Understanding Substructures in Commonsense Relations in ConceptNet

Ke Shen · Mayank Kejriwal

Abstract Acquiring commonsense knowledge and reasoning is an important goal in modern NLP research. Despite much progress, there is still a lack of understanding (especially at scale) of the nature of commonsense knowledge itself. A potential source of structured commonsense knowledge that could be used to derive insights is ConceptNet. In particular, ConceptNet contains several coarse-grained relations, including ‘HasContext’, ‘FormOf’ and ‘SymbolOf’, which can prove invaluable in understanding broad, but critically important, commonsense notions such as ‘context’. In this article, we present a methodology based on unsupervised knowledge graph representation learning and clustering to reveal and study substructures in three heavily used commonsense relations in ConceptNet. Our results show that, despite having an ‘official’ definition in ConceptNet, many of these commonsense relations exhibit considerable sub-structure. In the future, therefore, such relations could be sub-divided into other relations with more refined definitions. We also supplement our core study with visualizations and qualitative analyses.

Keywords Commonsense · ConceptNet · Context · Knowledge Graph · Knowledge Graph Embedding

1 Introduction

Despite the ubiquity of intelligent agents such as Alexa and Siri in modern life, these agents have yet to capture the human element in natural conversations. Even with advances in Natural Language Processing (NLP), deep learning, and knowledge graphs [24], [49], it is not clear if such agents are fully capable of answering questions (e.g., ‘Should I put my spare change in a piggy bank?’) with incomplete
information, or under-specified needs. Such questions tend to require more contextual and implicit knowledge that humans often take for granted when navigating daily situations. Among other things, lack of deep contextual understanding limits the agents’ commonsense reasoning abilities.

Commonsense reasoning is the process that involves processing information about a scenario in the world, and making inferences and decisions by using context, implicit and explicit information based on our collective ‘commonsense knowledge’. Commonsense knowledge is difficult to define precisely but it is usually assumed to be a broad body of knowledge of how the ‘world’ works [42]. Generally, such knowledge is essential for navigating social situations and interactions, ‘naive’ physical understanding (e.g., the simple knowledge that when an object on the table is ‘picked up’, it is not on the table anymore) and more controversially, knowledge that relies on reasoning about local culture and milieu [54], [33].

Commonsense knowledge and reasoning have both been recognized as essential for building more advanced ‘general’ AI systems that have human-like capabilities and reasoning ability when facing uncertain, implicit, or even potentially contradictory, information. Recognizing its importance, researchers in several communities have increasingly engaged in improving agent performance on commonsense question answering, abductive reasoning and other tasks pertinent to commonsense reasoning [9], [53], [54], [55].

ConceptNet is a large-scale, freely available knowledge graph (KG) that describes commonsense knowledge as a set of assertions or triples [59]. It is designed to represent the common knowledge needed to help machines better understand the meanings of concepts and inter-concept relationships that people rely on in everyday situations. The graph structure that represents knowledge in ConceptNet is particularly useful for textual reasoning over natural language documents. An example of how such knowledge is organized in ConceptNet is illustrated in Figure 1. ConceptNet originated from the Open Mind Common Sense [19] project, itself launched in 1999 at the MIT Media Lab. It was regularly updated to include new knowledge from crowdsourced resources, expert-curated resources, and ‘games with a purpose’ designed specifically to elicit certain kinds of commonsense annotations from people, such as intuitive word associations.

Our guiding hypothesis in this paper is that, due to its growth and usage over the years, ConceptNet can yield valuable insights about commonsense knowledge itself. The intuition behind our approach is relatively straightforward. First, using representation learning, we ‘embed’ each entity and relation into a dense, continuous real-valued vector space, learned in an unsupervised fashion using a state-of-the-art knowledge graph representation learning algorithm. Next, we isolate the triples or labeled edges, such as (people, CapableOf, socialize at night) in Figure 1 of interest from the raw knowledge base. For example, we isolate the triples with relation ‘hasContext’ if we want to study that relation. Finally, using the learned embeddings from the first step, we derive an embedding for the triple using the notion of translation that is more formally described subsequently. Once such embeddings have been obtained for each triple with the relation under study, we cluster them using an established algorithm such as k-Means to detect potential sub-structures. We also use statistical and quantitative measures to understand the quality and structural properties of these clusters. Finally, visualization and

1 https://conceptnet.io/
sampling-based qualitative analyses are used to provide more insight into the data. Specific contributions are enumerated below:

- We present a novel methodology for understanding structural aspects of commonsense knowledge by studying three relations (‘HasContext’, ‘SymbolOf’, and ‘FormOf’). These relations are both high-volume and coarse-grained, and are known to be significant in commonsense reasoning, explaining their high prevalence in ConceptNet. Our methodology relies on a judicious combination of knowledge graph representation learning, clustering and visualization.

- We conduct a detailed set of experiments by applying the proposed methodology on a large sample of triples that contain the three relations mentioned earlier. Using quantitative metrics for measuring cohesion and separation in the embedding space, our results show that, despite having an ‘official’ definition in ConceptNet, many of these high-volume, coarse-grained relations exhibit considerable sub-structure. In the future, therefore, such relations could be divided into other relations with more refined definitions.

- Using qualitative and sampling-based analyses, we show how a broad notion like ‘context’ is defined in practice in commonsense knowledge bases such as ConceptNet. These analyses potentially allow us to develop a deeper understanding of the definition and role of context in commonsense knowledge.

While our goal is not to present a complete understanding of common sense, we study the three relations noted earlier (‘HasContext’, ‘SymbolOf’, and ‘FormOf’) to understand relations that are believed to be critical to common sense reasoning. Our methodology and empirical study also relies on established methods from the KG representation learning and clustering literature. An explicit goal behind using established tools, besides their expected reliability, is to enable greater replicability for studying other such relations, including in other knowledge bases.

The rest of this work is structured as follows. In Section 2, we describe relevant lines of research related to this work. In Section 3, we present our materials and methods, followed by the experimental results in Section 4. Section 5 discusses the key implications. The article concludes in Section 6.
2 Related Work

Although both Wikipedia and the relatively recent Defense Advanced Research Projects Agency (DARPA) Machine Common Sense (MCS) program define commonsense reasoning as ‘the basic ability to perceive, understand, and judge things that are shared by nearly all people and can be reasonably expected of nearly all people without need for debate’, there is no official, sufficiently broad definition that we can cite outside of the psychology context. Within psychology, [57] has defined commonsense as ‘the system of implications shared by the competent users of a language’. One important commonality that is shared, no matter the definition or field considered, is that commonsense knowledge is (at least to a degree) implicit.

There is very little work on the nature of commonsense knowledge: one rare example is a recent book [15] that attempts to provide a ‘theory’ of commonsense by breaking down commonsense knowledge into various categories, and present formalisms for those categories. Other similar work along those lines include [23], [22], [39]. Unlike those works, we take an inductive, data-driven approach. Our work is potentially complementary to [15], since some of our findings may be used in the future to provide evidence for (or against) some of their purely theoretical claims.

Progress has been achieved in specific kinds of commonsense reasoning, especially in reasoning about time and internal relations [31], [48], reasoning about actions and change [43], and the sign calculus [9]. Semantics have played an important role in some of these successes [51]; in fact, ConceptNet itself has been described as a ‘semantic network’ [18]. A similar example is WordNet, whose entries are organized in terms of semantic inter-relationships. The easy-to-use network structure lends it to being frequently applied in multiple reasoning systems [7], [1], [37]. Other relevant areas in AI that could potentially benefit from this work include graph-based models [10], meta-heuristics [40], sarcasm detection [30].

Commonsense reasoning systems are typically measured against benchmark tests, some well known examples of such datasets including abductive Natural Language Inference (aNLI) [46], HellaSwag [71] and Physical Interaction Question Answering (PIQA) [4]. However, in this article, our focus is on understanding the nature of commonsense by studying a knowledge base such as ConceptNet, rather than building a reasoning system that performs well on benchmarks. We note that our findings can potentially be used to enhance performance of such systems. Recent work that combines knowledge bases with language representation learning models to improve commonsense question answering performance has been promising.

While knowledge bases and knowledge graphs have been extensively used for representing encyclopedic knowledge and other domains [32], [41], [5], ConceptNet and Cyc are the only two examples that we are aware of that serve as commonsense KBs [40], [53]. Owned by Cycorp, Cyc’s knowledge base claims to be the ‘broadest, deepest, and most complete repository ever developed’, but given its proprietary nature, the claim is difficult to validate. It is not known how similar

---

1. https://en.wikipedia.org/wiki/Common_sense
2. https://www.darpa.mil/program/machine-common-sense
3. https://www.cyc.com/
the commonsense content of Cyc is to that of ConceptNet, but potentially, the
same methodology proposed in this paper could be applied to Cyc to replicate,
strengthen or refute some of our findings. ConceptNet also expresses knowledge
in an easy-to-use format (i.e., as sets of 3-tuple assertions, illustrated as edges in
Figure[1]) rather than in higher-order logic notation. Because of these advantages,
ConceptNet has rapidly emerged as a practical dataset and resource for various
kinds of machine learning and NLP in the last ten years in particular [59], [8], [36].

Compared with WordNet [41], which focuses on preserving lexicographic infor-
mation and the relationship between words and their senses, ConceptNet maintains
a semantic network structure that is designed to capture commonsense assertions.
In particular, ConceptNet contains more relations than WordNet.

In contrast to Cyc, ConceptNet is a freely available multilingual KG that con-
nects everyday entities using a rich set of relations. As mentioned in the introduc-
tion, ConceptNet serves as an important background resource for commonsense
reasoning and question answering. However, it has not been studied directly for
gaining insights into commonsense knowledge, even though there is precedent.
For example, studies on DBpedia and YAGO have been conducted specifically to
understand their relational structure and the structural properties of the encyclo-
depedic knowledge that these KGs are known for [50], [52], [13]. We attempt to do
the same, but with commonsense knowledge as the focus.

Another line of work highly related to this one is representation learning, es-
pecially as they pertain to KGs. Such algorithms take as input the KG, including
entities and relations, and embed them into continuous vector spaces, as surveyed
by [64]. Models such as RESCAL [15], TransE [6], TransH [55], KG2E [20], and
RotateE [62] all achieve good performance on various tasks, such as KG comple-
tion [6] and relation extraction [67], which allows for further improvement of the
KG. In this article, we use a graph embedding package that builds on the ones
above (especially TransE) and is especially designed for graphs with many millions
of triples.

Finally, we note recent advances in commonsense reasoning and question an-
swering by using transformer-based language representation learning models such
as Bidirectional Encoder Representations from Transformers (BERT) [11], GPT-
3 [14], UnifiedQA [26], and several others. There is also a trend in improving
zero-shot learning with commonsense reasoning [44], [69]. Separately, ensemble
applications of symbolic and sub-symbolic AI models have also been proposed
for commonsense reasoning [61], [68]. While these advances have led to state-of-
the-art performance across NLP tasks, of which question answering is only one
example, these models do not help us understand the nature of commonsense
reasoning itself. Developing such an understanding is necessary both as a scien-
tific mode of investigation into human commonsense (which is ubiquitous), and
to further improve transformers and other neural models to be more interpretable
when producing answers to questions. This work does not propose a new method
for achieving higher task-specific performance on commonsense benchmarks, but
rather, proposes a computational methodology for studying structured common-
sense knowledge. By applying a data-driven and empirically grounded method-
ology on a dataset like ConceptNet, our hope is that we can learn more about
commonsense as a fundamental phenomenon.
3 Materials and Methods

Our guiding principle in this paper is that a commonsense KB such as ConceptNet could be used as the basis for understanding some of the structural properties of commonsense knowledge. Conceptually, ConceptNet can be defined as a multi-relational graph $G = (V, R, E)$, where $V$ is the set of entities or nodes, $R$ is the set of 49 relations and $E$ is the set of triples or edges where each triple $e = (h, r, t) \in E \subseteq V \times R \times V$. While $G$ can also be thought of in a non-graph-theoretic way as a set of triples[$^5$], the distinction is not relevant for the purposes of this paper, and we use knowledge bases and graphs interchangeably. However, it is more natural to think about ConceptNet as a graph due to its structural properties. Earlier, Figure 1 expressed a fragment of ConceptNet as a multi-relational graph with 21 edges, or alternatively, 21 triples. Another example-set of actual triples from ConceptNet is also provided in Table 1.

Table 1: Five actual example triples (each using a different relation) from ConceptNet

| Example triple                                      |
|-----------------------------------------------------|
| ![Example triple image](image.png)                  |

The symbols, $h$ and $t$, are respectively referred to as the source (or the head) and the destination (or the tail) entity, and $r$ is the relation. For the purposes of maintaining consistent terminology, we use the terms triple, head, tail, and relation to refer to the elements $e$, $h$, $t$ and $r$ respectively. Where applicable, these head and tail entities are collectively referred to as entities.

Entities and relations are projected into a continuous vector space by PyTorch-BigGraph (PBG) for further computation and analysis [34]. Next, we briefly introduce the PBG system. We then describe the raw data and our sampling methodology, followed by the setup and training of the PBG system on our sample.

3.1 PyTorch-BigGraph (PBG) System

PBG is an efficient and recent embedding system for learning real-valued vector representations of nodes and edges (referred to as ‘entities’ and ‘relations’ in the context of KGs like ConceptNet) in large-scale graphs. It was developed and publicly released[$^6$] by Facebook AI Research [34]. It is able to deal with large-scale graphs because it relies on distributed computing, in addition to other high-scale techniques, such as graph partitioning and batched negative sampling. At present, it also supports GPU training.

---

[$^5$] This definition is sometimes used in the early natural language community when discussing these datasets as knowledge bases rather than (multi-relational) knowledge graphs, where triples are usually interpreted as directed edges [25].

[$^6$] https://github.com/facebookresearch/PyTorch-BigGraph.
PBG operates by first training on an input graph, which can be a knowledge graph with labeled edges or ‘relations’. Similar to other knowledge graph representation learning algorithms (discussed also in the Related Work) like TransE and RESCAL [6], [45], it outputs embeddings by optimizing an objective, whereby unconnected entities are pushed further apart in the vector space, and entities that are ‘adjacent’ (connected via an edge) are pushed closer to each other.

Compared to network embeddings such as DeepWalk and node2vec [47], [16], PBG supports multi-entity and multi-relation graph embeddings. Its embedding quality has been found to be comparable with (or even exceed) existing KG embedding systems, evaluated on the Freebase [5], LiveJournal [2] and YouTube [63] graphs.

3.2 Raw Data, Sampling and Representation Learning

We use a recently released version 7 of ConceptNet 5.7, for the empirical study in this paper. One important aspect of the dataset is the ratio of the number of unique entities to the total number of triples, which is much higher in ConceptNet (0.833) than in other similarly-sized KGs such as Freebase (0.055) or WordNet 8. Additionally, while ConceptNet tends to contain more entities than ‘encyclopedic’ KGs such as Freebase or DBpedia, there are fewer unique relations.

In practice, these significant deviations in expected entity/relation ratios can cause problems for KG representation learning packages, even those designed for large-scale KGs such as the PBG system. For example, while embedding the full ConceptNet knowledge base, we found that, during the training process, the PBG algorithm fails with a ‘bus error’ message if the number of input triples exceeds 4 million. This occurs due to the algorithm running out of shared memory (despite the fact that we execute the algorithm on a machine with 60+ GB memory). To address the memory issue and ensure that our results can be extended or replicated in the future using reasonable computation resources, we randomly sampled 4 million triples from ConceptNet for this study. Key statistics are tabulated in Table 2. The head entities-set is twice the size of the tail entities-set and their overlap is approximately 1/20 of the total entities. Cursory analysis also showed that the head entity ‘/c/en/person’ and tail entity ‘/c/fr/francais’ were found to have the most triples associated with them. Other relation-specific statistics are tabulated in Table 3. ’/r/RelatedTo’ was found to be the most frequent relation, occurring in more than 1 million triples.

| Num. triples | Num. entities | Num. head entities | Num. tail entities | Overlap |
|--------------|---------------|--------------------|--------------------|--------|
| 4,000,000    | 3,933,840     | 2,781,892          | 1,387,571          | 235,623 |

Table 2: The numbers of triples and entities, including separate head and tail entity counts, and their overlap (the number of entities that serve both as head entity or tail entity, possibly in different triples, in the dataset), in the sample of ConceptNet considered in this study.

7 Downloaded at https://github.com/commonsense/conceptnet5/wiki/Downloads
8 For the interested reader, we are specifically referring to the FB15k and WN18 datasets, as designated in multiple papers on knowledge graph embeddings [58], [66], [72].
We input these 4 million triples into the PBG algorithm for representation learning. We partition the 4 million sampled triples into training, validation and test datasets, containing 75%, 12.5%, 12.5% of the total triples, respectively. Before doing the sampling, we remove triples with the ‘ExternalURL’ relation. ExternalURL is a ‘non-semantic’ relation that is only referring to a URL identifier and cannot be used for structural analysis of the kind proposed in this paper. Finally, we train and validate PBG on a single server in the Amazon cloud with 4 Intel Xeon cores, with one socket and 61 GB of RAM. After training is concluded, the algorithm outputs a single vector for each unique relation and entity in the training dataset. In the next section, we discuss the validation of the quality of these embeddings.

### 3.3 Validating Quality of Embeddings

Due to the sampling described earlier, a reasonable question arises as to whether the quality of the learned representations or ‘embeddings’ output by PBG can be trusted. We propose and use a quantitative measure to validate the quality and effectiveness of these embeddings. Specifically, we first compute a centroid vector for each relation, as described below. Recall that we denoted the graph using the symbol $G = (V, R, E)$, where $E$ was the set of triples or ‘edges’ in the graph. In a slight abuse of notation, we use the symbol $G_{E}$ to represent the set $E$ associated with $G$.

Given a relation $r \in R$, let $G_r \subseteq G_E$ be the subset of triples in $G_E$ with relation $r$. For each such triple $(h, r, t)$ in $G_r$, we define the translation vector $v = t - h$, where $t$ and $h$ are the embeddings output by PBG for entities $t$ and $h$, respectively. The centroid vector $r_c$ of $r$ is defined simply as the mean of the translation vectors in $G_r$.

| Relation                  | Num. triples | Num. entities | Relation                  | Num. triples | Num. entities |
|----------------------------|--------------|---------------|----------------------------|--------------|---------------|
| LocatedNear                | 13           | 26            | LocatedNear                | 13           | 19            |
| LocatedIn                  | 13           | 23            | LocatedIn                  | 13           | 19            |
| LocatedAt                  | 13           | 19            | LocatedAt                  | 13           | 19            |
| LocatedInRegion            | 27           | 151           | LocatedInRegion            | 27           | 151           |
| NotHasProperty             | 44           | 81            | NotHasProperty             | 44           | 81            |
| NotCapableOf               | 72           | 132           | NotCapableOf               | 72           | 132           |
| Entails                    | 73           | 134           | Entails                    | 73           | 134           |
| HasProperty                | 87           | 168           | HasProperty                | 87           | 168           |
| InstanceOf                 | 81           | 243           | InstanceOf                 | 81           | 243           |
| DefinedAs                  | 433          | 812           | DefinedAs                  | 433          | 812           |
| ReceivesAction             | 1,003        | 1,480         | ReceivesAction             | 1,003        | 1,480         |
| HasLastSubevent            | 571          | 867           | HasLastSubevent            | 571          | 867           |
| HasFirstSubevent           | 2,625        | 3,267         | HasFirstSubevent           | 2,625        | 3,267         |
| MadeOf                     | 2,715        | 2,936         | MadeOf                     | 2,715        | 2,936         |
| HasA                       | 2,962        | 3,908         | HasA                       | 2,962        | 3,908         |
| HasProperty                | 3,573        | 4,885         | HasProperty                | 3,573        | 4,885         |
| Causes                     | 3,705        | 4,737         | Causes                     | 3,705        | 4,737         |
| CausesDesire               | 4,121        | 4,185         | CausesDesire               | 4,121        | 4,185         |
| AtLocation                 | 4,497        | 5,983         | AtLocation                 | 4,497        | 5,983         |
| SimilarTo                  | 6,980        | 10,676        | SimilarTo                  | 6,980        | 10,676        |
| HasContext                 | 10,003       | 13,530        | HasContext                 | 10,003       | 13,530        |
| IsA                        | 11,996       | 12,186        | IsA                        | 11,996       | 12,186        |
| FormOf                     | 630,914      | 912,022       | FormOf                     | 630,914      | 912,022       |
| RelatedTo                  | 1,001,359    | 1,536,157     | RelatedTo                  | 1,001,359    | 1,536,157     |

Table 3: The numbers of triples and entities corresponding to each of the 49 relations in the ConceptNet sample studied in this article. Note that an entity could occur in multiple triples, each with a different relation.
\[ \mathbf{r}_c = \frac{1}{|G_r|} \sum_{(h, r, t) \in G_r} (t - h) \]  

(1)

Note that this yields two distinct vectors for \( r \): the vector ‘directly’ output by the graph embedding (denoted as \( r \)) and the centroid vector \( \mathbf{r}_c \). We use the symbol \( \mathcal{R} \) to indicate the set of directly output embeddings for all 49 relations and the symbol \( \mathcal{R}_c \) to indicate the set of (derived) centroid vectors.

With this technical machinery in place, we validate our 4 million-triples sample as follows. First, we calculate two similarity lists, \( \mathcal{SL}_r \) and \( \mathcal{SL}'_r \), per relation, using each of these two notions of embedding a relation. Specifically, \( \mathcal{SL}_r \) is constructed as a list of the cosine similarities between \( r \) and each translation vector \( t - h \) in \( G_r \). The number of entries in \( \mathcal{SL}_r \) will equal \( |G_r| \). Similarly, \( \mathcal{SL}'_r \) is constructed as a list of cosine similarities between the centroid vector \( \mathbf{r}_c \) and each translation vector, and also has size \( |G_r| \). Furthermore, if we impose an arbitrary ordering on the triples in \( G_r \), the two similarity lists are aligned by virtue of the common translation vectors computed over triples in \( G_r \). However, in the general case, the values in \( \mathcal{SL}_r \) and \( \mathcal{SL}'_r \) will differ since the former relies on the direct embedding of \( r \) in its construction, while the latter relies on the centroid vector \( \mathbf{r}_c \).

Given these two per-relation lists, we establish that the two lists are, in fact, highly correlated. The Spearman’s rank correlation is designed to measure both the strength and direction of association between two ranked variables and ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation). Because of the geometric features of the embedding space, we are interested in whether there is correlation (i.e. the strength), rather than the direction of the correlation. For this reason, given the two aligned lists per relation (\( \mathcal{SL}_r \) and \( \mathcal{SL}'_r \)), we computed the absolute value of the Spearman’s rank correlation for each relation, in Table 4. As expected, some of the correlations are indeed negative. Specifically, of the 49 relations, 24 relations have a Spearman’s rank correlation greater than 0.6, while 25 other relations have negative correlations (approximately 50%, as would be statistically expected). However, in no case is the absolute value less than 0.4.

Since \( \mathbf{r}_c \) is a function of the entities in the triples, and never uses the direct embedding \( r \) output by PBG, this result serves as an independent check on the quality of the embeddings. The high absolute correlations show that, not only are the embeddings learned on our sample self-consistent, but also that they conform closely to the notion of translation that is an important feature of neural graph embeddings \(^9\). In contrast, if \( r \) had showed little or no correlation (compared to \( \mathbf{r}_c \)), it would have begged the question about whether the embeddings were learned by PBG in a sufficiently non-random way that, at least approximately, model the translation operation in vector space. Furthermore, to ensure the results are not an artifact of using Spearman’s correlation, we replicated it using an alternate measure (KL-Divergence), with similar conclusions. That is, the distributions of \( \text{vecr}_c \) and \( \mathbf{r}_c \) were found to exhibit low KL-Divergence for all relations\(^{10}\).

---

9 Note that, unlike \( r \), the translation vector (defined earlier as \( t - h \)) clearly depends on the triple.

10 As the conclusions are largely identical, we do not reproduce the KL-Divergence table herein.
Table 4: The Spearman’s rank correlation score between $SL_r$ and $SL'_r$, for each of the 49 relations. The methodology for constructing these two (aligned) similarity lists is described in the text.

3.4 Vectorizing and Clustering Relation-Specific Triples

Certain relations in ConceptNet are deliberately designed to be broad. A good example is the HasContext relation, which is defined on the ConceptNet website as: A HasContext B is declared in the knowledge base if ‘A is a word used in the context of B, which could be a topic area, technical field, or regional dialect’. In this article, we investigate the hypothesis that, despite being originally defined so broadly, there is considerable substructure in such relations. In considering the definition of HasContext above, multiple contexts are suggested e.g., technical field, regional dialect, and presumably, other contexts that may be similar to these explicit cases. Another example is a relation such as FormOf, where a triple A FormOf B may be declared if ‘A is an inflected form of B; B is the root word of A’. Even the basic official definition suggests breadth, since A could either be an ‘inflected’ form of B, or the ‘root word’ of B. Furthermore, there is nothing in the definition that places a strict constraint on such triples, either in theory or in practice.

Since ConceptNet is crowdsourced to a great extent, it is quite likely that many people have interpreted these relations at ‘face value’ i.e., in accordance with what one would understand their ‘everyday’ meaning to be. Therefore, our goal is to measure and describe the empirical substructures, if any, in these three specific relations (HasContext, FormOf and SymbolOf\(^{11}\)) using a systematic methodology. An important aspect of these three relations is not just that they are defined broadly and are coarse-grained, but are also relatively high-volume. Within our sample of 4 million triples, HasContext, FormOf and SymbolOf are asserted in 133,038, 630,914 and 63,785 triples respectively. This provides an added incentive to study these relations further, since they are clearly central to the knowledge

\(^{11}\) The SymbolOf relation is succinctly defined as: the triple A SymbolOf B is asserted in the knowledge base if ‘A symbolically represents B’.
Understanding Substructures in Commonsense Relations in ConceptNet

base and its purpose of capturing commonsense knowledge as sets of assertions. While these are not the most voluminous relations\textsuperscript{12}, we aimed for a set of three relations that are expected to have different practices around them. We leave for future work to replicate our methodology for other such high-volume relations.

An established unsupervised methodology for discovering structure in large collections of data points is clustering \textsuperscript{24}. The relations in ConceptNet were meant to capture common, informative patterns from various data sources that feed into ConceptNet (along with crowdsourcing). If well-defined clusters exist, there is good evidence to suggest that these coarse-grained relations could be further sub-divided or ontologized (possibly by declaring relation-subtypes). By studying both the consistency of the clusters, as well as the subjective nature of data within them, we can start gaining insight into each relation. These insights allow us to gain an empirical understanding of concepts, such as ‘context’ and ‘form’, that are important in commonsense reasoning and communication, beyond their theoretically broad definitions.

Most established clustering algorithms require the collection and representation of data points to be described in advance. In our case, the goal is to cluster asserted triples of the form \((h, r, t)\) in three independent experiments (with \(r\) belonging to one of HasContext, FormOf and SymbolOf in each experiment). However, such a clustering would require us to represent each triple as a vector. If the relation is fixed, as it would be within an experiment, we can represent the triple using translation vector \(t - h\) that we earlier introduced, and with the entity embeddings \(t\) and \(h\) output by PBG.

For the clustering algorithm itself, we chose to use the classic k-Means algorithm \textsuperscript{17}. There were several reasons, including the large numbers of data points (which requires efficient clustering), the lack of a task-specific objective function or training labels, and importantly, the methodological preference for an established and reasonably robust clustering algorithm.

To briefly review k-Means, the algorithm works iteratively to partition the dataset into \(k\) clusters, each of which is disjoint, owing to the clusters constituting a partition. Let us assume a set \(D = \{d_1, \ldots, d_n\}\) of \(n\) data points, each of which is \(q\)-dimensional. We set up the algorithm so that the \(k\) means or clusters are randomly initialized, and each of the \(n\) data points are assigned to exactly one of the \(k\) means, depending on which cluster the point is closest to. Next, the mean for each cluster is re-computed by taking the mean of the vectors assigned to that cluster. The steps above are then repeated: each of the \(n\) data points is re-assigned to exactly one of the \(k\) clusters (ties are broken arbitrarily), depending on which cluster’s mean it is closest to. The means are then re-computed, and so on. We run the algorithm till convergence is achieved, and cluster-assignment of points does not change from one iteration to the next.

Note that \(k\) is a hyper-parameter that must be predefined prior to executing the algorithm. There are several ways to obtain the ‘best’ value of \(k\) given a collection of points. The underlying commonality between these methods is to compute, for each value of \(k\), an error ‘score’, with lower values implying better quality. This

\textsuperscript{12} For example, RelatedTo and Synonym have more than a million triples each, the reason for their breadth (and high volume) is more evident than for a relation such as HasContext (for example), since context is a much more ambiguous concept in commonsense reasoning. We hypothesize that a relation like Synonym will behave similarly as a relation like SymbolOf, although we leave for future work to investigate it.
score is computed from the clusters obtained after executing k-Means for that $k$. In practice, $k$ is varied over a predetermined range. By plotting the error score versus $k$, and looking for sudden shifts in the second derivative of the curve, we can determine a value of $k$ that captures the structure in the data. Intuitively, we are seeking a clear ‘bend’ in the score vs. $k$ curve to deduce where the second derivative is being minimized.

Fig. 2: Selection of the parameter $k$ for k-means clustering of HasContext triples, with $k$ on the x-axis and the clustering score on the y-axis, as measured (from left to right) by the elbow method, Silhouette Coefficient, Davies-Bouldin index, and Calinski-Harabasz index, respectively. The last two are in the same subplot.

A specific method that implements the $k$-selection principles conceptually described above is the elbow method [28]. The elbow method computes an error-based score based on the ‘dispersion’ of points within each cluster. However, as the first subplot in Figure 2 shows, there is no visible decline in the second derivative with $k$. Namely, the ‘bend’, if it even exists, is far too subtle to be useful as a decision-making criterion. For this reason, and also for purposes of robustly selecting a $k$, we also considered three other established alternatives – the Silhouette Coefficient method, the Davies-Bouldin Index, and the Calinski-Harabasz Index [70], [3]. The Silhouette Coefficient value measures how similar a point is to its own cluster’s centroid (cohesion) compared to other clusters’ centroids (separation). The Index-based measures compute their scores in slightly different ways, but with the same underlying philosophy that clusters should be cohesive and well-separated. Specific details and formulae may be found in the cited works.

As indicated in Figure 2, while both the elbow method and Davies-Bouldin index are inconclusive, the other two are fairly evident and consistent. In the case of the Silhouette Score, we find that there is an explicit minima at $k = 20$. In general, we found $k = 20$ to be a robust choice for all three relations (equivalent plots for SymbolOf and FormOf may be found in the supplementary information). The exercise illustrates the methodological utility of using more than one $k$-selection procedure.

Although the curve can be monotonic for some methods, it is not always guaranteed. Hence, it is incorrect to look for a ‘minimum’.
In future work, it may be possible to address the limitation of selecting \( k \) heuristically by using hierarchical or agglomerative clustering, and also by using recent clustering algorithms that do not require such hyperparameter selection (e.g., the recent work by [29]). Our main reason for using k-Means (along with hyperparameter selection methods such as elbow and Calinski-Harabasz Index) for this paper is that it is an established and scalable procedure that can be easily replicated without re-implementation or significant hyperparameter tuning. It also provides a reference and benchmark for future research that is looking to explore the use of other, more advanced algorithms for discovering finer-grained sub-structures in the ConceptNet relations (both the ones that are used in this paper, as well as others, such as RelatedTo, that are not considered in this paper).

4 Results

With the selected value of \( k = 20 \) in place, we conducted k-Means clustering for each of the three relations, as discussed earlier. Following the clustering, we computed quantitative metrics to measure the cohesion and separation of the clusters. As its name suggests, cohesion measures the extent to which the points in each cluster are tightly grouped together. In other words, a cluster with high dispersion has low cohesion. While this intuitive measure can be quantitatively measured in several ways, we consider a simple, easily interpretable methodology and formulae described below in detail.

First, for each cluster, we compute its centroid and then normalize all points in the clusters, as well as the computed centroid of the cluster. Specifically, let us assume \( k \) clusters (with \( k \) set to 20, as detailed earlier) \( \{C_1, \ldots, C_k\} \), which are non-overlapping, non-empty and form a partition over the set \( D \) of data points (with each data point \( d \in D \) being a vector \( d = [d_1, \ldots, d_q] \) with \( q \) dimensions, as discussed in Section 3.4) being clustered. The centroid \( c_m \) of a cluster \( C_m \) is defined using the formula below:

\[
c_m = \frac{\sum_{d_i \in C_m} d_i}{|C_m|} \tag{2}
\]

The centroid always exists, since each cluster is non-empty. The sum in the numerator is element-wise. Following centroid computation for each cluster, we normalize each vector \( d \) in \( D \) as well as all \( k \) centroids (which are also vectors, with the same dimensionality \( q \) as \( d \)), such that the vector now lies on the unit hypersphere, we use the following formula:

\[
v' = \frac{v}{\sqrt{\sum_{i=1}^{i=q} |v[i]|^2}} \tag{3}
\]

Here, \( v \) is any vector from \( D \) or a centroid vector, and \( v' \) is the normalized vector. The division is again element-wise, and \( v[i] \) is the \( i^{th} \) element of the vector \( v \).

14 Since the graph embeddings are not normalized to lie on a unit-radius hypersphere, we normalize the embeddings before computing the distance to enable cross-cluster comparisons, as well as comparisons with the (subsequently described) separation measures.
Next, we calculate the average *Euclidean* distance (designated as $t_m$) between the normalized points in the cluster $C_m$ and the cluster’s centroid $c_m$ (which is also normalized):

$$t_m = \frac{\sum_{d_i \in C_m} Euc[d_i, c_m]}{|C_m|}$$

Here, $Euc[x, y]$ between two $q$-dimensional vectors $x$ and $y$ is $Euc[x, y] = \sqrt{\sum_{i=1}^{q} (x_i - y_i)^2}$, and is a scalar. Since smaller distances indicate greater cohesion, we subtract the average from 1 to obtain a cohesion $coh_m = 1 - t_m$ of the cluster $C_m$ on a scale of 0.0 to 1.0, with 1.0 indicating perfect cohesion i.e., all points inside the cluster coincide after normalization). In Table 5, we report the cohesion for each of the 20 clusters obtained, for each of the three relations. We also report the means and standard deviations, for each of the three relations being studied. Formally, given the cohesions $coh_1, \ldots, coh_k$, the mean cohesion $M_{coh}$ is given by the formula $\frac{\sum_{i=1}^{k} coh_i}{k}$, and the standard deviation $Std_{coh}$ is $\sqrt{\frac{\sum_{i=1}^{k} (coh_i - M_{coh})^2}{k}}$.

| Cluster ID | FormOf | HasContext | SymbolOf |
|------------|--------|------------|----------|
| 0          | 4.517  | 5.953      | 5.358    |
| 1          | 4.576  | 5.386      | 4.815    |
| 2          | 4.085  | 4.661      | 5.436    |
| 3          | 4.254  | 4.640      | 5.578    |
| 4          | 4.785  | 5.738      | 4.703    |
| 5          | 4.503  | 5.555      | 4.735    |
| 6          | 4.600  | 3.284      | 3.677    |
| 7          | 3.601  | 5.072      | 4.854    |
| 8          | 3.966  | 4.054      | 4.624    |
| 9          | 4.547  | 3.731      | 4.730    |
| 10         | 4.365  | 4.297      | 6.319    |
| 11         | 4.741  | 3.371      | 4.553    |
| 12         | 4.214  | 4.215      | 4.647    |
| 13         | 3.918  | 5.398      | 4.031    |
| 14         | 5.300  | 4.334      | 4.715    |
| 15         | 4.451  | 3.739      | 4.585    |
| 16         | 4.684  | 5.515      | 4.443    |
| 17         | 4.184  | 4.987      | 4.132    |
| 18         | 5.151  | 3.677      | 4.533    |
| 19         | 4.119  | 3.649      | 5.149    |
| Mean       | 4.439  | 4.562      | 4.831    |
| Std. Dev.  | 3.302  | 13.588     | 6.084    |

Table 5: The *cohesion* of FormOf, HasContext and SymbolOf clusters, along with per-relation mean and standard deviation. Note that cluster IDs are assigned arbitrarily and independently across relations, and not ‘aligned’ in any way. The IDs are re-used again in visualizations in Figure 3 as well as in Table 7 (for HasContext only) where we provide examples of hasContext triples per cluster.

Based on the table, we find that the mean cohesion for FormOf, HasContext and SymbolOf clusters is 4.439, 4.562 and 4.831 respectively. While the mean cohesion scores of clusters in these three relations may seem close in value, their
standard deviations exhibit significant differences. The standard deviations of Has-Context cluster cohesion scores are generally higher than the standard deviations of the other two relations’ cohesion scores. This simple result suggests that Has-Context may be more diverse (and hence, more dispersed in embedding space) than the other two relations. Furthermore, while the deviation is inversely related to the number of triples corresponding to each relation, it is not linear. Finally, it is important to note that the absolute values here are less meaningful than the values relative to one other.

While cohesion is a good measure for characterize clusters, it is not adequate by itself. An ‘optimal’ cohesion can be obtained by assigning each point to its own cluster (in which case, the point becomes the centroid of the cluster). An additional metric, even after controlling for $k$, is the separation of the clusters i.e., how ‘far apart’ the different clusters are in the embedding space. Similar to cohesion, there are multiple mathematical ways to capture this qualitative notion. We employ a simple method that is analogous to the cohesion measure—namely, for a given cluster $C_m$, we compute its separation $s_m$ by computing the average Euclidean distance from its centroid $c_m$ to each of the other $k - 1 = 19$ centroids. For simplicity, let us define the centroid-set $C = \{c_1, \ldots, c_k\}$ as the set of (normalized) centroids of all $k = 20$ clusters:

$$s_m = \frac{\sum_{c_i \in C, c_i \neq c_m} Euc(c_i, c_m)}{k - 1} \quad (5)$$

Note that a subtraction from 1.0 is not necessary (as was the case for the cohesion computations), since the higher the average Euclidean distances between the centroids, the higher the separation. Table 6 reports the results for all three relations, along with the mean and standard deviation. The same formulae apply for the mean and standard deviation as noted earlier for cohesion, the only difference being that we use the separations rather than the cohesions.

We find that, once again, HasContext has highest average separation (4.622). This further suggests that the ‘contexts’ represented by these clusters are well-separated. The FormOf and SymbolOf clusters obtain similar average separations of 2.985 and 2.970, respectively. Unlike cohesion, the standard deviation of cluster separation scores is highest for the SymbolOf relation.

In comparing the cohesion and the separation of clusters for all three relations in Tables 5 and 6, we find that mean separation of HasContext clusters is close to their mean cohesion. In other words, the mean distance from a cluster centroid to a within-cluster data point is similar to the mean distance from that cluster-centroid to other cluster-centroids. The mean separations of FormOf and SymbolOf clusters are lower than the respective mean cohesions, suggesting that substructures in these two relations may be less independent than those in HasContext.

We can also visualize the clustering results by first performing dimensionality reduction (to two dimensions) using the t-Stochastic Neighbor Embedding (t-SNE) method, which has emerged as a state-of-the-art neural visualization technique in the machine learning community [38]. Next, we plot these points in 2D space by using a different color to represent each cluster. Results for all three relations are visualized in Figure 3. For all three relations (and especially, FormOf), there are some homogeneous clusters, where the embeddings are close to each other. However, other clusters can exhibit dispersion. For both SymbolOf and HasContext,
some clusters exhibit high dispersion and overlap with other clusters. These dispersed clusters provide an explanation for why HasContext and SymbolOf were found to have much high standard deviations on both the separation and cohesion measures described earlier (compared with much lower standard deviations for FormOf).

It is also instructive to study the substructures for a given relation by further analyzing triples qualitatively within each of the 20 clusters. We conduct such a qualitative study for the HasContext relation. Specifically, we randomly sample...
five triples per cluster to determine if we can deduce the ‘theme’ of the cluster from these five triples. These samples are reproduced in Table 7.

Even the limited set of sampled triples (per cluster) in the table demonstrates a pattern. For example, some of the clusters clearly embody scientific ‘domains’ of study such as geography (Cluster 14), chemistry (Cluster 6), medicine (Cluster 17), mathematics (Cluster 18), and physics (Cluster 19). However, there are also ‘mixed’ clusters that seem to be related to more than one theme, at least on the surface. Cluster 3 contains some ‘locality’ triples, even though Cluster 15 is predominantly concerned with localities, and Cluster 3 is mainly concerned with zoology.

We believe that there could be two causes or interpretations for such ‘confusion’. The first is due to the automatic and unsupervised nature of the embedding and the second is due to ConceptNet itself, both in terms of the noise within ConceptNet and also because some tail entities, such as /c/fr/localites, may be imbalanced in terms of the head entities associated with them. Some other clusters also include some interesting combinations, such as Cluster 12 which contains triples corresponding to both ‘computing’ and ‘slang’. In the embedding and clustering process, ‘computing’ and ‘slang’-related triples are thought to be close to each other in vector space, although their semantic similarity is not intuitively evident. Yet other clusters, such as Cluster 0, seem to encapsulate the broad notion of HasContext, and do not have an evident thematic classification that we can determine.

Additional interpretations of these overlapping clusters are also possible. One interpretation is historical context, especially concerning how these fields (‘sub-structures’) have co-evolved over time. For example, fields such as mathematics, physics and even computing have become increasingly entwined over time. The overlap may simply be a consequence of this co-evolution. Another possibility is that the fields share common semantics (including common terms), and this is reflected in overlap as well. Finally, we note that ConceptNet is meant to be a repository of commonsense knowledge, and not necessarily a detailed description of highly specialized domains within science (or other areas). The overlap may be arising not because the fields themselves are highly overlapping but because their commonsense content is. We leave for future work to investigate these interpretations more rigorously and quantitatively.

Aside from hypothesis validation of commonsense knowledge and evolution of commonsense semantics in specialized areas, the sub-structures could be used in novel domain-specific applications of AI, including domain-specific versions of fuzzy clustering and expert systems [35], [27], [12]. One such application is in knowledge acquisition: our discovered sub-structures could be used for isolating specific portions of, or triples within, ConceptNet that are useful for the application domain being considered. Once isolated, these triples could be used in a domain-specific fuzzy clustering algorithm or expert system to enhance performance. Further research is needed to build and evaluate such applications.
Table 7: Examples of five triples randomly sampled from each of the 20 HasContext clusters. The cluster IDs are consistent with those used in Figure 3 (left figure). Columns 2 and 3 list the head and tail entities respectively, with the relation always being hasContext, by definition.

| Cluster ID | Head Entity | Tail Entity |
|------------|-------------|-------------|
| 0          | /c/en/immunosenescent/a | /c/en/pathology |
| 1          | /c/en/immunosenescent/a | /c/en/philosophy |
| 2          | /c/en/immunosenescent/a | /c/en/philosophy |
| 3          | /c/en/immunosenescent/a | /c/en/philosophy |
| 4          | /c/en/immunosenescent/a | /c/en/philosophy |
| 5          | /c/en/immunosenescent/a | /c/en/philosophy |
| 6          | /c/en/immunosenescent/a | /c/en/philosophy |
| 7          | /c/en/immunosenescent/a | /c/en/philosophy |
| 8          | /c/en/immunosenescent/a | /c/en/philosophy |
| 9          | /c/en/immunosenescent/a | /c/en/philosophy |
| 10         | /c/en/immunosenescent/a | /c/en/philosophy |
| 11         | /c/en/immunosenescent/a | /c/en/philosophy |
| 12         | /c/en/immunosenescent/a | /c/en/philosophy |
| 13         | /c/en/immunosenescent/a | /c/en/philosophy |
| 14         | /c/en/immunosenescent/a | /c/en/philosophy |
| 15         | /c/en/immunosenescent/a | /c/en/philosophy |
| 16         | /c/en/immunosenescent/a | /c/en/philosophy |
Further investigation of the inter-relationships between these clusters in Figure 3 yields other insights. For example, the cluster focused on chemistry-related triples overlaps with the cluster containing biology, as well as with astronomy-related triples, as we would intuitively expect. While some of the overlap in the figure is exaggerated due to dimensionality reduction, it is nonetheless indicative of the low separation between these two clusters in high-dimensional space. It is an indirect acknowledgement of the shared lineage of these scientific disciplines. An interesting avenue for future exploration is to quantify and explain the observed topical overlap between such clusters, by using techniques such as hierarchical clustering.

5 Discussion

In exploring three specific relations (SymbolOf, FormOf, and HasContext), we found and characterized significant ‘substructures’ that are thematically diverse (especially in the case of HasContext), illustrating distinct and complex sub-relations within the overall relation. Sub-structures were also noted in FormOf and SymbolOf, but were less interesting and had clear separations than HasContext.

We have also empirically observed that, while ‘super-class’ semantics tend to be associated with the definition of HasContext, there are significant substructures that can’t be uniformly explained by an ‘umbrella’ term like HasContext. These substructures may help us better understand what the different contexts are in which people interpret pairs of words or entities. Understanding context is critical for building systems that have commonsense, such as chatbots and conversational agents, that need to understand sentences in the specific context in which the sentences are uttered.

Another interesting case is SymbolOf. The ratio of the size of the head-entity set of SymbolOf to the size of its tail-entity set is 0.011, which is almost 1/200 of the average value observed for other relations. This potentially implies that the number of words or phrases that are used to ‘describe’ the symbols is far greater than the

---

\[ h \] is a word that is used in the context of tail entity \( t \), \( t \) tends to be a more general, abstract ‘super-class’ of \( h \), such as a topic area, technical field, or regional dialect, as is also mentioned in the official definition of HasContext.
number of symbols themselves. Symbols, and arguably, by extension, the emojis used on social media, are necessarily under-determined, and the semantics assigned to symbols vary in different contexts. While not qualitatively unsurprising, our results suggest that the ‘diffusion’ of symbol-semantics is far greater than one might have thought. We leave a detailed study of this diffusion for future research.

6 Conclusion and Future Work

Commonsense knowledge and reasoning are not only ubiquitous among humans, they have also been recognized as essential for building ‘general’ AI architectures with human-like reasoning abilities, especially when facing uncertain, implicit, or even potentially contradictory, information. Despite much progress in commonsense-related tasks such as question answering, there is a lack of understanding of structural properties of commonsense knowledge. At the same time, the release and growth of commonsense knowledge graphs, such as ConceptNet, has provided an opportunity to conduct such a study using rigorous and replicable computational techniques.

In this article, we presented and applied such a data-driven methodology to understand structure in commonsense assertions through an empirical study of three high-volume, coarse-grained relations, namely, FormOf, SymbolOf and HasContext. All of these relations (and especially, hasContext) are known to be important in everyday commonsense tasks, including communication and conversation. Using both qualitative and quantitative analyses, we found that there are at least 20 distinct kinds of context that can be discovered within ConceptNet, some very well-defined (such as a scientific field of study), with others being more diffuse. Similar findings seem to apply for the other two relations. In some cases, there are unusual, but non-random, degrees of overlap and association between contexts, such as computer science and slang. Some of these contexts could be used to semi-automatically refine ConceptNet and develop more comprehensive ontological resources for the NLP community. Similarly, different sub-categories of symbols and forms can be semi-automatically discovered.

There are several promising opportunities for future research. For example, it may be worthwhile going even deeper into a relation like HasContext to discover if there are hierarchical substructures, rather than a single set of substructures. Hierarchical clustering algorithms in the graph embedding space could be used to achieve this goal methodologically. An even more ambitious line of study would be to connect these empirical results to theoretical claims (e.g., by [15]) about commonsense knowledge. Replicating the methodology on other coarse-grained and high-volume relations, both in ConceptNet and other knowledge bases, is also a valuable avenue for future investigation.

Finally, while the use of ConceptNet and other static resources is valuable for studying commonsense knowledge at a given time, they may not necessarily account for dynamic changes in meanings and semantics that occur over time. Future research may want to take into account other graph structures that can grow

---

16 One reason for this implication is that the head entity of ‘SymbolOf’ is a symbol and the tail entity is some kind of description, name or other information pertinent to characterizing that symbol.
and change over time, such as co-occurrence graphs. From those graphs, coarse-grained relationships could be extracted, similar to the approach presented in this paper, and the consequently extracted sub-structures might better reflect topical relationships with respect to a reference time-slice. Additionally, taxonomies can be automatically extracted from such sub-structures. Expanding the analysis in this paper using such co-occurrence graphs, rather than just the sub-graph from a single relation (within the context of a single experiment), is a promising area for future work.

**Declarations**

**Funding**

This project was funded under MOWGLI, a project in the Defense Advanced Research Projects Agency (DARPA) Machine Common Sense program, supported by United States Office Of Naval Research under Contract No. N660011924033. The views expressed in this paper are solely of the authors.

**Conflicts of interest/Competing interests**

The authors declare that they have no competing interests.

**Availability of data and material**

The subject of study is ConceptNet 5.7, downloadable from [http://www.conceptnet.io/](http://www.conceptnet.io/). Identifiers of the 4 million sampled triples are available at [https://drive.google.com/file/d/1RLHkwvuYmg0MqNf4UT0EDz5MmbKvX4s/view?usp=sharing](https://drive.google.com/file/d/1RLHkwvuYmg0MqNf4UT0EDz5MmbKvX4s/view?usp=sharing).

**Code availability**

The study relies on publicly available codebases that have been developed by others, including (i) t-SNE: [https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html](https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html) (ii) PBG: [https://github.com/facebookresearch/PyTorch-BigGraph](https://github.com/facebookresearch/PyTorch-BigGraph) and (iii) standard clustering and machine learning packages available at [https://scikit-learn.org/stable/](https://scikit-learn.org/stable/)

**Authors’ contributions**

Kejriwal was responsible for research supervision, project management, formulation of research question, and editing. Shen was responsible for all core research, including experimental work and writing.
References

1. Angeli G, Manning C (2014) Naturalli: Natural logic inference for common sense reasoning. pp 534–545, DOI 10.3115/v1/D14-1059
2. Backstrom L, Huttenlocher D, Kleinberg J, Lan X (2006) Group formation in large social networks: Membership, growth, and evolution. vol 2006, pp 44–54, DOI 10.1145/1150402.1150412
3. Bandyopadhyay S (2015) Unsupervised Classification. Springer Publishing Company, Incorporated
4. Bisk Y, Zellers R, bras R, Gao J, Yejin C (2020) Pigqa: Reasoning about physical commonsense in natural language. Proceedings of the AAAI Conference on Artificial Intelligence 34:7432–7439, DOI 10.1609/aaai.v34i05.6239
5. Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD Conference
6. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in Neural Information Processing Systems 26, Curran Associates, Inc., pp 2787–2795, URL http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf
7. Botschen T, Sorokin D, Gurevych I (2018) Frame- and entity-based knowledge for common-sense argumentative reasoning. In: Proceedings of the 5th Workshop on Argument Mining, Association for Computational Linguistics, Brussels, Belgium, pp 90–96, DOI 10.18653/v1/W18-5211, URL https://www.aclweb.org/anthology/W18-5211
8. Chen M, Tian Y, Chang KW, Skiena S, Zaniolo C (2018) Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. arXiv preprint arXiv:180606478
9. Davis E, Marcus G (2015) Commonsense reasoning and commonsense knowledge in artificial intelligence. Communications of the ACM 58:92–103, DOI 10.1145/2701413
10. Devezas J, Nunes S (2021) A review of graph-based models for entity-oriented search.
11. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805
12. Diederich J, Ruhmann I, May M (1987) Kriton: a knowledge-acquisition tool for expert systems. International Journal of Man-Machine Studies 26(1):29–40
13. Färber M, Ell B, Menne C, Rettinger A (2015) A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. Semantic Web Journal 1(1):1–5
14. Floridi L, Chiariotti M (2020) Gpt-3: Its nature, scope, limits, and consequences. Minds and Machines 30(4):681–694
15. Gordon AS, Hobbs JR (2017) A formal theory of commonsense psychology: How people think people think. Cambridge University Press
16. Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp 855–864
17. Hartigan JA, Wong MA (1979) Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society Series C (Applied Statistics) 28(1):100–108, URL http://www.jstor.org/stable/2346830
18. Havasi C, Speer R, Alonso J (2007) Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In: Recent advances in natural language processing, Citeseer, pp 27–29
19. Havasi C, Speer R, Arnold K, Lieberman H, Alonso J, Moeller J (2010) Open mind common sense: Crowd-sourcing for common sense. In: Proceedings of the 2nd AAAI Conference on Collaboratively-Built Knowledge Sources and Artificial Intelligence, AAAI Press, AAAIWS’10-02, p 53
20. He S, Liu K, Ji G, Zhao J (2015) Learning to represent knowledge graphs with gaussian embedding. pp 623–632, DOI 10.1145/2806416.2806502
21. Hirschberg J, Manning CD (2015) Advances in natural language processing. Science 349(6245):261–266, DOI 10.1126/science.aaa8685, URL https://science.sciencemag.org/content/349/6245/261
22. Hobbs JR, Gordon AS (2005) Toward a large-scale formal theory of commonsense psychology for metacognition. In: AAAI Spring Symposium: Metacognition in Computation, pp 49–54
49. QI GuiLin WT GAO Huan (2017) The research advances of knowledge graph. Technology Intelligence Engineering 3:4–25

50. Rajabi E, Sanchez-Alonso S, Sicilia MA (2014) Analyzing broken links on the web of data: An experiment with dbpedia. Journal of the Association for Information Science and Technology 65(8):1721–1727

51. Rajagopal D, Cambria E, Ohser D, Kwok K (2013) A graph-based approach to commonsense concept extraction and semantic similarity detection. In: Proceedings of the 22nd International Conference on World Wide Web, pp 565–570

52. Rebele T, Nekoei A, Suchanek FM (2017) Using yago for the humanities. In: WHiSe@ISWC, pp 99–110

53. Roemmele M, Bejan C, Gordon A (2011) Choice of plausible alternatives: An evaluation of commonsense causal reasoning.

54. Sap M, Rashkin H, Chen D, Bros R, Yejin C (2019) Social isq: Commonsense reasoning about social interactions. pp 4453–4463, DOI 10.18653/v1/D19-1454

55. Sap M, Shwartz V, Bosselut A, Choi Y, Roth D (2020) Commonsense reasoning for natural language processing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, Association for Computational Linguistics, Online, pp 27–33, DOI 10.18653/v1/2020.acl-tutorials.7, URL https://www.aclweb.org/anthology/2020.acl-tutorials.7

56. Shi B, Weninger T (2017) Open-world knowledge graph completion. arXiv preprint arXiv:171103438

57. Smedslund J (1982) Common sense as psychosocial reality: A reply to sjöberg. Scandinavian Journal of Psychology 23(1):79–82

58. Song HJ, Kim AY, Park SB (2020) Learning translation-based knowledge graph embeddings by n-pair translation loss. Applied Sciences 10:3964, DOI 10.3390/app10113964

59. Speer R, Havasi C (2013) Conceptnet 5: A large semantic network for relational knowledge. The people’s web meets NLP, theory and applications of natural language processing pp 161–176

60. Speer R, Chin J, Havasi C (2017) Conceptnet 5.5: An open multilingual graph of general knowledge. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, AAAI17, pp 4444–4451

61. Storks S, Tao Y, Chai JY (2019) Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. arXiv preprint arXiv:190401172 pp 1–60

62. Sun Z, Deng ZH, Nie JY, Tang J (2019) Rotate: Knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=B1vE6tKQ0

63. Tang L, Liu H (2009) Scalable learning of collective behavior based on sparse social dimensions. In: CIKM

64. Wang Q, Mao Z, Wang B, Guo L (2017) Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering PP:1–1, DOI 10.1109/TKDE.2017.2754499

65. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI Press, AAAI14, pp 1112–1119

66. Wen Zhang LWJCHZWZAB Bibek Paudel, Chen H (2019) Iteratively learning embeddings and rules for knowledge graph reasoning. In: 2019 World Wide Web Conference (WWW’19)

67. Weston J, Bordes A, Yakhnenko O, Usunier N (2013) Connecting language and knowledge bases with embedding models for relation extraction. EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference

68. Xu Y, Zhu C, Xu R, Liu Y, Zeng M, Huang X (2020) Fusing context into knowledge graph for commonsense reasoning. arXiv preprint arXiv:201204808

69. Ye Q, Ren X (2021) Zero-shot learning by generating task-specific adapters. arXiv preprint arXiv:210100420

70. Yuan C, Yang H (2019) Research on k-value selection method of k-means clustering algorithm. J 2:226–235, DOI 10.3390/j2020016

71. Zellers R, Holtzman A, Bisk Y, Farhadi A, Yejin C (2019) Hellaswag: Can a machine really finish your sentence? pp 4791–4800, DOI 10.18653/v1/P19-1472

72. Zhao F, Sun H, Jin L, Jin H (2020) Structure-augmented knowledge graph embedding for sparse data with rule learning. Computer Communications 159, DOI 10.1016/j.comcom.2020.05.017