(c') + 1, & if \quad hypo(c) \neq \phi \\ 1, & otherwise \end{cases} \tag{1}$$

where $hypo(c)$ is a set of hyponyms of c, and ϕ denotes an empty set.

The abstract words have higher level scores and specific terms have smaller level scores. For example, the level scores of *dog*, *mammal*, and *animal*, are 72, 89, and 362, respectively.



(a) Probabilities of Level Scores

(b) Probabilities of Entropy Scores

Figure 4: Illustration of data distribution of Core WordNet and the graph of PrimeNet, considering of the level scores and entropy scores of nodes.

It is also observed that, for an abstract term, its hyponyms are usually positioned at diversified levels, while its hyponyms would be more concentrated for a specific term. Based on it, Liu et al. [9] defined an entropy-based metric for the abstractness measurement. Formally, the entropy score of a term is defined as following.

Definition 3 (Entropy Score). *Given a term c, its entropy score is defined as:*

$$entropy(c) = \begin{cases} 0, & if\ c\ is\ a\ leaf\ term \\ -\sum_{i=1}^{l} p_i(c) \cdot \log p_i(c) & otherwise \end{cases} \tag{2}$$

where l is the maximum level, and $p_i(c)$ is the ratio of the number of c's hyponyms at the i-th level to the total number of c's hyponyms.

The entropy of abstract terms is often greater than that of specific terms. For example, the entropy scores of *pupil*, *student*, and *people* are 0.563, 0.927, and 1.790, respectively.

In general, abstract *concepts* and concrete *entities* are differentiated using these abstractness measure methods by manually-defined thresholds [9]. However, these methods are inaccurate and not suitable when applied to complex graphs with large-scale commonsense knowledge. The primary reason is the vast amount of knowledge, inevitably leading to the presence of cycles and isolated subgraphs, significantly reducing the accuracy of the aforementioned methods. Furthermore, some commonly used vocabulary lacks numerous lower-level nodes, e.g., *voice*, *track*, and *driver*, and they have lower scores compared with other words with more hyponyms, e.g., *transport*, *symbol*, and *medicine*. As such, the conceptualization methods which only rely on hierarchical information are not reasonable for such cases.

5.2. Conceptualization

Previous methods employed a *bottom-up* approach to measure abstractness, where a word's score relies on its hyponym set. Leaves without hyponyms are initiated as the seed set and then inferred for the others. In this work, we initialize the core concepts and then infer other words accordingly.

Specifically, the initial set of concepts, denoted as $\mathbb{C}^0 = \{c_1, c_2, c_3, \cdots\}$, comprises commonly used words from Core WordNet that describe the world in human daily life. In an ideal scenario, the hypernyms of these core words are expected to be more abstract and should be considered as concepts. However, in practical scenario, not all of their hypernyms can be unequivocally regarded as concepts due to the intricate interweaving of commonsense knowledge. For instance, relationships such as *(dog, isA, animal)*, *(dog, isA, pet)*, *(pet, isA, animal)*, and *(dog, isA, species)* are all deemed correct and coexist within the knowledge base. Thus, we need a more accurate method to measure the abstractness of hypernyms. It is observed that not all hypernyms have the same weight when working as the concept of a *dog*. This problem has been deeply studied, and a large-scale probabilistic taxonomy, i.e., Probase [16], has been constructed to provide statistical insights of isA relations. It includes "isA" relations for 2.7 million terms, automatically mined from a corpus of 1.68 billion web pages. That is, each triplet *(t, isA, c)* is linked to a frequency score $frec(t, c)$, providing frequency information computed through a data-driven method based on large-scale corpus.

For example, *(dog, isA, animal)* and *(dog, isA, species)* show that both *animal* and *species* are concepts of *dog*, and $freq(dog, animal) > freq(dog, species)$ shows *animal* is a more typical concept for *dog*, compared with *species*. Given a triplet *(t, isA, c)*, it is associated with a frequency score $freq(t, c)$ in Probase. The frequency score is an important signal to identify whether this relation is typical or not. Based on this observation, Wang et al. [17] propose a *typicality score*, which is defined based on the frequency information to tell how popular a concept c is as far as an entity t is concerned, and how popular an entity t is as far as a concept c is concerned:

Definition 4 (Typicality Score). *Given an term t, the conditional probability $Pr(c|t)$ of a term c is defined as:*

$$Pr(c|t) = \frac{freq(t, c)}{\sum_{c_i \in hyper(t)} freq(t, c_i)}, \tag{3}$$

where $hyper(t) = \{c_1, c_2, c_3, \cdots\}$ is the set of hypernyms of t.

Given a concept c, the conditional probability $Pr(t|c)$ of an entity t is defined as:

$$Pr(t|c) = \frac{freq(t, c)}{\sum_{t_i \in hypo(c)} freq(t_i, c)}, \tag{4}$$

where $hypo(c) = \{t_1, t_2, t_3, \cdots\}$ is the set of hyponyms of c.

It is observed that a terms tends to be abstract when it is strongly connected with multiple concepts. Continuing the previous example, the term *animal*, *pet*, *species* link to 98, 435, 22 concepts in $\mathcal{C}^0$, respectively. To formalize this regularity, an linking-based metric is designed as follows:

Definition 5 (Conceptual Score). *Given a term w and a set of concepts $\mathcal{C}$, the conceptual score of w is defined as:*

$$abstract(w) = \sum_{t_i \in hypo(w)} \mathbb{1}(t_i \in \mathcal{C}) * \frac{freq(t_i, w)}{\sum_{o_j \in hyper(t_i)} freq(t_i, o_j)} \tag{5}$$

where $hypo(w) = \{t_1, t_2, \cdots, t_i \cdots\}$ is the set of hyponyms of w, $hyper(t_i) = \{o_1, o_2, \cdots, o_j \cdots\}$ is the set of hypernyms of t_i, and $\mathbb{1}(t_i \in \mathcal{C})$ is set to 1, otherwise 0.

This scoring method is designed to quantify the extent to which a term functions as a universal, abstract link across a diverse array of concepts. Utilizing the initial set $\mathcal{C}^0$, we calculate the abstraction scores of their hypernyms, presenting the top 50 terms in Fig. 6. According to human analysis, all of them are confirmed as conceptual terms. In addition, we present their *level scores* and *entropy scores*, revealing that these metrics fall short in inferring them as abstract terms. For instance, *topic*, *song*, and *adjective* exhibit low level scores (i.e., 3, 3, and 28), and *author* and *classic* display low entropy scores (i.e., 0.59 and 1.72), excluding them from being identified as concepts.

We employ an iterative approach to augment the concept set by systematically incorporating terms with high abstraction scores. In i-th iteration, we introduce the top-n (e.g., $n = 3$) hypernyms for each concept in $\mathcal{C}^{i-1}$. The constraint imposed is that these hypernyms must surpass a specified threshold T_{abs}. This process results in the construction of an updated concept set, denoted as $\mathcal{C}^i$.

6. Primitive Discovery

The primitive discovery is to identify the most basic and essential element of the world knowledge, which provides a way to represent and organize knowledge in a structured and meaningful manner [38, 39]. The well-designed primitive set can help to produce more accurate, scalable, and reusable knowledge bases. However, creating a thorough set of primitives is extremely time-consuming and labor-intensive, hence it is not generally employed in most knowledge bases [24, 20, 38, 14]. In this work, we apply automatically discover a primitive set for commonsense knowledge. The main idea is concept clusters that are semantically connected and have a similar lexical function, and then label each cluster as a conceptual primitive.

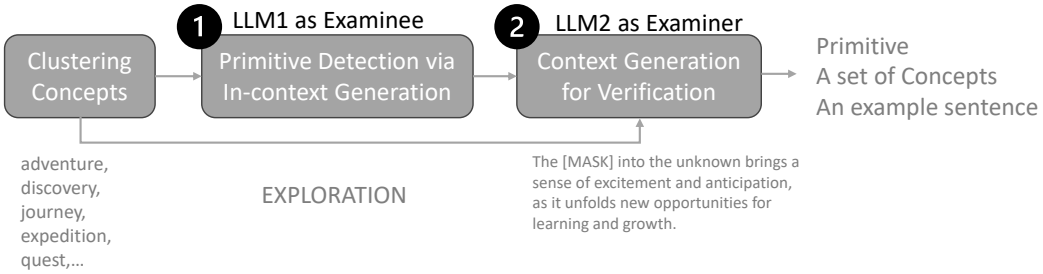


Figure 5: The overall framework for primitive detection. LLM1 is used as an examinee to generate representative primitive for each concept cluster, and LLM2 is used as an examiner to verify the primitive and its related concepts.

6.1. Concept Clustering

To achieve this goal, we employ a lexical substitution task which is to replace a concept in a sentence with a different concept, and if the grammatical structure and overall meaning of the sentence are preserved, these two concepts are considered to have similar meanings. For example, in the sentence "the landlord tried to eject the tenants for not paying rent on time", one could substitute the word "eject" with "dispossess", "remove", "oust", or "evict" without changing the overall meaning of the sentence. They are clustered and assigned with a label as their primitive, i.e., EXPEL.

We implement the primitive discovery method by fine-tuning pre-trained language models using the lexical substitution task. Specifically, there are mainly three steps.

1. Training Data. We extract all the verb-noun and adjective-noun concepts from ConceptNet 5.7 [4] together with a sample sentence for each concept. The collection of concepts is denoted as $\mathcal{E} = \{e_1, e_2, e_3, \cdots, e_n\}$, where each concept $e_i \in \mathcal{E}$ is assigned with a sample sentence s_i. For each concept e_i, we remove it from the sentence s_i and the remaining sentence is denoted as its context c_i. We employ pre-trained language models to represent the concept e_i and its context c_i as fixed-dimensional embeddings, i.e., $\mathbf{e}_i$ and $\mathbf{c}_i$, respectively.

2. Training Objective. Then, we fine-tune the pre-trained language model with a lexical substitution task. The assumption is that a relevant lexical substitute should be both semantically similar to the target word and have a similar contextual background. Given a concept e_i, its context c_i is regarded as the positive example. We create negative examples by sampling random concepts, which are denoted as $\mathcal{N}(e_i) = \{e^*_{i,1}, e^*_{i,2}, \cdots, e^*_{i,z}\}$. The training objective function is defined as:

$$O = \sum_{i=1}^{n} (\log(\sigma(\mathbf{e}_i, \mathbf{c}_i)) + \sum_{e^*_{i,j} \in \mathcal{N}(e_i)} \log(\sigma(-\mathbf{e}^*_{i,j}, \mathbf{c}_i))), \tag{6}$$

where n is the number of training examples, z is the number of negative words for each example, and $\mathbf{e}^j_i$ denotes the representation of a negative concept. After fine-tuning, the representation model is expected to map concepts and context into a embedding space, where concepts that are appropriate for a given context are located close to one another.

3. Semantic Measure. We design a semantic measure to find the replacement of the concept in the embedding space. Given a concept e_i and its sentential context c_i, we calculate the cosine distance of all the other concepts, e.g., $w \in \mathcal{E}$ in the embedding space as:

$$Sim(\mathbf{w}, (\mathbf{e}_i, \mathbf{c}_i)) = cos(\mathbf{w}, \mathbf{e}_i) \cdot cos(\mathbf{w}, \mathbf{c}_i) \cdot cos(\mathbf{s}_i, \mathbf{s}^w_i), \tag{7}$$

where s_i is the original sentence, and s^w_i is a sentence by replacing c_i in s_i with w. The list of potential lexical substitutions is generated by ranking candidate concepts according to the designed measure. As such, we generate the concept clusters.

6.2. Primitive Detection

The primitive detection involves detecting the errors in each cluster, and associating a meaningful and generalizable primitive with a cluster of related concepts. For example, the concepts like *ingest*, *slurp*, *munch* are represented by a primitive EAT. It is inherent to human nature to try to categorize things, events and people, finding patterns and forms they have in common.

Table 6
Accuracy (%) assessed by human annotators. *Size* denotes the number of triplets in different knowledge bases.

Knowledge Bases	Size	Accept	Reject	No Judgment
TransOMCS	18.5M	41.7	53.4	4.9
ATOMIC	877K	88.5	10.0	1.5
ConceptNet	21M	88.6	7.5	3.9
PrimeNet	6M	**92.4**	5.2	2.4

In this work, we explore the generative ability of large language models (LLMs) for primitive detection. To ensure the accuracy, as illustrated in Fig. 5, we design a detection-verification framework, where the first LLM works as examinee to generate primitive for the a concept cluster, and another LLM works as examiner to check whether the generated primitive is correct. Specifically,

Step-1: Primitive Detection by Examinee LLM The input of examinee (denoted as LLM1) is a cluster of concepts. The designed prompt is *"Please generate a primitive for the following concepts: <C>."*, where *C* is a list of concepts in a cluster.

Step-2: Primitive Verification by Examiner LLM The examiner (denoted as LLM2) is to verify whether the primitive generated by LLM1 is correct or not. To setup LLM2, we input the primitive <P> and the related concepts <C> into it, concatenated to the following instructions: *Do you think <P> is representative for the following concepts: <C>. Please answer "yes" or "no".*.

Step-3: Explainable context by Examiner LLM For the correct primitive and cluster, we ask the LLM2 to generate a sentence as explainable context. With the primitive <P> and the related concepts <C> into it, concatenated to the following instructions: *Please generate a short sentence to describe the primitive <P>. The sentence is assiated with a [MASK], where can be replaced by the concepts in <C>.*.

7. Experiments

We start with the human assessment and verify PrimeNet is a high-quality knowledge base. Then, we test PrimeNet on enriching distributional representations and commonsense reasoning tasks.

7.1. Task-1: Human Assessment

We first evaluate the accuracy of the knowledge presented in PrimeNet. We adopt the evaluation method and criteria established by Hwang et al. Hwang et al. [7] and randomly select 3,000 triplets from PrimeNet and present each triplet in the format of (head_concept, relation, tail_concept), with the description of relations provided as a guide. The evaluation involves three annotators who hold Ph.D. degrees in computer science. The annotators use four labels to assess each triplet: 1) *always/often*, indicating the triplet is frequently true; 2) *sometimes/likely*, indicating it is occasionally or probably true; 3) *farfetched/never*, indicating it is false or extremely unlikely; and 4) *invalid*, indicating it is illogical. Triplets labeled as *always/often* or *sometimes/likely* are categorized as *Accept*, while others are categorized as *Reject*. To ensure impartial evaluation, annotators are allowed to skip unfamiliar triplets by labeling *No Judgment*. The final results are determined by the majority vote among three annotators.

This experiment assesses PrimeNet's quality and compares it to other commonsense knowledge bases, including:

- **TransOMCS** [5]: This is a knowledge base containing 18.5M triplets that were automatically extracted from syntactic parses of sentences from various web sources, including Wikipedia, Yelp, and Reddit.
- **ATOMIC** [6]: It contains 877K textual descriptions of inferential knowledge. It is organized as typed if-then relations with variables, such as "if X pays Y a compliment, then Y will likely return the compliment".
- **ConceptNet** [4]: This is a large-scale knowledge base that contains relational knowledge collected from resources created by experts, crowd-sourcing, and games with a purpose [40].

Table 6 shows the human assessments of different knowledge bases[8]. It is observed that PrimeNet stands out as the highest quality knowledge base with an acceptance rate of 92.4%, showing that PrimeNet is highly reliable and contains

[8]Performances of compared knowledge bases are reported by [7], which are evaluated through crowd-sourcing on the Amazon Mechanical Turk platform.

commonsense knowledge that is consistent with human understanding. ConceptNet, ATOMIC$^{20}_{20}$, and ATOMIC also demonstrate high quality, with acceptance rates of 88.6%, 91.3%, and 88.5%, respectively. Although TransOMCS has a vast number of triplets (i.e., 18.5M), it has a lower accuracy compared to the other resources, with an acceptance rate of only 41.7%, indicating it may not be as reliable as the other knowledge bases.

7.2. Task-2: Semantic Similarity

We evaluate the effectiveness of PrimeNet by examining its impact on improving distributional representations on the word semantic similarity task. Following previous works [41, 4, 42, 7], knowledge bases are used as external knowledge to adjust pre-trained word embeddings. The resulting refined embeddings, molded by insights from various knowledge bases, undergo systematic evaluation in downstream tasks, such as word semantic similarity assessments. Enhanced performance serves as an indicator of the superior quality of knowledge bases in improving distributional representations.

We employ a retrofitting method[9] designed by Faruqui et al. Faruqui et al. [41] to improve pre-trained word embeddings with different knowledge bases. It is designed to make words that are known to be related in a given knowledge base have similar representations in embedding space. The training objective is to make the new embedding of a word to be both similar to its initial embedding and nearby words in the knowledge base, by minimizing the following objective function:

$$L = \sum_{i=1}^{n} (\alpha_i ||\mathbf{w}_i - \mathbf{w}_i^*||^2 + \sum_{(w_i,w_j)\in\mathcal{R}} \beta_{i,j} ||\mathbf{w}_i - \mathbf{w}_j||^2), \tag{8}$$

where α and β control the relative strengths of associations, $\mathbf{w}_i^*$ is the original embedding of word w_i, and $\mathbf{w}_i$ is its new embedding, $\mathcal{R}$ denotes a set of relations extracted from the knowledge base, and (w_i, w_j) denotes a relation which connects w_i and w_j. We test the retrofitted embeddings with different knowledge bases on two tasks, i.e., semantic similarity and SAT-style analogy.

This task is to measure the degree of similarity between word pairs by calculating the cosine similarities between their embeddings, and then compare the similarities to human judgments. A good method should provide similarities that are strongly correlated with the human judgments evaluated by Spearman correlation coefficient [43]. We conduct experiments on eight word similarity datasets, including YP-130 [44], MenTR-3K [45], RG-65 [46], MTurk-771 [47], SimLex-999 [48], SimVerb-3500 [49], VERB-143 [50], and WS-353 [51].

Two popular pre-trained word embeddings are used in our experiments, including Word2Vec [52], which is trained on the first 100M of plain text from Wikipedia[10], and GloVe [53], which are trained on 6 billion words from Wikipedia and English Gigaword[11]. In this task, we compare PrimeNet with FrameNet, WordNet, and ConceptNet, which contain synonyms knowledge.

Table 7 presents the overall performance on different word similarity datasets. PrimeNet demonstrated a significant improvement in retrofitting semantic representations, with an average increase of 6.73%, 5.49%, and 5.31% for Word2Vec (300d), GloVe (50d), and GloVe (300d), respectively. WordNet also achieved notable performance gains, with an average improvement of 4.75%, 3.79%, and 3.98%, benefiting the high-quality synonyms knowledge constructed by experts. While the crowd-sourced ConceptNet only slightly outperformed Word2Vec (300d) and GloVe (50d), and slightly worse than GloVe (300d). The solid performance gain achieved by PrimeNet suggests that it is successful in integrating knowledge from various sources into PrimeNet and creating a robust knowledge base.

7.3. Task-3: Neuro-symbolic Commonsense Reasoning

Commonsense knowledge is important to natural language understanding through contextual reasoning. An effective method for assessing this understanding is through commonsense question-answering (QA) tasks, wherein the ability to answer questions often hinges on possessing commonsense knowledge. In commonsense QA tasks, pre-trained language models like BERT and RoBERTa have demonstrated their effectiveness in bridging the gap between human and machine performance. Additionally, the incorporation of external knowledge bases has proven crucial

[9] https://github.com/mfaruqui/retrofitting

[10] We use the Text8Corpus which is available in Gensim: https://github.com/RaRe-Technologies/gensim-data, and the CBOW model for training: https://code.google.com/archive/p/word2vec/

[11] https://nlp.stanford.edu/projects/glove/

Table 7
Overall performance on semantic similarity. d denotes the dimension of embeddings. The best performance is marked in bold.

Methods	YP-130	MenTR-3K	RG-65	MTurk-771	SimLex-999	SimVerb-3500	VERB-143	WS-353	Average (Δ)
Word2Vec (300d)	0.215	0.600	0.633	0.554	0.287	0.155	0.358	0.705	0.438
+FrameNet	0.334	0.589	0.620	0.571	0.295	0.227	0.321	0.651	0.451 (+1.25%)
+WordNet	0.316	0.620	**0.717**	0.598	0.377	0.237	0.318	0.705	0.486 (+4.75%)
+ConceptNet	**0.386**	0.582	0.577	0.533	0.341	0.229	0.302	0.651	0.450 (+1.16%)
+PrimeNet	0.325	**0.638**	0.680	**0.617**	**0.416**	**0.271**	**0.385**	**0.715**	**0.506 (+6.73%)**
GloVe (50d)	0.377	0.652	0.602	0.554	0.265	0.153	0.250	0.499	0.419
+FrameNet	**0.459**	0.622	0.617	0.568	0.288	0.217	0.240	0.471	0.435 (+1.61%)
+WordNet	0.510	0.649	0.688	0.540	0.342	0.239	0.188	0.500	0.457 (+3.79%)
+ConceptNet	0.427	0.599	0.558	0.493	0.356	0.234	0.236	0.489	0.424 (+0.50%)
+PrimeNet	0.443	**0.674**	**0.707**	**0.597**	**0.376**	**0.236**	**0.273**	0.485	**0.474 (+5.49%)**
GloVe (300d)	0.561	0.737	0.766	0.650	0.371	0.227	0.305	0.605	0.528
+FrameNet	0.589	0.701	0.756	0.639	0.361	0.278	0.274	0.558	0.519 (-0.84%)
+WordNet	**0.610**	0.759	**0.841**	0.679	0.470	0.313	0.256	0.612	0.568 (+3.98%)
+ConceptNet	0.561	0.700	0.747	0.583	0.420	0.288	0.300	0.595	0.524 (-0.34%)
+PrimeNet	0.593	**0.764**	0.818	**0.684**	**0.496**	**0.316**	**0.350**	**0.626**	**0.581 (+5.31%)**

for enhancing answer accuracy, providing valuable insights for contextual comprehension and reasoning. Hence, approaches that combine neural pre-trained language models with symbolic knowledge bases, known as *neuro-symbolic* methods, have exhibited significant potential for advancing commonsense reasoning.

Task Setting Following previous methods [8, 54], we use a neuro-symbolic method to evaluate the commonsense QA under a zero-shot setting proposed by Ma et al. [55]. Formally, given a natural language question q and a set of possible answers $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$, the task is to select the most probable answer a^* from $\mathcal{A}$. The wrong answers in $\mathcal{A}$ are denoted as distractors. The pre-trained language models are used as backbone. RoBERTa-large is used in our experiments. In a zero-shot setting, the model has no access to the training data. The neuro-symbolic solution is to transform knowledge from different knowledge bases into an artificial QA set for pre-training. For example, a triplet *(losing weight, usedFor, being healthier)* is generated as *losing weight is for being healthier*, and several distractors are generated by negative sampling. After pre-training, the model are tested on different datasets. We follow the parameter settings in Ma et al. [55]. The experiments are tested for five rounds, and the average accuracy of the predicted answers is used as the metric.

Baseline We compare the neuro-symbolic methods with the following baselines. *Majority* answers each question with the most frequent option in the entire dataset. Self-Talk [56] is an unsupervised method. It generates clarification prompts based on a template prefix, which are leveraged to elicit knowledge from another language model, which is used jointly with the original context and question to score each answer candidate. SMLM [57] is designed to pre-train the LM with three representation learning functions which aim to complete a knowledge triple given two of its elements. To show the upper bound, we report the supervised methods on RoBERTa-large model with access to the training data, as well as the human performance. of this work, we include results of a supervised fine-tuned RoBERTa system and of human evaluation. To facilitate the neuro-symbolic method for commonsense reasoning, we compare PrimeNet with ATOMIC, ConceptNet, Wikidata, WordNet, and CSKG. Please refer to [58] for more details about QA data generation with different knowledge bases, distractors sampling, and training regimes.

Benchmarks Following Ma et al. [55], we use five commonsense QA benchmarks for evaluation, including:

- Abductive Natural Language Inference (aNLI) [59] is a binary-classification task, which is to apply abductive reasoning and commonsense to form possible explanations for a given set of observations. Given two observations from narrative contexts, the goal is to pick the most plausible explanatory hypothesis.
- Commonsense Question Answering (CSQA) [60] contains 12,247 examples. Each example includes a question and five answer candidates. The questions are sourced from a ConceptNet. Answer candidates are formed by combining ConceptNet nodes with additional distractors gathered through crowdsourcing.

Table 8
Performance of neuro-symbolic methods across five commonsense QA tasks in a zero-shot setting. RoBERTa-large* denotes the performance of RoBERTa-large under a supervised setting.

Model	Knowledge Base	aNLI	CommonsenseQA	PIQA	SocialIQA	WinoGrande
Majority	-	50.8	20.9	50.5	33.6	50.4
Self-talk	-	-	32.4	70.2	46.2	54.7
SMLM	-	65.3	38.8	-	48.5	-
RoBERTa-large*	-	85.6	78.5	79.2	76.6	79.3
Human Performance	-	91.4	88.9	94.9	86.9	94.1
RoBERTa-large	-	65.5	45.0	67.6	47.3	57.5
	ATOMIC	70.8	64.2	72.1	63.1	59.6
	ConceptNet, Wikidata, WordNet	70.0	67.9	72.0	54.8	59.4
	CSKG	70.5	67.4	72.4	63.2	60.9
	PrimeNet	**71.2**	**68.3**	**72.4**	**64.5**	**62.1**

- Physical Interaction Question Answering (PIQA) [61] is a dataset for reasoning about physical commonsense. Each question is associated with two possible solutions. The task is to choose the most appropriate solution, of which exactly one is correct.
- Social Intelligence Question Answering (SIQA) [62] is a dataset for commonsense reasoning about social situations, with 38,000 multiple choice questions. Each example comprises a context, a question, and three answer candidates. The context is derived from ATOMIC, questions are generated based on nine templates corresponding to relations in ATOMIC, and answers are obtained through crowdsourcing.
- WinoGrande (WG) [63] contains 44K problems inspired by pronoun resolution problems in Winograd Schema Challenge (WSG) [58]. Each example includes a context description featuring an emphasized pronoun, with two options provided as possible references.

Performance It is observed that pre-training the language model with external knowledge is effectiveness to improve the performance of commonsense QA task. The main reason is that the external knowledge is important supplementary information for implicit knowledge embedding in pre-trained language models. Our PrimeNet achieved the best performance when RoBERTa is used as backbone, showing that PrimeNet has a good quality in organizing commonsense knowledge.

7.4. Concept Detection

We perform a probing experiment as illustrated in Fig. 4. We assume that words from Core WordNet are concepts, given their fundamental role in describing the world. For all nodes in Core WordNet and our knowledge graph $\mathcal{G}$ of PrimeNet, we show probability distributions of their level scores and entropy scores. It is observed that a considerable number of words in Core WordNet have level scores below 50, and entropy scores under 1. These words are readily excluded from concept sets, by applying previous methods for conceptualization.

7.5. Case Study

In our method, we manually checked the detected primitives. This step is conduct by 5 senior Ph.D. students majors in natural language processing. We manually code the explainable of primitives. For example, `INCREASE` is defined as `INCREASE(obj) := obj++`, which is the basic operation that increments the value of an object and provides a foundation for more complex reasoning. It is observed that some primitives have a hierarchical structure. We show examples of primitives in Fig. 7. At *Level-1*, the primitive `GROW` is defined as `GROW(obj) = INCREASE(obj.SIZE) := obj.SIZE++ = obj(l++, h++, w++)`, which is accomplished by using the INCREASE primitive to increment the object's `SIZE` attribute, such as length (l), height (h), and width (w). The *Level-2* primitive `LENGTHEN` is even more specific, adding only length to an object, and it is defined as `LENGTHEN(obj)=INCREASE(obj.SIZE.LENGTH) :=obj.SIZE.LENGTH++ = obj(l++, h, w)`.

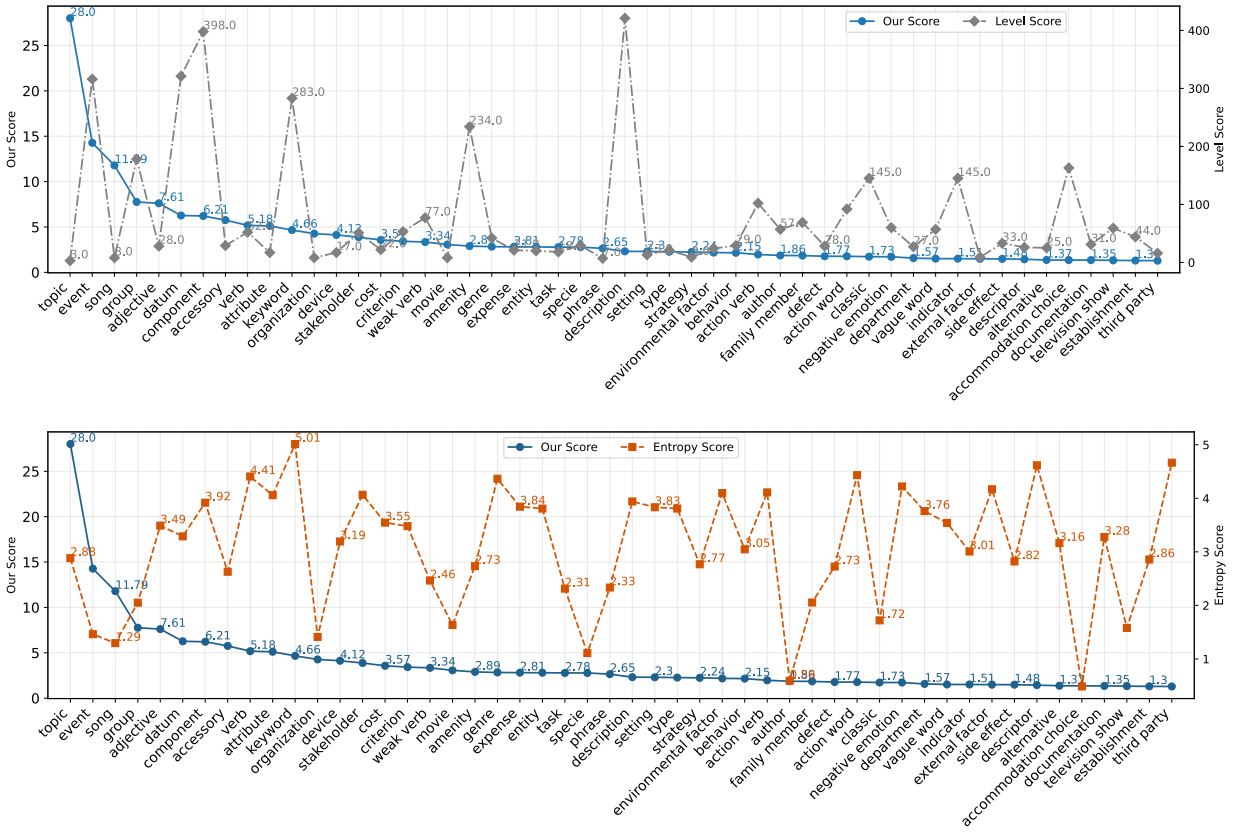


Figure 6: Examples of top-50 words scored by the designed conceptual score function. We compare their level scores and entropy scores with our conceptual scores.

8. Related Works

In this section, we conduct a comprehensive literature review on commonsense knowledge acquisition, including crowdsourcing methods, automatic extraction methods, and approaches centered around extracting implicit knowledge from pre-trained language models. Then, we introduce the conceptual primitives theory, which is a pivotal component in the construction of our commonsense knowledge base.

8.1. Commonsense Knowledge Acquisition

Commonsense knowledge is not explicitly defined. It is an inherent understanding of the world that humans possess but machines lack. To narrow the gap between human and machine intelligence, the process of acquiring commonsense knowledge is crucial for improving machine intelligence. There are mainly three major methods to the knowledge acquisition, i.e., crowdsourcing, automatic extraction, and mining from pre-trained language models.

8.1.1. Crowdsourcing

Crowdsourcing is a useful approach for collecting commonsense knowledge from a diverse group of human contributors, such as human experts [33, 64], web users [65, 66], and participants in human computation games [67, 68]. By tapping into the collective wisdom of individuals, this approach captures intuitions and insights commonly held by people, thus contributing valuable data to the construction of commonsense knowledge bases. The crowdsourcing approach exhibits high adaptability across diverse tasks and domains. By involving a varied group of contributors, it ensures that multiple viewpoints are considered, leading to the creation of a more comprehensive and balanced knowledge pool. The existing knowledge bases built through crowdsourcing typically encompass the following categories of commonsense knowledge.

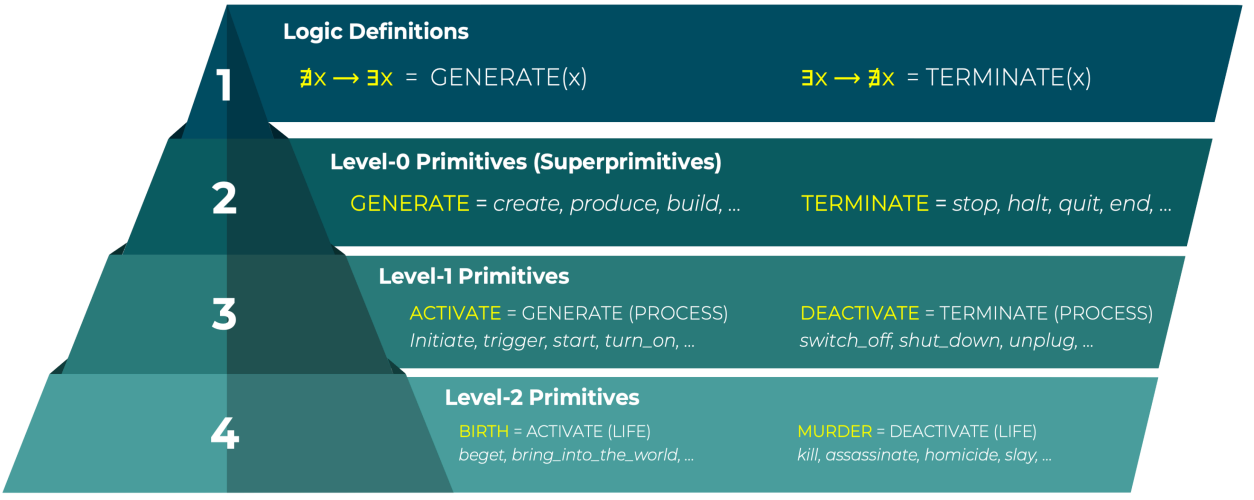


Figure 7: Examples of the hierarchical structure of primitives in PrimeNet.

Factual Knowledge It represents concrete and specific details about the world, events, people, places, objects, and other observable phenomena, such as "wheel is part of bicycle", "dog is an animal", and "Los Angeles is located in California". In the early 1980s, the Cyc [64] project undertook the task of manually constructing a comprehensive knowledge base using the CycL representation language, encompassing the basic facts and rules about the world. After the efforts of its first decade, the Cyc project expanded to include around 100,000 terms. By the time of its release in 2012, known as OpenCyc 4.0, the knowledge base had undergone substantial growth, encompassing over 2 million facts across 239,000 concepts. In 2002, the DOLCE [69] (Descriptive Ontology for Linguistic and Cognitive Engineering) project was designed to manually collect the ontological categories underlying natural language and human commonsense with disambiguated concepts and relations. Freebase [70] is a collaborative knowledge base by gathering data from various sources, including Wikipedia, the Notable Names Database, and contributions from community users. Google Knowledge Graph [71] is powered in part by Freebase, with an extensive collection of billions of facts about people, places, and things. It is served as a foundation for Google's search results, enabling the search engine to deliver useful and accurate information to users. ConceptNet [4] leverages crowd-sourcing contributions from users to acquire commonsense knowledge. It originated from the Open Mind Common Sense [65] and has grown by incorporating data from other crowd-sourced resources, expert-created content, and purposeful games. ConceptNet is a widely used commonsense knowledge base with over 21 million edges and 8 million nodes, covering a diverse range of 36 commonsense relations, such as isA, partOf, usedFor, and capableOf. Moreover, ConceptNet can be linked to other knowledge bases, such as WordNet, Wiktionary, OpenCyc, and DBpedia, and now, it is a multi-lingual knowledge base that can also build connections among 83 languages.

Lexical Knowledge There are several lexical databases manually created by experts, such as WordNet [33], Roget's Thesaurus [34], FrameNet [3], MetaNet [72], VerbNet [73], and PropBank [74]. Among these lexical knowledge bases, WordNet is a highly popular lexical knowledge base which captures semantic relations between words. Within WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. These synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet is now available in over 200 languages, allowing researchers and linguists worldwide to explore the complexities of language and word associations across diverse contexts.

Encyclopedic Knowledge Encyclopedic knowledge is related to a broad understanding of various subjects and topics. For example, Wikidata [75] is a knowledge graph coupled with Wikipedia, which is a free, open, and multilingual online encyclopedia that is collaboratively edited by volunteers. DBpedia [76] extracts structured information from Wikipedia data and converts it into a machine-readable format for use in the Semantic Web and data mining domains. The encyclopedic knowledge resources offer a wide range of information to help people understand various topics and fields.

Domain Knowledge More recently, commonsense knowledge bases have been specifically developed to cater to particular tasks. For example, SenticNet [14] is a sentiment knowledge base which captures the affective commonsense and emotions expressed in natural language. Visual Genome [77] contains annotations of concepts and their relations found in a collection of images. The image descriptions are manually written by crowd workers, and the concepts are automatically mapped to WordNet senses and further refined by crowd workers. ATOMIC [6] is developed to capture inferential commonsense knowledge, such as cause-and-effect relationships. It is developed by domain experts who contribute and validate information about everyday events and their implied causality. ATOMIC$^{20}_{20}$ [7] is proposed to unify the triples from ConceptNet and ATOMIC, together with some newly developed relations.

8.1.2. Automatic Extraction

Despite commonsense knowledge is not explicitly defined, it has been observed that certain types of commonsense knowledge can be extracted through automatic methods, such as text mining and information extraction. Compared with crowdsourcing, these automatic extraction methods can handle large volumes of data efficiently and at a lower cost, making them valuable tools for efficiently capturing and updating commonsense knowledge from various domains.

First, automatic extraction methods generally acquire commonsense knowledge from large-scale text and web pages. For example, NELL [78] (Never-Ending Language Learning system) is designed to automatically extract structured information from unstructured Web pages. With hundreds of pre-defined categories and relations and 10 to 15 examples of each, NELL extracts knowledge from more than 500 million web pages, resulting in a large knowledge base comprising over 2.8 million instances. WebChild [79] is constructed through automated extraction and disambiguation from Web contents. It utilizes seeds derived from WordNet and pattern matching techniques on large-scale text collections to gather information, including fine-grained relations like "hasShape," "hasTaste," and "evokesEmotion". ASER [15] (activities, states, events, and their relations) is a large-scale eventuality knowledge graph extracted from more than 11-billion-token unstructured textual data. SenticNet [14] is constructed using auto-regressive language models and kernel methods to extract polarity from text in a completely interpretable and explainable manner. Probase [80] is constructed by extracting and organizing knowledge from a vast collection of Web pages and documents. Its subsequent version, named as Microsoft Concept Graph [81], harnesses billions of web pages and search logs to build a huge graph of relations between concepts, and has been proven valuable in enhancing search engines, spell-checkers, recommendation engines, and other AI-driven systems.

Second, several methods are used to improve the existing commonsense knowledge bases. The automatic extraction methods can help fill gaps, update outdated information, and supplement missing commonsense knowledge in existing knowledge bases. For example, BabelNet [82] is a multilingual knowledge base which is automatically created by mapping the multilingual encyclopedic knowledge repository (Wikipedia) to the English WordNet based on multilingual concept lexicalizations and machine translations. Dense-ATOMIC [83] is designed to overcome the limitations of ATOMIC in knowledge coverage and multi-hop reasoning, by employing a knowledge graph completion approach to train a relation prediction model and infer missing links within ATOMIC, ensuring high knowledge coverage and facilitating massive multi-hop paths.

Third, some efforts have been made to automatic integrate diverse commonsense knowledge bases, enhancing the overall coverage and richness of the knowledge base. For example, YAGO [84] (Yet Another Great Ontology) is designed to extract commonsense knowledge from Wikipedia, WordNet, WikiData, GeoNames, and other data sources. Bouraoui et al. [85] employed Region Connection Calculus to merge open-domain terminological knowledge. CommonSense Knowledge Graph (CSKG) [86] integrates knowledge bases from seven diverse, disjoint sources such as ConceptNet and WordNet. Based on ASER, Zhang et al. [5] have developed TransOMCS with an algorithm for discovering patterns from the overlap of existing commonsense and linguistic knowledge bases, and a commonsense knowledge ranking model to select the highest-quality extracted knowledge.

8.1.3. Implicit Knowledge in Pre-trained Models

Recent advancements in pre-trained models have demonstrated significant improvements across various tasks, underscoring their robust representation and generalization capabilities. These models, pre-trained on large-scale corpora, have proven adept at encoding diverse forms of knowledge [87, 88]. For example, BERT (Bidirectional Encoder Representations from Transformers) uses a masked language model objective in pre-training, where parts of the input are masked, enabling the model to predict concealed words bidirectionally. This process empowers BERT to capture contextualized representations, comprehensively understanding intricate relationships and meanings in different linguistic contexts. Similarly, GPT [89, 90, 91] (Generative Pre-trained Transformer) follows the generative language

model paradigm, predicting the next word based on preceding context. With a unidirectional architecture processing text from left to right during training, it acquires knowledge of grammar, facts, reasoning, and even some degree of commonsense.

Currently, there is a trend to mine commonsense knowledge directly from pre-trained language models, leveraging the rich information embedded in these large models. Several works are designed to probe commonsense knowledge directly from large pre-trained models, such as KB-BERT [92], KB-BERTSAGE [93], and PseudoReasoner [94]. These approaches involve fine-tuning pre-trained language models, such as BERT and BART, on commonsense knowledge bases like ATOMIC, ConceptNet, and ASER, with the tasks typically entails providing the head and relation in a commonsense triple as input, with the tail serving as the expected output. COMET [95] (COMmonsEnse Transformers) is designed to leverage GPT to generate rich and diverse commonsense descriptions in natural language. It effectively transforms implicit knowledge from pre-trained models into explicit knowledge within commonsense knowledge graphs, and generates novel knowledge that humans rate as high quality. LAMA [96] (LAnguage Model Analysis) is an unsupervised method to leverage BERT to acquire commonsense knowledge. It also serves as a framework[12] for probing and evaluating the factual knowledge encoded in pre-trained language models [97]. West et al. [98] design a symbolic knowledge distillation to leverage some seeds from ATOMIC as prompts to acquire commonsense knowledge from GPT-3, resulting a large commonsense knowledge graph ATOMIC^{10x} and a compact commonsense model COMET$^{DIS}_{TIL}$. Their work demonstrates the efficacy of collaborative efforts between humans and language models for curating commonsense knowledge graphs and training efficient, high-performing commonsense models.

8.2. Conceptual Primitives

Conceptual primitives can be defined as concepts that cannot be defined in terms of other concepts in an integration data model which provides an overview of data, thereby forming foundations for definitions of other concepts [99]. Conceptual primitives have been of practical and theoretical interest to researchers in computer science [24], linguistics [20, 21] and psychology [100]. Such research reports that the decomposition of meanings into lower-level parts is essential for conceptualization.

We apply the idea of conceptual primitives to construct commonsense knowledge by comprising a small core of primitive commonsense concepts and relations, linked to a much more extensive base of factual knowledge instances. Naturally, humans tend to categorize things, events, and people by identifying common patterns and forms, which is the basis of the conceptual primitive theory. Thus, commonsense knowledge bases built upon conceptual primitives possess the greater potential to facilitate reasoning tasks. Recently, Cambria et al. [14] constructed SenticNet by generalizing words and multi-word expressions into primitives and super-primitives annotated with emotion labels via pre-trained language models, which achieved better performances on various affective tasks and showed the power of conceptual primitives. Unlike SenticNet, which focuses on sentiment knowledge, we build PrimeNet to cover a broader range of general commonsense knowledge based on conceptual primitives.

9. Conclusion

We present a new commonsense knowledge base based on the conceptual primitive theory, named PrimeNet. Different from existing knowledge bases, PrimeNet is constructed based on a small core of primitive commonsense and relations, linked to extensive concepts and entities, which is suited for supporting commonsense reasoning. Our studies demonstrate that PrimeNet contains high-quality commonsense knowledge and conceptual primitives. The developed functions also enable the application and extension of PrimeNet for various reasoning tasks. The current API of PrimeNet is `https://sentic.net/api/primenet/`, and PrimeNet is available on `https://github.com/senticnet/primenet`. In the future, we are going to exploit additional intelligent methods for mining commonsense knowledge and we will deploy a more convenient framework for delivering commonsense knowledge services.

References

[1] Erik Cambria, Amir Hussain, Catherine Havasi, and Chris Eckl. Common sense computing: From the society of mind to digital intuition and beyond. In *Biometric ID Management and Multimodal Communication*, volume 5707 of *Lecture Notes in Computer Science*, pages 252–259. 2009.

[2] Douglas Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38, 12 1998.

[12] `https://github.com/facebookresearch/LAMA`

[3] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 86–90, 1998.

[4] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pages 4444–4451, 2017.

[5] Hongming Zhang, Daniel Khashabi, Yangqiu Song, and Dan Roth. Transomcs: From linguistic graphs to commonsense knowledge. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, pages 4004–4010, 2020.

[6] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.

[7] Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[8] Filip Ilievski, Pedro A. Szekely, and Bin Zhang. CSKG: the commonsense knowledge graph. In *Proceedings of The Semantic Web - 18th International Conference, ESWC*, volume 12731 of *Lecture Notes in Computer Science*, pages 680–696, 2021.

[9] Jingping Liu, Tao Chen, Chao Wang, Jiaqing Liang, Lihan Chen, Yanghua Xiao, Yunwen Chen, and Ke Jin. Vocsk: Verb-oriented commonsense knowledge mining with taxonomy-guided induction. *Artificial Intelligence*, 310:103744, 2022.

[10] Eugene B. Zechmeister, Andrea M. Chronis, William L. Cull, Catherine A. D'Anna, and Noreen A. Healy. Growth of a functionally important lexicon. *Journal of Reading Behavior*, 27(2):201–212, 1995.

[11] Mengshi Ge, Rui Mao, and Erik Cambria. Explainable metaphor identification inspired by conceptual metaphor theory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10681–10689, 2022. doi: 10.1609/aaai.v36i10.21313. URL https://ojs.aaai.org/index.php/AAAI/article/view/21313.

[12] Rui Mao, Xiao Li, Kai He, Mengshi Ge, and Erik Cambria. MetaPro Online: A computational metaphor processing online system. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 127–135, Toronto, Canada, 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.acl-demo.12.

[13] Rui Mao, Kelvin Du, Yu Ma, Luyao Zhu, and Erik Cambria. Discovering the cognition behind language: Financial metaphor analysis with MetaPro. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2023.

[14] Erik Cambria, Xulang Zhang, Rui Mao, Melvin Chen, and Kenneth Kwok. SenticNet 8: Fusing emotion AI and commonsense AI for interpretable, trustworthy, and explainable affective computing. In *International Conference on Human-Computer Interaction (HCII)*, 2024.

[15] Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. ASER: A large-scale eventuality knowledge graph. In *Proceedings of The Web Conference 2020, WWW*, pages 201–211, 2020.

[16] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD*, pages 481–492, 2012.

[17] Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao. An inference approach to basic level of categorization. In *Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM*, pages 653–662, 2015.

[18] Noam Chomsky. *Syntactic Structures*. De Gruyter Mouton, Berlin, Boston, 1957. ISBN 9783112316009. doi: doi:10.1515/9783112316009.

[19] Roger C. Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):pages 532–631, 1972.

[20] Ray Jackendoff. Toward an explanatory semantic representation. *Linguistic inquiry*, 7(1):89–150, 1976.

[21] Anna Wierzbicka. *Semantics: Primes and universals: Primes and universals*. Oxford University Press, UK, 1996.

[22] Ray S Jackendoff et al. Semantics and cognition. 1983.

[23] S Pesina and T Solonchak. Semantic primitives and conceptual focus. *Procedia-Social and Behavioral Sciences*, 192:339–345, 2015.

[24] Marvin Minsky. A framework for representing knowledge, 1974.

[25] Jean Piaget, Margaret Cook, et al. *The origins of intelligence in children*, volume 8. International Universities Press New York, 1952.

[26] D.E. Rumelhart and A. Ortony. *The Representation of Knowledge in Memory*. Technical report (University of California, San Diego. Center for Human Information Processing). Center for Human Information Processing, Department of Psychology, University of California, San Diego, 1976.

[27] Terry Winograd. Towards a procedural understanding of semantics. *Revue internationale de philosophie*, pages 260–303, 1976.

[28] Daniel G Bobrow and Donald A Norman. Some principles of memory schemata. In *Representation and understanding*, pages 131–149. Elsevier, 1975.

[29] Mark Johnson. The body in the mind: The bodily basis of meaning, imagination, and reason. *Journal of Aesthetics and Art Criticism*, 47(4), 1989.

[30] Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[31] Matthew West. *Developing High Quality Data Models*. Morgan Kaufmann Publishers Inc., 2011.

[32] Lennart Wachowiak and Dagmar Gromann. Systematic analysis of image schemas in natural language through explainable multilingual neural language processing. In *Proceedings of the International Conference on Computational Linguistics, COLING*, pages 5571–5581, 2022.

[33] George A. Miller. Wordnet: A lexical database for english. *Communications of The ACM*, 38:39–41, 1995.

[34] Barbara Ann Kipfer. *Roget's 21st century thesaurus in dictionary form (Third Edition)*. 2005.

[35] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*, volume 4825, pages 722–735, 2007.

[36] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[37] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, pages 3980–3990, 2019.

[38] Roger C Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive psychology*, 3(4):552–631, 1972.

[39] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *International journal of human-computer studies*, 43 (5-6):625–640, 1995.

[40] Luis Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[41] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, 2015.

[42] Qian Liu, Heyan Huang, Guangquan Zhang, Yang Gao, Junyu Xuan, and Jie Lu. Semantic structure-based word embedding by incorporating concept convergence and word divergence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5261–5268, 2018.

[43] Jerome L Myers and Arnold D Well. *Research Design & Statistical Analysis*. Routledge, 1995.

[44] Dongqiang Yang and David Martin Powers. *Measuring semantic similarity in the taxonomy of WordNet*. Australian Computer Society, 2005.

[45] Distributional semantics in technicolor. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL*, pages 136–145, 2012.

[46] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[47] Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. Large-scale learning of word relatedness with constraints. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, pages 1406–1414, 2012.

[48] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[49] Daniela Gerz, Ivan Vulic, Felix Hill, Roi Reichart, and Anna Korhonen. Simverb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 2173–2182, 2016.

[50] Simon Baker, Roi Reichart, and Anna Korhonen. An unsupervised model for instance level subcategorization acquisition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 278–289, 2014.

[51] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: the concept revisited. In *Proceedings of the International World Wide Web Conference, WWW*, pages 406–414, 2001.

[52] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems, NIPS*, pages 3111–3119, 2013.

[53] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1532–1543, 2014.

[54] Filip Ilievski, Alessandro Oltramari, Kaixin Ma, Bin Zhang, Deborah L. McGuinness, and Pedro A. Szekely. Dimensions of commonsense knowledge. *Knowledge-Based Systems*, 229:107347, 2021.

[55] Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Proceedings of Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 13507–13515, 2021.

[56] Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 4615–4629, 2020.

[57] Pratyay Banerjee and Chitta Baral. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 151–162, 2020.

[58] Hector J. Levesque. The winograd schema challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06*. AAAI, 2011.

[59] Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. In *Proceedings of International Conference on Learning Representations, ICLR*. OpenReview.net, 2020.

[60] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4149–4158, 2019.

[61] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pages 7432–7439, 2020.

[62] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 4462–4472, 2019.

[63] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pages 8732–8740, 2020.

[64] Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of The ACM*, 38(11):32–38, 1995.

[65] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems*, volume 2519 of *Lecture Notes in Computer Science*, pages 1223–1237. Springer, 2002.

[66] Timothy Chklovski. Learner: a system for acquiring commonsense knowledge by analogy. In John H. Gennari, Bruce W. Porter, and Yolanda Gil, editors, *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003)*, pages 4–12, 2003.

[67] Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: a game for collecting common-sense facts. In *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI*, pages 75–78, 2006.

[68] Yen-Ling Kuo, Jong-Chuan Lee, Kai-yang Chiang, Rex Wang, Edward Shen, Cheng-wei Chan, and Jane Yung-jen Hsu. Community-based

game design: experiments on social games for commonsense data collection. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 15–22, 2009.

[69] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with DOLCE. In *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW*, volume 2473 of *Lecture Notes in Computer Science*, pages 166–181, 2002.

[70] Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD*, pages 1247–1250, 2008.

[71] A Singhal. Official google blog: Introducing the knowledge graph: things, not strings. 2012.

[72] Ellen Dodge, Jisup Hong, and Elise Stickles. MetaNet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 40–49, 2015.

[73] Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.

[74] Martha Palmer, Paul R. Kingsbury, and Daniel Gildea. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[75] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[76] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[77] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[78] Tom Mitchell and E. Fredkin. Never-ending language learning. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 1–1, 2014.

[79] Niket Tandon, Gerard de Melo, Fabian M. Suchanek, and Gerhard Weikum. Webchild: harvesting and organizing commonsense knowledge from the web. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM*, pages 523–532, 2014.

[80] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of ACM SIGMOD International Conference on Management of Data, SIGMOD*, pages 481–492, 2012.

[81] Lei Ji, Yujing Wang, Botian Shi, Dawei Zhang, Zhongyuan Wang, and Jun Yan. Microsoft concept graph: Mining semantic concepts for short text understanding. *Data Intelligence*, 1(3):238–270, 2019.

[82] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[83] Xiangqing Shen, Siwei Wu, and Rui Xia. Dense-atomic: Towards densely-connected ATOMIC with high knowledge coverage and massive multi-hop paths. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL*, pages 13292–13305, 2023.

[84] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *Proceedings of the International Conference on the World Wide Web*, pages 697–706, 2007.

[85] Zied Bouraoui, Sébastien Konieczny, Thanh Ma, Nicolas Schwind, and Ivan Varzinczak. Region-based merging of open-domain terminological knowledge. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, KR*, pages 81–90, 2022.

[86] Filip Ilievski, Pedro A. Szekely, and Bin Zhang. CSKG: the commonsense knowledge graph. In *Proceedings of The Semantic Web - 18th International Conference, ESWC*, volume 12731, pages 680–696, 2021.

[87] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona T. Diab, and Marjan Ghazvininejad. A review on language models as knowledge bases. *CoRR*, abs/2204.06031, 2022.

[88] Prajjwal Bhargava and Vincent Ng. Commonsense knowledge reasoning and generation with pre-trained language models: A survey. In *Proceedings of Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 12317–12325, 2022.

[89] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[90] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[91] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[92] Liang Yao, Chengsheng Mao, and Yuan Luo. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193, 2019.

[93] Tianqing Fang, Weiqi Wang, Sehyun Choi, Shibo Hao, Hongming Zhang, Yangqiu Song, and Bin He. Benchmarking commonsense knowledge base population with an effective evaluation dataset. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 8949–8964, 2021.

[94] Tianqing Fang, Quyet V. Do, Hongming Zhang, Yangqiu Song, Ginny Y. Wong, and Simon See. Pseudoreasoner: Leveraging pseudo labels for commonsense knowledge base population. In *Findings of the Association for Computational Linguistics, EMNLP*, pages 3379–3394, 2022.

[95] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 4762–4779, 2019.

[96] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 2463–2473, 2019.

[97] Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. How context affects language models' factual predictions. In *Proceedings of Conference on Automated Knowledge Base Construction, AKBC*, 2020.

[98] Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL*, pages 4602–4625, 2022.

[99] Matthew West. *Developing high quality data models*. Elsevier, 2011.

[100] David E Rumelhart and Andrew Ortony. The representation of knowledge in memory. *Schooling and the acquisition of knowledge*, 99:135, 1977.