(COMET-)ATOMIC$^{20}$
On Symbolic and Neural Commonsense Knowledge Graphs

Jena D. Hwang$^{†,*}$, Chandra Bhagavatula$^{†,*}$, Ronan Le Bras$^{†}$, Jeff Da$^{†}$, Keisuke Sakaguchi$^{†}$, Antoine Bosselut$^{*†}$ and Yejin Choi $^{††}$

$^{†}$ Allen Institute for AI, WA, USA
$^{†*}$ Paul G. Allen School of Computer Science & Engineering, WA, USA
$^{‡}$ Stanford University, CA, USA

Abstract
Recent years have brought about a renewed interest in commonsense representation and reasoning in the field of natural language understanding. The development of new commonsense knowledge graphs (CSKG) has been central to these advances as their diverse facts can be used and referenced by machine learning models for tackling new and challenging tasks. At the same time, there remain questions about the quality and coverage of these resources due to the massive scale required to comprehensively encompass general commonsense knowledge.

In this work, we posit that manually constructed CSKGs will never achieve the coverage necessary to be applicable in all situations encountered by NLP agents. Therefore, we propose a new evaluation framework for testing the utility of KGs based on how effectively implicit knowledge representations can be learned from them.

With this new goal, we propose ATOMIC$^{20}$, a new CSKG of general-purpose commonsense knowledge containing knowledge that is not readily available in pretrained language models. We evaluate its properties in comparison with other leading CSKGs, performing the first large-scale pairwise study of commonsense knowledge resources. Next, we show that ATOMIC$^{20}$ is better suited for training knowledge models that can generate accurate, representative knowledge for new, unseen entities and events. Finally, through human evaluation, we show that the few-shot performance of GPT-3 (175B parameters), while impressive, remains $\sim12$ absolute points lower than a BART-based knowledge model trained on ATOMIC$^{20}$ despite using over 430x fewer parameters.

1 Introduction
Commonsense understanding and reasoning remain longstanding challenges in general artificial intelligence. However, large-scale language models have brought tremendous progress in the sub-field of natural language processing. Such large-scale language models (Radford et al. 2018; Devlin et al. 2019; Brown et al. 2020) trained on extreme-scale data have been shown to effectively adapt to diverse downstream tasks, achieving significant performance gains across natural language benchmarks (Wang et al. 2019). Interestingly, as these models have grown larger (and trained on larger amounts of data), their benchmark performance has continued to improve (Raffel et al. 2019) despite limited conceptual improvements, leaving open questions regarding the source of these remarkable generalization properties.

Recent work has hypothesized that many of these performance gains could be a result of language models being able to memorize facts in their parameters during training (Roberts, Raffel, and Shazeer 2020) that can be leveraged at evaluation time. As a result, a new paradigm of language models as knowledge bases has emerged (Petroni et al. 2019). In this setting, language models are prompted with natural language prefixes or questions, and they express knowledge through language generation. The initial success of this paradigm for representing commonsense knowledge (Davison, Feldman, and Rush 2019; Tamborrino et al. 2020) has led to the optimistic claim that language models comprehensively encode commonsense knowledge, and remove the need for structured knowledge resources.
We take a more skeptical view of this capacity of language models – Does scaling up language models actually endow them with commonsense knowledge? While language models can successfully express certain types of knowledge, their best results are observed in narrowly specific conditions – we show (cf. §5) that they perform better when evaluated on knowledge bases that prioritize ontological relations and whose examples resemble language-like assertions (e.g., mango isA fruit). Consequently, the types of knowledge that can be directly accessed through the language model’s interface remains limited.

However, prior work has also shown that training language models on knowledge graph tuples leads them to learn to express their implicit knowledge directly (Bosselut et al. 2019), allowing them to provide commonsense knowledge on-demand. These adapted knowledge models have exhibited promising results on commonsense benchmarks compared with methods that require linking entities to knowledge graphs (Shwartz et al. 2020; Liu et al. 2020). Inspired by these successes, we propose a dual use for commonsense knowledge bases going forward: as static graphs that can be linked to for discrete knowledge access, and as resources for adapting language models to hypothesize commonsense knowledge about un-annotated entities and events.

With this second purpose in mind, we propose evaluating commonsense knowledge resources based on the complementary information they can bring to pretrained language models. We construct ATOMIC$^{20}$, a new, high-quality knowledge graph with 1.33M commonsense knowledge triples across 23 commonsense relations. We compare ATOMIC$^{20}$ with respect to its coverage and accuracy in competition with other highly used CSKBs, such as CONCEPTNET (Speer, Chin, and Havasi 2017). Our results show that ATOMIC$^{20}$ is able to cover more correct facts about more diverse types of commonsense knowledge than any existing, publicly-available commonsense knowledge resource. However, our results also indicate that there remains a large amount of exclusivity between these KGs, highlighting the challenge of creating resources that cover the scale and diversity of general commonsense knowledge.

Furthermore, we formalize the COMET framework of Bosselut et al. (2019) across different seed language models and training knowledge graphs, and evaluate the commonsense knowledge hypothesized by these adapted knowledge models. Our empirical study yields two promising conclusions. First, it confirms that KG-adapted language models learn to express knowledge more precisely than naive language models trained only on language. And second, we show that ATOMIC$^{20}$ as a transfer resource leads to COMET models that achieve the largest increase over their seed language model (across all seed LMs) for the commonsense knowledge types it covers, validating the importance of constructing knowledge resources with examples of knowledge not readily found in language models.

Key Contributions: In summary, we make three key contributions in this paper. We present ATOMIC$^{20}$—a new commonsense knowledge graph covering social, physical, and eventive aspects of everyday inferential knowledge (cf. §3). Next, we compare ATOMIC$^{20}$ with other prominent CSKBs head-to-head and show that our new symbolic knowledge graph is more accurate than any current CSKB (see Table 2) (cf. §4). Finally, we show that our new neural knowledge model COMET-ATOMIC$^{20}$ successfully transfers ATOMIC$^{20}$’s declarative knowledge to beat GPT-3, the largest pre-trained language model, in spite of using 400x fewer parameters (see Table 6) (cf. §5). This demonstrates the utility and importance of high-quality symbolic knowledge provided by ATOMIC$^{20}$ to generalize on commonsense information that LMs cannot expressively capture on their own (cf. §6).

2 Background

Commonsense Knowledge Graphs Large scale commonsense knowledge graphs are ubiquitous tools in natu-
ral language processing tasks as access to their facts allows models to learn to reason over commonsense knowledge to make predictions (Lin et al. 2019; Feng et al. 2020). In this work, we evaluate three existing knowledge graphs, CONCEPTNET, ATOMIC, and TRANSOMCS on their coverage and precision relative to our new resource ATOMIC\textsuperscript{20} \textsuperscript{3}.

The CONCEPTNET (v5.7) knowledge graph (Speer, Chin, and Havasi 2017) consists of 36 relations focusing mostly on taxonomic and lexical knowledge (e.g., RelatedTo, Synonym, IsA) and physical commonsense knowledge (e.g., MadeOf, PartOf). CONCEPTNET (v5.7) contains 3.4M entity-relation tuples (in English) collected by crowdsourcing and merged with existing knowledge databases from DBPedia, WordNet, Wiktionary, and OpenCyc. Since the knowledge are derived from human efforts, the accuracy of CONCEPTNET (v5.7) knowledge is fairly high, though the quality does vary depending on the sources of knowledge and relation types. However, as highlighted in (Davis and Marcus 2015; Sap et al. 2019), and shown in Figure 2, the coverage of CONCEPTNET (v5.7) is limited to mostly taxonomic, lexical, and object-centric physical commonsense knowledge. In fact, out of 3.4M tuples, 90% of them correspond to taxonomic (e.g., IsA) or lexical (e.g., Synonym, RelatedTo) knowledge, making the commonsense portion of CONCEPTNET (v5.7) relatively small.

The ATOMIC (Sap et al. 2019) knowledge graph consists of 880K of tuples across 9 relations that cover social commonsense knowledge (e.g., X gets X’s car repaired xIntent to maintain the car), including dynamic aspects of events such as causes and effects, if-then conditional statements, and mental states. The ATOMIC dataset is collected and validated completely through crowdsourcing.

The TRANSOMCS (Zhang et al. 2020a) knowledge graph consists of 18.48M tuples that were automatically converted from syntactic parses of sentences from various web sources including Wikipedia, Yelp, and Reddit. The set of relations used for the mapping is copied from CONCEPTNET. Although TRANSOMCS is much larger than other commonsense knowledge graphs, the precision of the extracted knowledge is significantly lower compared to other resources (cf. §4), and performs poorly as an adaptation resource relative to other KGs (cf. §5).

Language Models as Knowledge Bases Recent work hypothesizes that pretrained language models represent commonsense knowledge implicitly (Petroni et al. 2019; Roberts, Raffel, and Shazeer 2020). However, the results motivating these observations are often limited to narrowly scoped subsets of commonsense knowledge that primarily include taxonomic knowledge (e.g., mango IsA fruit) and that are often found explicitly stated in text. However, commonsense facts are often implied (Gordon and Van Durme 2013), and as will be seen in our studies (cf. §4), state of the art neural models struggle to express implicit commonsense knowledge that involves complex relationships.

To overcome this limitation, Bosselut et al. (2019) take the best of both worlds between commonsense knowledge graphs and pretrained language models. The commonsense transformer, or COMET, adapts pretrained neural language models by training on example tuples from commonsense knowledge graphs. It takes a head/source phrase and a relation (e.g., take a nap Causes) and generates the tail/target phrase (e.g., have energy). Bosselut et al. (2019) show that COMET trained on the CONCEPTNET and ATOMIC knowledge graphs is able to adapt to generate novel (and valid) commonsense knowledge tuples.

Importantly, these neural knowledge models can produce commonsense knowledge on-demand for any head entity that can be expressed through language. This flexibility allows them to be used out-of-the-box, and they have been applied to new, previously unexplored tasks, such as sarcastic comment generation (Chakrabarty et al. 2020), therapy chatbots (Kearns et al. 2020), and automated story plot generation (Ammanabrolu et al. 2020). These contributions show that progress on knowledge models opens up new downstream applications that were challenging to model before.

3 ATOMIC\textsuperscript{20} \textsuperscript{20}

We present ATOMIC\textsuperscript{20}, a commonsense knowledge graph with 1.33M everyday inferential knowledge tuples about entities and events. ATOMIC\textsuperscript{20} represents a large-scale commonsense repository of textual descriptions that encode both the social and the physical aspects of common human everyday experiences, collected with the aim of being complementary to commonsense knowledge encoded in current language models. ATOMIC\textsuperscript{20} introduces 23 commonsense relations types. They can be broadly classified into three categorical types: 9 commonsense relations of social interaction, 7 physical-entity commonsense relations, and 7 event-centered commonsense relations concerning situations surrounding a given event of interest. The full inventory of ATOMIC\textsuperscript{20} relations is listed in Table 1.

In terms of physical and event-centered commonsense, by far, the two largest new relations in ATOMIC\textsuperscript{20} are ObjectUse and HinderedBy. For ObjectUse, we focused on affordances of everyday objects such as “popcorn bucket” that may be used for “holding popcorn” or “storing things”. For HinderedBy, we explore the notion that many events in real world can be defeasible (Lascarides and Asher 1991) by collecting hindrances to goals that may be useful for tasks such as counterfactual reasoning. For example X’s desires to adopt a cat may be hindered by finding out that X is allergic to cats, which would necessitate X to adjust future actions accordingly (say, opt for hypoallergenic options like tortoises).

In the case of ObjectUse, we collected over 130K everyday object-use pairs by asking crowdworkers for necessary objects and their uses for each event in ATOMIC\textsuperscript{20}. For example, given “X eats popcorn” we elicited items such as “popcorn bucket” with their various expected uses. The number also reflects atypical usages gathered in a separate pass where workers were asked to provide creative or resourceful but feasible uses of the objects. Given “popcorn

\textsuperscript{3}We were unable to include Cyc (Lenat 1995) in our study due to the discontinuation of its research license and the cost of the commercial license (over $1M). CONCEPTNET includes a subset of Cyc – OpenCyc.
In this work, we compare our new ATOMIC\textsuperscript{20} knowledge graph to three other prominent CSKGs: ATOMIC\textsuperscript{20} (Sap et al. 2019), CONCEPTNET\textsuperscript{6} (Li et al. 2016), and TRANSOMCS (Zhang et al. 2020a). We measure the accuracy of tuples in each KG and compare the coverage of each CSKG w.r.t. other CSKGs head-to-head.

**Accuracy Assessment**

In order to assess the accuracy of the knowledge represented, 3K random instances were extracted from each of the knowledge graphs for a crowdsourced evaluation of the tuples.

**Human Evaluation Setup.** The evaluation was carried out through crowdsourcing on the Amazon Mechanical Turk platform. Workers were presented with knowledge tuples in

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Accept</th>
<th>Reject</th>
<th>No Judgment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC\textsuperscript{20}</td>
<td>91.3</td>
<td>6.5</td>
<td>2.2</td>
</tr>
<tr>
<td>ATOMIC</td>
<td>88.5</td>
<td>10.0</td>
<td>1.5</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>88.6</td>
<td>7.5</td>
<td>3.9</td>
</tr>
<tr>
<td>TRANSOMCS</td>
<td>41.7</td>
<td>53.4</td>
<td>4.9</td>
</tr>
</tbody>
</table>

Table 2: Accuracy - Percentage (%) of tuples in the knowledge base evaluated by human crowdsworkers as either always true or likely (Accept), farfetched/never or invalid (Reject), or unclear (No Judgment).

**Symbolic Knowledge Graph Comparison**

In this work, we compare our new ATOMIC\textsuperscript{20} knowledge graph to three other prominent CSKGs: ATOMIC\textsuperscript{20} (Sap et al. 2019), CONCEPTNET\textsuperscript{6} (Li et al. 2016), and TRANSOMCS (Zhang et al. 2020a). We measure the accuracy of tuples in each KG and compare the coverage of each CSKG w.r.t. other CSKGs head-to-head.

**Accuracy Assessment**

In order to assess the accuracy of the knowledge represented, 3K random instances were extracted from each of the knowledge graphs for a crowdsourced evaluation of the tuples.

**Human Evaluation Setup.** The evaluation was carried out through crowdsourcing on the Amazon Mechanical Turk platform. Workers were presented with knowledge tuples in

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Accept</th>
<th>Reject</th>
<th>No Judgment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC\textsuperscript{20}</td>
<td>91.3</td>
<td>6.5</td>
<td>2.2</td>
</tr>
<tr>
<td>ATOMIC</td>
<td>88.5</td>
<td>10.0</td>
<td>1.5</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>88.6</td>
<td>7.5</td>
<td>3.9</td>
</tr>
<tr>
<td>TRANSOMCS</td>
<td>41.7</td>
<td>53.4</td>
<td>4.9</td>
</tr>
</tbody>
</table>

Table 2: Accuracy - Percentage (%) of tuples in the knowledge base evaluated by human crowdsworkers as either always true or likely (Accept), farfetched/never or invalid (Reject), or unclear (No Judgment).

**Symbolic Knowledge Graph Comparison**

In this work, we compare our new ATOMIC\textsuperscript{20} knowledge graph to three other prominent CSKGs: ATOMIC\textsuperscript{20} (Sap et al. 2019), CONCEPTNET\textsuperscript{6} (Li et al. 2016), and TRANSOMCS (Zhang et al. 2020a). We measure the accuracy of tuples in each KG and compare the coverage of each CSKG w.r.t. other CSKGs head-to-head.

**Accuracy Assessment**

In order to assess the accuracy of the knowledge represented, 3K random instances were extracted from each of the knowledge graphs for a crowdsourced evaluation of the tuples.

**Human Evaluation Setup.** The evaluation was carried out through crowdsourcing on the Amazon Mechanical Turk platform. Workers were presented with knowledge tuples in

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Accept</th>
<th>Reject</th>
<th>No Judgment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC\textsuperscript{20}</td>
<td>91.3</td>
<td>6.5</td>
<td>2.2</td>
</tr>
<tr>
<td>ATOMIC</td>
<td>88.5</td>
<td>10.0</td>
<td>1.5</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>88.6</td>
<td>7.5</td>
<td>3.9</td>
</tr>
<tr>
<td>TRANSOMCS</td>
<td>41.7</td>
<td>53.4</td>
<td>4.9</td>
</tr>
</tbody>
</table>

Table 2: Accuracy - Percentage (%) of tuples in the knowledge base evaluated by human crowdsworkers as either always true or likely (Accept), farfetched/never or invalid (Reject), or unclear (No Judgment).

**Symbolic Knowledge Graph Comparison**

In this work, we compare our new ATOMIC\textsuperscript{20} knowledge graph to three other prominent CSKGs: ATOMIC\textsuperscript{20} (Sap et al. 2019), CONCEPTNET\textsuperscript{6} (Li et al. 2016), and TRANSOMCS (Zhang et al. 2020a). We measure the accuracy of tuples in each KG and compare the coverage of each CSKG w.r.t. other CSKGs head-to-head.

**Accuracy Assessment**

In order to assess the accuracy of the knowledge represented, 3K random instances were extracted from each of the knowledge graphs for a crowdsourced evaluation of the tuples.

**Human Evaluation Setup.** The evaluation was carried out through crowdsourcing on the Amazon Mechanical Turk platform. Workers were presented with knowledge tuples in

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Accept</th>
<th>Reject</th>
<th>No Judgment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC\textsuperscript{20}</td>
<td>91.3</td>
<td>6.5</td>
<td>2.2</td>
</tr>
<tr>
<td>ATOMIC</td>
<td>88.5</td>
<td>10.0</td>
<td>1.5</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>88.6</td>
<td>7.5</td>
<td>3.9</td>
</tr>
<tr>
<td>TRANSOMCS</td>
<td>41.7</td>
<td>53.4</td>
<td>4.9</td>
</tr>
</tbody>
</table>

Table 2: Accuracy - Percentage (%) of tuples in the knowledge base evaluated by human crowdsworkers as either always true or likely (Accept), farfetched/never or invalid (Reject), or unclear (No Judgment).
### Table 3: KG accuracy values broken down by relation.

<table>
<thead>
<tr>
<th>Atomic²⁰/²⁰</th>
<th>Atomic</th>
<th>Relation</th>
<th>CN</th>
<th>T-OMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>92.3</td>
<td>ALocation*</td>
<td>89.4</td>
<td>34.3</td>
<td></td>
</tr>
<tr>
<td>93.9</td>
<td>CapableOf*</td>
<td>84.4</td>
<td>50.0</td>
<td></td>
</tr>
<tr>
<td>94.6</td>
<td>Causes</td>
<td>90.0</td>
<td>50.0</td>
<td></td>
</tr>
<tr>
<td>96.9</td>
<td>Desires*</td>
<td>96.3</td>
<td>48.2</td>
<td></td>
</tr>
<tr>
<td>93.9</td>
<td>HasProperty*</td>
<td>86.3</td>
<td>52.4</td>
<td></td>
</tr>
<tr>
<td>82.3</td>
<td>ObjUse/UsedFor</td>
<td>96.3</td>
<td>31.6</td>
<td></td>
</tr>
<tr>
<td>98.5</td>
<td>NotDesires*</td>
<td>96.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96.9</td>
<td>HasSubevent</td>
<td>88.1</td>
<td>57.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HasFirstSubevent</td>
<td>93.8</td>
<td>52.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HasLastSubevent</td>
<td>95.6</td>
<td>38.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HasPrerequisite</td>
<td>94.4</td>
<td>30.0</td>
<td></td>
</tr>
<tr>
<td>75.4</td>
<td>MadeUpOf/MadeOf</td>
<td>88.1</td>
<td>15.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PartOf</td>
<td>71.9</td>
<td>46.5</td>
<td></td>
</tr>
<tr>
<td>96.9</td>
<td>HinderedBy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96.2</td>
<td>isAfter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95.4</td>
<td>isBefore</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96.2</td>
<td>isFilledBy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91.5</td>
<td>oEffect</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91.5</td>
<td>oReact</td>
<td>87.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.5</td>
<td>oWant</td>
<td>89.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87.7</td>
<td>xAttr</td>
<td>91.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.8</td>
<td>xEffect</td>
<td>87.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>93.1</td>
<td>xIntent/MotivByGoal</td>
<td>84.4</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>87.5</td>
<td>xNeed</td>
<td>85.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90.8</td>
<td>xReact</td>
<td>91.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96.2</td>
<td>xReason</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82.3</td>
<td>xWant/CausesDesire</td>
<td>90.0</td>
<td>35.9</td>
<td></td>
</tr>
</tbody>
</table>

Gray cells indicate statistically significant difference from Atomic²⁰ values. Dark gray cells signal instances where Atomic²⁰ values are significantly higher than its counterpart KB. Relational cognates have been grouped together and exact matches are asterisked (*) (cf. Table 1).

### Results.

According to Table 2, Atomic²⁰ outperforms other KGs in crowdsourced accuracy as shown in Table 2. Atomic ties with ConceptNet with reasonably high accuracy, while TOMATIC lags behind others with far lower accuracy. We provide a per-relation breakdown of accuracies in Table 3.

Between Atomic²⁰ and Atomic, the variations in the assessed accuracies are not found to be statistically significant. Among the Atomic²⁰ and ConceptNet relations that represent exact matches (marked with * in Table 3), the differences are either not statistically significant or when they are, Atomic²⁰ improves upon the associated facts, reflecting that the preprocessing stages of ConceptNet integration were helpful in improving the quality of these relations (§3). Among cognates in Atomic²⁰ and ConceptNet relations, two sets of relations fare significantly worse in Atomic²⁰ than in ConceptNet. The case of ObjectUse/UsedFor, this is likely due to the fact that Atomic²⁰’s ObjectUse includes atypical affordances (cf. §3). In an annotation setting where workers are asked to evaluate the truth or likelihood of an assertion rather than feasibility of use, a portion of the atypical usages are seen as ‘farfetched’ and thus, rejected. In the case of MadeUpOf/MadeOf, there may be some room for improvement for Atomic²⁰. Unlike the Atomic²⁰ label that successfully joins together ConceptNet’s Has(First/Last)Subevent labels for an improved accuracy, Atomic²⁰’s MadeUpOf union of MadeOf, PartOf, and a subset of HasA, did not seem to have resulted in improved quality. The rest of the Atomic²⁰ cognates see a significantly higher or similar accuracy in comparison to ConceptNet.

### Coverage Assessment

We make a pairwise comparison between the CSKGs to assess their coverage with regards to the commonsense knowledge they contain. For a reliable head-to-head comparison, we map relations and tuples between various KGs.

### Mapping Relations.

Since Atomic²⁰ is built on existing Atomic relations, we primarily need to align relations between Atomic²⁰ and ConceptNet. We carefully align them based on the definitions for the labels as supplied by the two graphs, then the resulting alignment was verified by sampling at random approximately 20 instances per relation.

### Mapping Tuples.

In order to resolve syntactic differences in how the concepts are expressed in each of the KGs (e.g., Atomic’s “PersonX eats breakfast” vs. ConceptNet’s “eat breakfast”), we preprocess each of the head and tail concepts of each tuple in each KG in the following manner: (1) the concept is lowercased and stripped of extra spaces, punctuations, and stopwords; (2) any exact tuple duplicates within each KB removed, and (3) remaining content words are lemmatized according to their POS category. For Atomic and Atomic²⁰, an extra step is added to remove mentions of “PersonX”, “PersonY” and “PersonZ” if occurring at the beginning of a string, and to replace with ‘person’ if they occur elsewhere (e.g., “PersonX greets PersonY”).

### Metrics.

We use two metrics to evaluate the coverage of knowledge graphs. For each pair of CSKGs, we compute precision and recall with respect to a target KG.

---

7Overall inter-rater agreement measured by Fleiss’ $\kappa$ of 0.46
There is very little overlap between A and B; the latter shows the widest coverage: A focuses on social behaviors and C on decision and recall assessment among the CSKGs. A assesses the proportion of tuples in the source KG that are correct according to tuples in the target KG. Coverage recall reflects the proportion of tuples in the target KB that the tuples in the source KB successfully recalled.

### Results

Tables 4 and 5 show a pairwise coverage precision and recall assessment among the CSKGs. A shows the widest coverage: A is able to recall all of ATOMIC (as expected) and just under half of CONCEPTNET. There is very little overlap between ATOMIC and CONCEPTNET, which is unsurprising as all of ATOMIC knowledge is focused on social behaviors. CONCEPTNET does not cover while CONCEPTNET leans on physical commonsense which falls outside ATOMIC's scope. Overall, TRANSMOMCS intersects very little with any of the other three KBs.

### 5 Neural Knowledge Graph Comparison

Language models are powerful tools for representing knowledge, but their ability to serve as generative knowledge bases is limited by the fact they are directly trained to represent the distribution of language. Previous work shows knowledge graphs can help language models better transfer as knowledge engines (Bosselut et al. 2019) by re-training them on examples of structured knowledge. As a result, a new purpose for knowledge graphs is to be useful in helping language models generalize to hypothesizing knowledge tuples.

**Experimental Setup.** To evaluate whether knowledge graphs can help language models effectively transfer to knowledge models, we train different pretrained language models on the knowledge graphs described in Section 4, which we describe below:

- **GPT2** (Radford et al. 2019) is a Transformer (Vaswani et al. 2017) based language model. In our experiments, we use the largest GPT2 model, GPT2-XL, that has 1.5B parameters.

We fine-tune GPT2-XL on each of our CSKGs to predict the tail of a tuple (e.g., wheat) given the head (e.g., bread) and a relation (e.g., MadeUpOf). The hyperparameter settings used for training are described in more detail in Appendix C. Additionally, we use GPT2-XL in a zero-shot setting as a baseline to measure the effect of transfer learning on knowledge graphs. For fair comparison, we convert each relation manually to an English language prompt expecting the tail of each tuple as output generated by the model.

**BAR** (Lewis et al. 2020) is a Bidirectional and Autoregressive Transformer, an adaptation from BERT (Devlin et al. 2019) that is better suited for natural language generation (e.g., translation, summarization). Additional training details are provided in Appendix C.

**GPT-3** (Brown et al. 2020) is an autoregressive language model that has 175B (over 100X more parameters than GPT2-XL) parameters and is trained on a corpus of web text. We use the GPT-3 API to prime the language model to generate the tail for a given prefix – (head, relation) pair. Thus, GPT-3 is evaluated in a few-shot setting. Additional details of our implementation are provided in Appendix C.

**Evaluation Setup.** To assess language-to-knowledge transfer capabilities, we evaluate how language models generalize to new, unseen entities, concepts, or events. We split each knowledge graph into training, validation, and test sets such that the heads of the knowledge tuples do not overlap between these sets. This adversarial split forces the language models to generalize the relationships they learn from training on the knowledge graphs to the entities learned during language pretraining. Also, to avoid overpopulating the validation and test sets with generic heads (e.g., "I", "You", "He", "We", and "They" collectively account for over 2.2M tuple heads in TRANSMOMCS), we enforce that the head of any knowledge tuple in the dev and test sets is involved in at

<table>
<thead>
<tr>
<th>Target KB →</th>
<th>ATOMIC</th>
<th>CN</th>
<th>T-OMCS</th>
<th>ATOMIC&lt;sup&gt;20&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC</td>
<td>-</td>
<td>0.1</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>0.3</td>
<td></td>
<td>5.5</td>
<td>45.6</td>
</tr>
<tr>
<td>TRANSMOMCS</td>
<td>0.0</td>
<td>0.4</td>
<td>4.3</td>
<td>0.3</td>
</tr>
<tr>
<td>ATOMIC&lt;sup&gt;20&lt;/sup&gt;</td>
<td>60.2</td>
<td>9.3</td>
<td>1.4</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 4: Coverage Precision** - Average number of times (in %) a tuple in Source KB is found in Target KB.

<table>
<thead>
<tr>
<th>Source KB ↓</th>
<th>ATOMIC</th>
<th>CN</th>
<th>T-OMCS</th>
<th>ATOMIC&lt;sup&gt;20&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC</td>
<td>-</td>
<td>0.1</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>0.1</td>
<td></td>
<td>0.3</td>
<td>8.9</td>
</tr>
<tr>
<td>TRANSMOMCS</td>
<td>0.0</td>
<td>7.6</td>
<td>-</td>
<td>1.3</td>
</tr>
<tr>
<td>ATOMIC&lt;sup&gt;20&lt;/sup&gt;</td>
<td>100.1&lt;sup&gt;†&lt;/sup&gt;</td>
<td>47.8</td>
<td>4.4</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 5: Coverage Recall** - Average number of times (in %) a tuple in Target KB is found in Source KB. <sup>†</sup>This value is greater than 100 because multiple tuples in ATOMIC<sup>20</sup> can map to the same tuple in ATOMIC.

We use GPT2-XL to convert each KB triple to a sentence.

<table>
<thead>
<tr>
<th>KG</th>
<th>Model</th>
<th>Accept</th>
<th>Reject</th>
<th>No Judgm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC&lt;sup&gt;20&lt;/sup&gt;</td>
<td>GPT2-XL</td>
<td>36.6</td>
<td>62.5</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>GPT-3</td>
<td>73.0</td>
<td>24.6</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>COMET(GPT2-XL)</td>
<td>72.3</td>
<td>26.6</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>COMET(BART)</td>
<td>84.5</td>
<td>13.8</td>
<td>1.7</td>
</tr>
<tr>
<td>ATOMIC</td>
<td>GPT2-XL</td>
<td>39.0</td>
<td>61.2</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>COMET(GPT2-XL)</td>
<td>64.1</td>
<td>34.7</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>COMET(BART)</td>
<td>83.1</td>
<td>15.3</td>
<td>1.6</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>GPT2-XL</td>
<td>50.3</td>
<td>42.1</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>COMET(GPT2-XL)</td>
<td>74.5</td>
<td>19.0</td>
<td>6.4</td>
</tr>
<tr>
<td></td>
<td>COMET(BART)</td>
<td>75.5</td>
<td>17.9</td>
<td>6.6</td>
</tr>
<tr>
<td>TRANSMOMCS</td>
<td>GPT2-XL</td>
<td>28.7</td>
<td>53.5</td>
<td>17.8</td>
</tr>
<tr>
<td></td>
<td>COMET(BART)</td>
<td>26.9</td>
<td>60.9</td>
<td>12.2</td>
</tr>
<tr>
<td></td>
<td>COMET(BART)</td>
<td>23.8</td>
<td>65.9</td>
<td>10.3</td>
</tr>
</tbody>
</table>

**Table 6: Human evaluation of generation accuracy (%). Each model uses greedy decoding to generate the tail of 5K randomly-sampled test prefixes (head, relation) from each knowledge graph. GPT2-XL, GPT-3 and BART have 1.5B, 175B and 440M parameters, respectively.
First, we note the large divide between the zero-shot GPT2-XL model and the two COMET models across the test set of each knowledge graph. For a subset of 5000 generated tuples (Zhang et al. 2020b). For a subset of 5000 generated tuples from the test set of each knowledge graph, we also run the human evaluation described in Section 4.

We score the tuples generated by these knowledge models using common evaluation metrics for text generation: BLEU (Papineni et al. 2002), ROUGE (Lin 2004), CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015), and BERT Score (Papineni et al. 2002), ROUGE (Lin 2004), CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015), and BERT Score.

Using these evaluation metrics, we report the performance of language models trained on language (GPT2-XL), declarative knowledge (ATOMIC and ATOMIC), and COMET models (Table 6). This large gap indicates that language models can benefit from learning facts from commonsense knowledge graphs. They do not have the means to precisely express this knowledge directly. With this framing in mind, the COMET training paradigm proposed by Bosselet et al. (2019) can perhaps be viewed less as a means of learning knowledge from KGs, and more as a method of learning an interface for language models to hypothesize encoded knowledge through language generation. We look forward to future work in this space that attempts to disentangle these two ideas.

Most 500 tuples. Finally, we remove low-quality tuples from TRANSOMCS by imposing a confidence score of $≥ 0.5$.

We score the tuples generated by these knowledge models using common evaluation metrics for text generation: BLEU (Papineni et al. 2002), ROUGE (Lin 2004), CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015), and BERT Score (Zhang et al. 2020b). For a subset of 5000 generated tuples from the test set of each knowledge graph, we also run the same human evaluation described in Section 4.

**Results.** We present our main results in Tables 6 and 7. First, we note the large divide between the zero-shot GPT2-XL model that produces commonsense knowledge without any fine-tuning and the two COMET models across the ATOMIC, ATOMIC, and CONCEPTNET knowledge graphs (Table 6). This large gap indicates that language models can benefit from learning facts from commonsense knowledge graphs. They do not have the means to precisely express this knowledge directly from just pretraining on language. This observation is supported by the gaps between these models in the automatic evaluations (Table 7), as well. Additionally, human evaluation of GPT-3 (Table 6) shows a ~12 point deficit compared to the performance of COMET(BART), in spite of GPT-3 (175B) having over ~430 times more parameters than COMET(BART) (406M). Similarly, we see a large gap in performance across all automated metrics in Table 7. The performance gap indicates that high-quality declarative knowledge is valuable even after the advent of extreme scale language models.

In addition to this main result, two particularly interesting observations emerge. First, we note that the gap between the zero-shot model and COMET is larger on the ATOMIC and ATOMIC knowledge graphs, than on CONCEPTNET, supporting the reflection that ATOMIC supports categories of knowledge that are more difficult to learn from pretraining. Second, the results on the human evaluation show that COMET models trained on TRANSOMCS are not able to generalize knowledge to new entities, implying that language models benefit more from accurate knowledge examples, which TRANSOMCS lacks (cf. §4).

### Table 7: Automated metrics for the quality of the tail generations of the GPT2-XL language model and the knowledge models COMET(GPT2-XL) and COMET(BART). Each approach uses greedy decoding for sampled 5k test prefixes for each KG. The 5k prefixes correspond to the ones for the human eval. Similar results are obtained on the full test sets (cf. Appendix C).

<table>
<thead>
<tr>
<th>Knowledge Graph</th>
<th>Bleu-1</th>
<th>Bleu-2</th>
<th>Bleu-3</th>
<th>Bleu-4</th>
<th>METEOR</th>
<th>ROUGE-L</th>
<th>CIDEr</th>
<th>BERT Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPT2-XL</strong></td>
<td>0.101</td>
<td>0.028</td>
<td>0.010</td>
<td>0.003</td>
<td>0.082</td>
<td>0.098</td>
<td>0.047</td>
<td>0.395</td>
</tr>
<tr>
<td><strong>GPT-3</strong></td>
<td>0.299</td>
<td>0.153</td>
<td>0.081</td>
<td>0.048</td>
<td>0.182</td>
<td>0.255</td>
<td>0.175</td>
<td>0.540</td>
</tr>
<tr>
<td><strong>COMET(GPT2-XL)</strong></td>
<td>0.407</td>
<td>0.248</td>
<td>0.171</td>
<td>0.124</td>
<td>0.292</td>
<td>0.485</td>
<td>0.653</td>
<td>0.638</td>
</tr>
<tr>
<td><strong>COMET(BART)</strong></td>
<td>0.469</td>
<td>0.286</td>
<td>0.189</td>
<td>0.130</td>
<td>0.330</td>
<td>0.495</td>
<td>0.658</td>
<td>0.639</td>
</tr>
<tr>
<td><strong>ATOM</strong></td>
<td>0.083</td>
<td>0.029</td>
<td>0.011</td>
<td>0.005</td>
<td>0.081</td>
<td>0.087</td>
<td>0.045</td>
<td>0.386</td>
</tr>
<tr>
<td><strong>COMET(GPT2-XL)</strong></td>
<td>0.419</td>
<td>0.296</td>
<td>0.228</td>
<td>0.189</td>
<td>0.292</td>
<td>0.517</td>
<td>0.733</td>
<td>0.634</td>
</tr>
<tr>
<td><strong>COMET(BART)</strong></td>
<td>0.515</td>
<td>0.324</td>
<td>0.220</td>
<td>0.159</td>
<td>0.347</td>
<td>0.546</td>
<td>0.740</td>
<td>0.646</td>
</tr>
<tr>
<td><strong>CONCEPTNET</strong></td>
<td>0.044</td>
<td>0.012</td>
<td>0.004</td>
<td>0.002</td>
<td>0.064</td>
<td>0.057</td>
<td>0.050</td>
<td>0.389</td>
</tr>
<tr>
<td><strong>COMET(GPT2-XL)</strong></td>
<td>0.155</td>
<td>0.119</td>
<td>0.095</td>
<td>0.078</td>
<td>0.134</td>
<td>0.193</td>
<td>0.425</td>
<td>0.552</td>
</tr>
<tr>
<td><strong>COMET(BART)</strong></td>
<td>0.172</td>
<td>0.111</td>
<td>0.072</td>
<td>0.049</td>
<td>0.130</td>
<td>0.184</td>
<td>0.368</td>
<td>0.535</td>
</tr>
<tr>
<td><strong>TRANSOMCS</strong></td>
<td>0.028</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.093</td>
<td>0.053</td>
<td>0.013</td>
<td>0.351</td>
</tr>
<tr>
<td><strong>COMET(GPT2-XL)</strong></td>
<td>0.301</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.180</td>
<td>0.302</td>
<td>0.254</td>
<td>0.677</td>
</tr>
<tr>
<td><strong>COMET(BART)</strong></td>
<td>0.351</td>
<td>0.170</td>
<td>0.003</td>
<td>0.000</td>
<td>0.198</td>
<td>0.352</td>
<td>0.297</td>
<td>0.678</td>
</tr>
</tbody>
</table>

**6 Discussion**

Do pretrained language models already encode commonsense knowledge? Our conclusions on this subject are mixed and hinge on the ambiguous meaning of what it means to encode knowledge. Despite the conclusions of prior work (Petroni et al. 2019; Roberts, Raffel, and Shazeer 2020; Tamborrino et al. 2020), our results in Table 6 are clear that language models fail to express large varieties of knowledge when prompted for it in a zero-shot manner. When converted to COMET models by training on a knowledge graph, their performance at hypothesizing knowledge tuples skyrockets – 47.9% absolute difference between COMET(BART) and GPT2-XL on ATOMIC.

However, the evaluation tuples are adversarially selected to not include head entities that were in the training set. The model must generalize its learned representations of relations to entities it has not observed these relationships for during fine-tuning, meaning the representation of these entities is solely formulated from learning language. As a result, language models may still encode this knowledge in their parameters, even if they are not capable of expressing it directly. With this framing in mind, the COMET training paradigm proposed by Bosselet et al. (2019) can perhaps be viewed less as a means of learning knowledge from KGs, and more as a method of learning an interface for language models to hypothesize encoded knowledge through language generation. We look forward to future work in this space that attempts to disentangle these two ideas.

What considerations should be made when designing commonsense knowledge resources? Based on our results in Section 5, we outline desiderata for the design and development of future commonsense knowledge graphs. Because certain types of knowledge are already encoded and expressible by pretrained language models, CSKG designers should...
focus on collecting examples and categories of knowledge that are less likely to be known by language models. For example, of the 378 test tuples evaluated by the GPT2-XL zero-shot model that contained the HinderedBy relation, only 1.3% were deemed plausible by human raters – jumping to 85% plausibility for COMET(BART) – pointing to an advantage in constructing ATOMIC$^{20}$ with this relationship in mind (see Appendix C for per-relation accuracy).

Second, commonsense knowledge resources should be designed with the goal of accuracy and relationship coverage. Because language models exhibit powerful adaptation (Brown et al. 2020), they can generalize many commonsense relationships as long they have examples on which to train. Consequently, we should construct commonsense resources that encapsulate larger numbers of relations so the knowledge in pretrained language models can be grounded to a variety of relationships. However, language models also benefit from learning from precise examples. Being able to train on a large collection of examples from TRANSOMCS (see Appendix C) did not allow COMET models to generalize to unseen entities as these examples were not of sufficient quality (See Table 2). Resources should be carefully validated for the quality of their facts, an example set by Speer, Chin, and Havasi (2017) and Sap et al. (2019).

7 Conclusion
In this work, we formalize a use for commonsense knowledge graphs as transfer learning tools for pretrained language models. With this new purpose, we hypothesize that commonsense knowledge graphs should be designed to contain knowledge that is not already expressible by language models without difficulty (e.g., not taxonomic and lexical knowledge). Consequently, we propose ATOMIC$^{20}$, a novel commonsense knowledge graph containing tuples whose relations are specifically selected to be challenging for pretrained language models to express. Our empirical studies demonstrate that ATOMIC$^{20}$ contains high-accuracy knowledge tuples across multiple novel relations not found in existing CSKGs or expressible by LMs. Furthermore, we show that ATOMIC$^{20}$ can be effectively used as a training set for adapting language models as knowledge models to generate high quality tuples on-demand.

References
Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Çelikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In ACL.
Gordon, J.; and Van Durme, B. 2013. Reporting bias and knowledge acquisition. In AKBC ’13. ACM.
Lascarides, A.; and Asher, N. 1991. Discourse relations and defeasible knowledge. In ACL.
Li, X.; Taheri, A.; Tu, L.; and Gimpel, K. 2016. Commonsense Knowledge Base Completion. In ACL.


A ATOMIC 2020 Details

ATOMIC 20 Relations

In this section we detail the relations in ATOMIC 20. Figure 3 shows the hierarchical breakdown of the ATOMIC 20 relation labels. While there is no internal structure directly encoded for ATOMIC 20 relations, they fall into three natural categories based on their meaning: physical-entity, social-interaction and event-centered commonsense.

Physical-Entity Commonsense. Physical-entity commonsense deals with inferential knowledge about common entities and objects. Physical commonsense such as these is crucial for interacting with the world: allowing us to distinguish the dangerous (e.g., “fire can be painful”) from the harmless (e.g., “teddy bears are comforting”), manipulate objects for our use (e.g., “helmets protect head”), and solve problems (e.g., “how do I open this door?”). We identify seven relations under this category.

- **ObjectUse** describes everyday affordances or uses of objects, and includes both typical and atypical uses. For example, “popcorn bucket” can typically be used to “hold popcorn” but it could also serve “as a hat” in atypical situations.
- **MadeUpOf** and **HasProperty** are two property relations, denote the relationship between an entity and its composition or characteristics. MadeUpOf describes a part, portion or makeup of an entity. For example, “cake” can be MadeUpOf “eggs” (composition/ingredient) or “icing” (part/portion). Similarly, HasProperty usually describes entities’ general characteristics such as “rose” is “red,” subjective attributes such as “thirst” is “uncomfortable.” In certain cases, the relation can also map to descriptors that speak to the substance or value of items such as “meat” has property of being “stored in the freezer” or “bike” is “powered by person’s legs.”
- **AtLocation** is a spatial relation that describes the location in/on/at which an entity is likely to be found (e.g., “gambler” can be found in “casino,” “wrench” can be found in “garage”).
- **CapableOf** is designed to describe abilities and capabilities of everyday living entities (e.g., humans, animals, insects) and natural entities that can exert a force (e.g., sun, storms). CapableOf includes general capabilities such as a “human” is capable of “thinking and reasoning” or drinking coffee.” It also includes specialized capabilities such as a “surgeon” is capable of “operating on a patient.”
- **Desires** and **NotDesires** are relations that deal with desires of sentient entities; e.g., “doctors” likely desire to “cure patient” but do not desire “malpractice suit.”

Social-Interaction Commonsense. Social-interaction relations comment on socially-triggered states and behaviors. Social commonsense is useful for gauging people’s intentions and purpose, and predicting situationally-relevant human reactions and behaviors. Following the definitions for ATOMIC relations (Sap et al. 2019), we identify a total of nine relations within this category.

- Three mental state relations address the emotional or cognitive states of the participants in a given event. **xIntent** defines the likely intent or desire of an agent (X) behind the execution of an event. Given the head “X gives Y gifts,” an xIntent might be that X wanted “to be thoughtful.” Relations **xReact** and **oReact** define the emotional reactions on the part of X or other participants in an event. As a result of gift giving, X might feel “good about [one]self” and others (in this case, Y) might feel “appreciated.”
- Five behavioral relations address the socially relevant responses to an event. **xNeed** describes a precondition for X achieving the event. For example, in order for X to give Y gifts, X must first “buy the presents.” **xWant** and **oWant** are postcondition desires on the part of X and others, respectively. As a result of X giving Y gifts, X may also desire “to hug [Y]” and Y may want to “open the gift.” **xEffect** and **oEffect** are social actions that may occur after the event: X may “get hugged” and Y may “blush” in response.
- The last relation **xAttr** describes X’s persona or attribute as perceived by others given an event. In the gift giving example, X may be seen as “generous” or “giving.” In contrast, in an event such as “X steals a car,” X may be perceived as “evil.”

Event-Centered Commonsense. While social-interaction commonsense gauges human behaviors and mental states given an event, the event-centered commonsense provides intuitions about how common events are related to one another. Commonsense about event interaction is useful for understanding likely causes and effects of events in the world. This knowledge allows humans to strategize and explore the best solutions for their objectives, make contingency plans, and revise goals when circumstances deviate from expectations. There are seven relations that fall under this category.

- We group three relations under force dynamics. This group conceptualizes dynamic interactions between events with regards to exerted causal forces and impelled actions. **Causes** specifically captures the causal relation between two events or entities – e.g. an “accident” can cause “injury.” Causes does have some overlap with behavioral relations such as **xEffect** in that they are postconditions of an event, but the postcondition in **Causes** is not socially triggered and can exist outside human control (e.g., “bad weather” causes “power
outages”). **HinderedBy** introduces hindrances that obstruct the natural path to the achievement of a goal. For example, the event “X adopts a cat” can be obstructed if “X is allergic to cats.” **xReason** provides a post-fact explanation of the cause of an event (e.g., why one has to “walk” could be explained by “car has broken down”), which is related to, but distinct from, **xIntent’s** intentions (i.e., “X walks” because X wanted to “go home”).

- **Three relations provide reasoning about event scripts or sequences.** **isAfter** and **isBefore** introduce events that can precede or follow an event, respectively. For example, “X is in a hurry to get to work” can happen after “X wakes up 15 minutes late” and before “X drives too fast.” These relations are distinguished from behavioral relations **xNeed** (pre-condition) and **xEffect** (post-condition) in that **isAfter** and **isBefore** are temporally situated without specific regard to the need or reaction of the person X. For example, “X pushes X’s luck” can happen before “X gets a broken nose” but getting a broken nose is not an action X intentionally may take after pushing one’s luck. Relation **HasSubEvent** provides the internal structure of an event, each tail denoting a step within the larger head event.

- The last relation in the event-centered category, **isFilledBy**, provides a filler phrase for an event with a blank that is sensical and commonly acceptable for the event. For example, the blank in an event such as “X catches ___ in the act” can be commonly filled by entities such as a “cheater,” a “burglar,” or a “mouse.”

### ATOMIC$^{20}$ Tuples

In this section, we detail the population of the ATOMIC$^{20}$ tuples (see Table 1 for counts per relation).

#### Social-Interaction Tuples

For social-interaction relations, we incorporated 877K tuples from ATOMIC, and crowdsourced an additional 34K tuples using the same approach as Sap et al. (2019). The rest of this section will refer to the head events in the social-interaction tuples as **base events**.

#### Crowdsourced Tuples

Tuples for relations **ObjectUse**, **HinderedBy**, **isFilledBy**, **isBefore** and **isAfter** were crowdsourced via Amazon Mechanical Turk. We paid an average of $15 an hour for our crowdsourcing efforts. We release all crowdsourcing templates as part of our codebase.\[^{10}\]

- For the collection of **HinderedBy**, we crowdsourced over 100K event-hindrance tuples by prompting the workers with base events from ATOMIC and eliciting reasons why one may not be able to achieve the event. In order to make the prompt events more worker-friendly, we processed the events as a desire (e.g., “X adopts a cat” → “X wants to adopt a cat”).\[^{11}\] We specifically elicited personal causes (e.g., “X is allergic to cats”), situational causes (e.g., “there are no pet stores nearby”), and social causes (e.g., “X’s landlord disallows pets”).

- 33K **isFilledBy** tuples were collected by presenting workers with base events. The workers were asked to provide two (up to four) common objects or entities that will make sense in the sentence.

- 46K tuples for **isBefore** and **isAfter** were collected together as sequences of events. Given a base event, the workers were asked to write a short 3-sentence story by

\[^{10}\]http://anonymous

\[^{11}\]To achieve this, we removed modal verbs, lemmatized the head verb of the sentence, and inserted a ‘want to’ phrase before the verb.
Table 8: CONCEPTNet relations mapped to ATOMIC20 relations. For labels mapping to multiple ATOMIC20 relations, the one that received the majority mapping is bolded.

<table>
<thead>
<tr>
<th>CONCEPTNet</th>
<th>ATOMIC20</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtLocation</td>
<td>AtLocation</td>
</tr>
<tr>
<td>CapableOf</td>
<td>CapableOf</td>
</tr>
<tr>
<td>Causes</td>
<td>Causes, xEffect</td>
</tr>
<tr>
<td>CausesDesire</td>
<td>xWant</td>
</tr>
<tr>
<td>Desires</td>
<td>Desires</td>
</tr>
<tr>
<td>MadeOf</td>
<td>MadeUpOf</td>
</tr>
<tr>
<td>HasProperty</td>
<td>HasProperty</td>
</tr>
<tr>
<td>HasA</td>
<td>MadeUpOf, HasProperty</td>
</tr>
<tr>
<td>HasPrerequisite</td>
<td>xNeed</td>
</tr>
<tr>
<td>HasSubevent</td>
<td>HasSubEvent</td>
</tr>
<tr>
<td>HasFirstSubevent</td>
<td>HasSubEvent</td>
</tr>
<tr>
<td>HasLastSubevent</td>
<td>HasSubEvent</td>
</tr>
<tr>
<td>NotDesires</td>
<td>NotDesires</td>
</tr>
<tr>
<td>MotivatedByGoal</td>
<td>xIntent, xReason</td>
</tr>
<tr>
<td>PartOf</td>
<td>MadeUpOf</td>
</tr>
<tr>
<td>UsedFor</td>
<td>ObjectUse</td>
</tr>
<tr>
<td>ReceivesAction</td>
<td>MadeUpOf, AtLocation, Causes, ObjectUse</td>
</tr>
</tbody>
</table>

Table 9: Human readable templates for each relation used for crowdsourced human evaluations.

<table>
<thead>
<tr>
<th>Relations</th>
<th>Human Readable Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtLocation</td>
<td>located or found at/in/on</td>
</tr>
<tr>
<td>CapableOf</td>
<td>is/are capable of</td>
</tr>
<tr>
<td>Causes</td>
<td>causes</td>
</tr>
<tr>
<td>CausesDesire</td>
<td>makes someone want</td>
</tr>
<tr>
<td>CreatedBy</td>
<td>is created by</td>
</tr>
<tr>
<td>Desires</td>
<td>desires</td>
</tr>
<tr>
<td>HasA</td>
<td>has, possesses or contains</td>
</tr>
<tr>
<td>HasFirstSubevent</td>
<td>begins with the event/action</td>
</tr>
<tr>
<td>HasLastSubevent</td>
<td>ends with the event/action</td>
</tr>
<tr>
<td>HasPrerequisite</td>
<td>to do this, one requires</td>
</tr>
<tr>
<td>HasProperty</td>
<td>can be characterized by being/having</td>
</tr>
<tr>
<td>HasSubEvent</td>
<td>includes the event/action</td>
</tr>
<tr>
<td>HinderedBy</td>
<td>can be hindered by</td>
</tr>
<tr>
<td>InstanceOf</td>
<td>is an example/instance of</td>
</tr>
<tr>
<td>isAfter</td>
<td>happens after</td>
</tr>
<tr>
<td>isBefore</td>
<td>happens before</td>
</tr>
<tr>
<td>isFilledBy</td>
<td>blank can be filled by</td>
</tr>
<tr>
<td>MadeOf</td>
<td>is made of</td>
</tr>
<tr>
<td>MadeUpOf</td>
<td>made (up) of</td>
</tr>
<tr>
<td>MotivatedByGoal</td>
<td>is a step towards accomplishing the goal</td>
</tr>
<tr>
<td>NotDesires</td>
<td>do(es) NOT desire</td>
</tr>
<tr>
<td>ObjectUse, UsedFor</td>
<td>used for</td>
</tr>
<tr>
<td>oEffect</td>
<td>as a result, Y or others will</td>
</tr>
<tr>
<td>oReact</td>
<td>as a result, Y or others feels</td>
</tr>
<tr>
<td>oWant</td>
<td>as a result, Y or others want</td>
</tr>
<tr>
<td>PartOf</td>
<td>is a part of</td>
</tr>
<tr>
<td>ReceivesAction</td>
<td>can receive or be affected by the action</td>
</tr>
<tr>
<td>xAttr</td>
<td>X is seen as</td>
</tr>
<tr>
<td>xEffect</td>
<td>as a result, PersonX will</td>
</tr>
<tr>
<td>xIntent</td>
<td>because PersonX wanted</td>
</tr>
<tr>
<td>xNeed</td>
<td>but before, PersonX needed</td>
</tr>
<tr>
<td>xReact</td>
<td>as a result, PersonX feels</td>
</tr>
<tr>
<td>xReason</td>
<td>because</td>
</tr>
<tr>
<td>xWant</td>
<td>as a result, PersonX wants</td>
</tr>
</tbody>
</table>

Integration of CONCEPTNet Tuples. The tuples for the remaining relations are populated through the integration of the commonsense portion of CONCEPTNet. As discussed in the main text, a select subset of CONCEPTNet(v5.7) tuples (172K) were integrated into ATOMIC20.

The primary challenge in integrating CONCEPTNet tuples into ATOMIC20 was in identifying knowledge that is most likely to reflect commonsense information. CONCEPTNet(v5.7) contains tuples built on not only concept relationships directly sourced from human informants, but also on information pulled from other lexical sources such as WordNet (Miller 1995) and DBpedia (Auer et al. 2007), which automatically extracts knowledge from Wikipedia articles (Speer, Chin, and Havasi 2017). As a result, even those relations that are designed to primarily represent commonsense knowledge (i.e., the OMCS relations) include among the mix, tuples that reflect factual or lexical co-occurrence knowledge. These examples deviate from the type of knowledge we would ideally consider as "commonsense," i.e., qualitative experiential knowledge gained through subjective observation of and interaction with the world. Relations such as InstanceOf ("is instance/example of") stands as a case in point (e.g., "tortilla" is an example of "flatbread" or "toffee" is an example of "candy"). While included within the OMCS relations, the encoded information can be hard to distinguish from the more accepted taxonomic relations such as IsA ("is a kind/type of"). Relationships found in relations such as RelatedTo and DistinctFrom are too underspecified with regards to the meaning they represent, and for other relations such as LocatedNear, and negative forms such as NotCapableOf or NotHasProperty, the relationships amount to general lexical relationships.

The process of CONCEPTNet(v5.7) knowledge selection (described in §3) was judiciously guided by three competing priorities: when possible, we prioritized (1) qualitative commonsense over factual knowledge, (2) general knowledge over highly specific knowledge (e.g., personal names), and (3) meanings that are specific enough to be

---

12In fact, CONCEPTNet(v5.7) recognizes the similarities between IsA and InstanceOf and has accordingly deprecated InstanceOf in favor of IsA. Nevertheless, InstanceOf is still found in CONCEPTNet(v5.7).
meaningfully categorized. Since the ideal route of verifying imported data via crowdsourcing can be resource-intensive, we opted for an approach whereby relations were first selected based on the data they represent; then tuples were pruned based on heuristics that leverage lexical and syntactic information of the concepts. As mentioned in the main text, 10% of the data selected for integration was validated by crowdsowers, yielding a greater than 93% acceptance rate. Three relations, namely HasProperty, CapableOf, and MotivatedByGoal, were sent for instance-by-instance crowdsourcing for the purpose of debiasing human-related descriptions, and subdividing semantically distinct elements within the category (e.g., MotivatedByGoal mapped to xIntent and xReason). The resulting CONCEPTNET-to-ATOMIC relation mapping details are shown in Table 8.

B Symbolic Knowledge Graph Details

Human Evaluation

Qualifying Crowdsource Workers. To ensure high-quality annotations, we qualified a pool of 173 workers through a paid qualification task that tested their ability to follow directions and provide reasonable answers to the qualification test. The qualification test contained 6 manually selected tuples from ATOMIC and CONCEPTNET, including both easy and tricky relations to annotate. A worker was qualified if they provided 100% acceptable answers. Workers providing 5 of 6 correct answers were also accepted only when they provided a reasonable written substantiation for their incorrect choice. Workers were paid an average of $15 per hour for their evaluations.

Human Readable Relation Templates. Since the KB relation labels are rather telegraphic on their own, we used human-readable language forms (based ATOMIC and CONCEPTNET definitions) for prompt display in crowdsourced evaluations. The complete list is available in Table 9.

KB Accuracy & Coverage

In Table 2, what type of tuples generally end up with no judgment? Tuples receiving no judgment fall into three general categories: (1) either the head or the tail concept is too specialized for the workers to judge without consulting reference (e.g., “klebsiella” is part of “bacteria,” “drug cocktail” made of “nucleoside reverse transcriptase inhibitor”); (2) concepts refer to highly specific entities or referents (e.g., “singh” capable of “bring key,” “falkland island islas malvinas” part of “argentina”); and (3) Reject candidates that workers have decided to hedge on (e.g., “dandelion” used for “love,” “democrat” desires “matter”). Such tuples are mostly found in TransOMCS, as evidenced by the high fraction of tuples that received No Judgment at less than half of ATOMIC’s Accept rate (see 2).

Does the accuracy ratings breakdown for each KB provide further insights? A closer look at the raw accuracy ratings shows an interesting emergent rating pattern across KBs (Table 2). For all KBs with the exception of TransOMCS, we observe that the majority of social-interaction Accept originate from the sometimes/likely rating. However, such preference is not seen in the physical-entity tuples, which show a slightly higher tendency for the always/often rating. For event-centered tuples, ATOMIC favors the sometimes/likely, while CONCEPTNET does not. TRANSOMCS shows highest ratings for the sometimes/likely and invalid ratings, and the patterns are invariant across the board.

One additional point to mention is that ATOMIC’s social-interaction and event-centered tuples proportionally contain more of the human-crowdsourced commonsense knowledge than the physical-entity category, which, with the sole exception of ObjectUse, includes tuples integrated from CONCEPTNET graph. The observation that much of the knowledge in ATOMIC is sometimes or likely true, reflects our intentional efforts to deprioritize factual information over qualitative commonsense knowledge. More importantly, it shows that most of the knowledge within the ATOMIC graph can be, under the right circumstances, defeasible. That is, one can pose a likely hypothesis that a hindrance to “X writes stories” is that “X can’t read;” however, such a hypothesis can be defeated if we also know that X has written stories before. We find that such context-dependent ambiguities are of more compelling interest to us, as certainities may be better covered by language models.

C Neural Knowledge Graph Details

Dataset Split. Table 10 reports the number of tuples for each three-way split (train/dev/test) of each knowledge graph. The ATOMIC split preserves the splits from ATOMIC and CONCEPTNET: any tuple in ATOMIC that appears in the train (resp. dev, test) set of ATOMIC or CONCEPTNET belongs to the train (resp. dev, test) set of ATOMIC. Overall, ATOMIC provides over 50% more tuples than the initial version ATOMIC.

Table 10: Number of tuples per KB and per split.

<table>
<thead>
<tr>
<th>Knowledge Graph</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOMIC</td>
<td>1,076,880</td>
<td>102,024</td>
<td>152,209</td>
<td>1,331,113</td>
</tr>
<tr>
<td>ATOMIC</td>
<td>709,993</td>
<td>79,599</td>
<td>87,480</td>
<td>877,072</td>
</tr>
<tr>
<td>CONCEPTNET</td>
<td>264,791</td>
<td>5,000</td>
<td>30,209</td>
<td>300,000</td>
</tr>
<tr>
<td>TRANSOMCS</td>
<td>5,424,478</td>
<td>10,243</td>
<td>100,033</td>
<td>5,534,754</td>
</tr>
</tbody>
</table>

Details about GPT2-XL Training. GPT2-XL (Radford et al. 2019) is a transformer language model trained on 8 million webpages (~40G of text data). We finetune the language model on each commonsense knowledge graph by converting a tuple into a formatted text input – e.g. `<head> <relation> [GEN] <tail> [SEP]` where [GEN] and [SEP] are special delimiter tokens that indicate the start and end of the tail of a given relation for a given head entity / event. At inference time, the head and relation of a tuple are given as input and the model’s generation following the [GEN] token is recorded as its prediction of the tail entity. We finetuned GPT2-XL on each CSKG for one epoch, using a batch size of 32 and a learning rate of 5e-5 on an Nvidia RTX-8000 GPU. The final trained models for each CSKG will be publicly released as part of our code release. In Figure 5, we include a few examples of code release.
Figure 4: Percentage distribution of raw accuracy ratings broken down by KB (i.e., breakdown of Table 2). From left to right are the ratings for social-interaction tuples, physical-entity tuples, and event-centered tuples. We use the CONCEPTNET-to-ATOMIC relation mappings (shown in Table 8) to categorize CONCEPTNET and TRANSOMCS relations into the three categories. For multiple mappings, we map the CONCEPTNET/TRANSOMCS labels to the majority mapped label (in bold in Table 8). Note that the latter two figures do not include ATOMIC as the KB only includes social-interaction relations.

Table 11: Automated metrics for the quality of the tail generations for the knowledge models COMET(GPT2-XL) and COMET(BART). Each approach uses greedy decoding for all test prefixes for each KG. Similar results were obtained on the 5K sampled prefixes that were randomly selected for the human evaluation (see Table 7).

<table>
<thead>
<tr>
<th>KB</th>
<th>COMET(GPT2-XL)</th>
<th>COMET(BART)</th>
<th>COMET(GPT2-XL)</th>
<th>COMET(BART)</th>
<th>COMET(GPT2-XL)</th>
<th>COMET(BART)</th>
<th>COMET(GPT2-XL)</th>
<th>COMET(BART)</th>
<th>COMET(GPT2-XL)</th>
<th>COMET(BART)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bleu-1</td>
<td>0.401</td>
<td>0.462</td>
<td>0.429</td>
<td>0.521</td>
<td>0.152</td>
<td>0.169</td>
<td>0.298</td>
<td>0.351</td>
<td>0.288</td>
<td>0.302</td>
</tr>
<tr>
<td>Bleu-2</td>
<td>0.247</td>
<td>0.280</td>
<td>0.300</td>
<td>0.330</td>
<td>0.115</td>
<td>0.108</td>
<td>0.000</td>
<td>0.216</td>
<td>0.168</td>
<td>0.092</td>
</tr>
<tr>
<td>Bleu-3</td>
<td>0.168</td>
<td>0.182</td>
<td>0.225</td>
<td>0.225</td>
<td>0.092</td>
<td>0.069</td>
<td>0.000</td>
<td>0.004</td>
<td>0.123</td>
<td>0.164</td>
</tr>
<tr>
<td>Bleu-4</td>
<td>0.123</td>
<td>0.124</td>
<td>0.187</td>
<td>0.164</td>
<td>0.080</td>
<td>0.046</td>
<td>0.000</td>
<td>0.000</td>
<td>0.288</td>
<td>0.179</td>
</tr>
<tr>
<td>METEOR</td>
<td>0.288</td>
<td>0.325</td>
<td>0.297</td>
<td>0.351</td>
<td>0.131</td>
<td>0.127</td>
<td>0.179</td>
<td>0.201</td>
<td>0.473</td>
<td>0.527</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>0.473</td>
<td>0.486</td>
<td>0.527</td>
<td>0.552</td>
<td>0.193</td>
<td>0.180</td>
<td>0.300</td>
<td>0.352</td>
<td>0.620</td>
<td>0.754</td>
</tr>
<tr>
<td>CIDEr</td>
<td>0.620</td>
<td>0.632</td>
<td>0.754</td>
<td>0.766</td>
<td>0.421</td>
<td>0.350</td>
<td>0.249</td>
<td>0.298</td>
<td>0.632</td>
<td>0.636</td>
</tr>
<tr>
<td>BERT Score</td>
<td>0.632</td>
<td>0.650</td>
<td>0.638</td>
<td>0.650</td>
<td>0.552</td>
<td>0.532</td>
<td>0.525</td>
<td>0.681</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Details about BART Training. BART (Lewis et al. 2020) is a denoising sequence-to-sequence pretrained language model. Similar to previous transformer-based language models (Devlin et al. 2019), BART’s pretraining objective is to recover its input, which is corrupted through various strategies such as token and span masking, and sentence permutation. For pretraining, BART uses a 160GB free-text dataset drawn from news, books, stories, and web texts. We used the BART-large version of the model, which has 24 layers, 1024-dimensional hidden states, 16 attention heads in its self-attention layers, and 406M total parameters. For hyper-parameter search, we fine-tuned BART on each commonsense knowledge graph for one epoch with batch sizes {64, 32, 16}, learning rates {1e-3, 1e-5, 1e-7}, and three random seeds.

Details about GPT-3 Evaluation. We evaluate GPT-3 (Brown et al. 2020) using OpenAI’s language completion API. Similar to zero-shot evaluation on GPT2-XL, we use templates to evaluate the ability of the language model to generate a tail given the head and relation. We use the same templates as GPT2-XL. For priming examples, we prime each relation with 5 examples of heads and tails per relation, randomly selected from relations in the training set. We ran 3 random seeds to select priming examples to avoid spelling mistakes and other fragments from data collection. We ran with temperature 0.4.

Additional Automated Evaluation. In order to have a direct comparison between automated and human evaluations, we report in Section 5 the automated metrics on the same test subsets that were used for human evaluation. For completeness, in this section, we provide the automated evaluation results on the full test sets (Table 11). These results confirm the findings of Section 5.
<table>
<thead>
<tr>
<th>Head</th>
<th>Relation</th>
<th>Human</th>
<th>COMET (GPT2 XL)</th>
<th>COMET (BART)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PersonX accepts PersonY’s apology</td>
<td>xIntent (Intent of PersonX)</td>
<td>To feel peaceful with PersonY</td>
<td>To be kind</td>
<td>To be forgiving</td>
</tr>
<tr>
<td>PersonX accepts PersonY’s apology</td>
<td>HinderedBy</td>
<td>PersonY has not apologized</td>
<td>PersonX is stubborn</td>
<td>PersonX is too angry</td>
</tr>
<tr>
<td>PersonX accepts PersonY’s apology</td>
<td>oReact (Reaction of PersonY)</td>
<td>Happy</td>
<td>Relieved</td>
<td>Happy</td>
</tr>
<tr>
<td>PersonX does yard work</td>
<td>isAfter</td>
<td>PersonX is off today</td>
<td>PersonX gets home from work</td>
<td>PersonX gets a job as a gardener</td>
</tr>
<tr>
<td>PersonX does yard work</td>
<td>isBefore</td>
<td>PersonX is proud of being productive</td>
<td>PersonX gets paid</td>
<td>PersonX takes a shower</td>
</tr>
<tr>
<td>PersonX abuses PersonX’s power</td>
<td>xAttr (attribute of PersonX)</td>
<td>Out of line</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>PersonX abuses PersonX’s power</td>
<td>xWant (Want of personX)</td>
<td>Harass employees</td>
<td>To get their way</td>
<td>To be in control</td>
</tr>
<tr>
<td>Common sense</td>
<td>ObjectUse</td>
<td>Prevent mistakes</td>
<td>Understand the situation</td>
<td>Make a good decision</td>
</tr>
<tr>
<td>Move car</td>
<td>HasSubEvent</td>
<td>Open car door</td>
<td>Turn steering wheel</td>
<td>Get out of car</td>
</tr>
<tr>
<td>PersonX loses PersonY’s wallet</td>
<td>xWant (Want of PersonX)</td>
<td>To get the wallet</td>
<td>To apologize</td>
<td>to find the wallet</td>
</tr>
<tr>
<td>PersonX breaks the news to PersonY</td>
<td>oWant (Want of PersonY)</td>
<td>to cry</td>
<td>to cry</td>
<td>to cry</td>
</tr>
<tr>
<td>family photograph</td>
<td>AtLocation</td>
<td>relatives house</td>
<td>Photograph book</td>
<td>Family room</td>
</tr>
<tr>
<td>mouse</td>
<td>HasProperty</td>
<td>Whiskers</td>
<td>Four legs</td>
<td>Long tail</td>
</tr>
<tr>
<td>bird lover</td>
<td>CapableOf</td>
<td>listen to mockingbird sing</td>
<td>watch birds fly</td>
<td>Look at birds</td>
</tr>
</tbody>
</table>

Figure 5: Example generations of models on relations from ATOMIC<sup>20</sup>. Red, purple and green rows represent social-interaction commonsense, event-centered commonsense, and physical-entity commonsense, respectively.
D Additional Reproducibility Items

All experiments were conducted on a cluster with 8 GPUs of type NVIDIA Quadro RTX 8000 with 48 GB of GDDR6 memory each. To allow replication of results, whenever possible, a default, fixed value was assigned to the random seed that initializes the pseudo-random number generator, as specified in the source code. The details of the experimentation of the models (i.e. GPT2-XL and BART), including their hyper-parameter settings, are described in Appendix C. All the data as well as the source code required for conducting experiments will be made publicly available upon publication.
Figure 6: A snapshot of commonsense knowledge relationships in ATOMIC. Gray nodes represent events. Red, purple and green nodes represent social-interaction commonsense, event-centered commonsense, and physical-entity commonsense, respectively. Rest of the colors represent intersection of the categories.