I2D2: Inductive Knowledge Distillation with NeuroLogic and Self-Imitation

Chandra Bhagavatula*, Jena D. Hwang*, Doug Downey¶*, Ronan Le Bras*,
Ximing Lu§*, Lianhui Qin§, Keisuke Sakaguchi‡, Swabha Swayamdipta†, Peter West§*,
Yejin Choi§*
*Allen Institute for AI, †University of Southern California,
‡Tohoku University, ¶Northwestern University, §University of Washington
i2d2.allen.ai

Abstract

Commonsense capabilities of pre-trained language models dramatically improve with scale, leading many to believe that scale is the only winning recipe. But is it? Here, we investigate an alternative that *a priori* seems impossible: can smaller language models (e.g., GPT-2) win over models that are orders of magnitude larger and better (e.g., GPT-3), if powered with novel commonsense distillation algorithms? The key intellectual challenge is to design a learning algorithm that achieves a competitive level of commonsense acquisition, without relying on the benefits of scale. In particular, we study *generative* models of commonsense knowledge, focusing on the task of generating *generics*, statements of commonsense facts about everyday concepts, e.g., birds can fly.

We introduce I2D2, a novel commonsense distillation framework that loosely follows West et al. (2022)'s Symbolic Knowledge Distillation but breaks the dependence on the extreme-scale teacher model with two innovations: (1) the novel adaptation of **NeuroLogic** Decoding (Lu et al., 2021) to enhance the generation quality of the weak, off-the-shelf language models, and (2) **self-imitation learning** to iteratively learn from the model's own enhanced commonsense acquisition capabilities. Empirical results suggest that scale is not the only way, as novel algorithms can be a promising alternative. Moreover, our study leads to a new corpus of generics, `Gen-A-tomic`, that is the largest and highest-quality available to date.

1 Introduction

Language models (LMs) become better with scale. However, even the largest LMs continue to fail in unexpected ways due to their lack of commonsense (Brachman and Levesque, 2021). *Knowledge models* – custom LMs trained to generate knowledge—provide on-demand access to task-specific knowledge to address this gap (Bosselut et al., 2019).

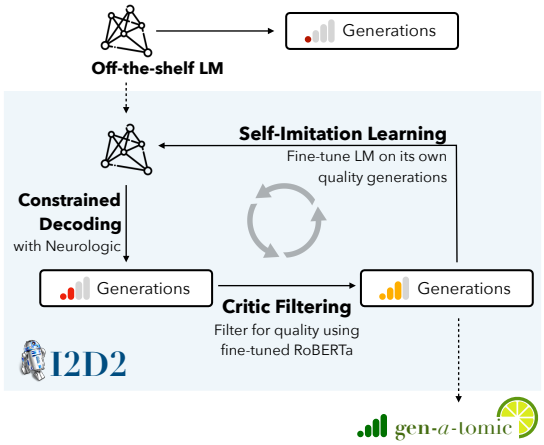

Figure 1: I2D2 radically improves over off-the-shelf generation from GPT-2 XL using constrained decoding and self-imitation.

Today, the best strategy for training a knowledge model depends on large-scale, albeit noisy knowledge generated from a large LM (West et al., 2022). Are massive-scale LMs the only way to build commonsense capabilities? In addition to being an interesting scientific inquiry, if smaller LMs can indeed generate high-quality commonsense, training knowledge models will become far more efficient and accessible compared to the state-of-the-art.

We study the generation of commonsense knowledge from GPT-2 (a small LM) and compare that against GPT-3, a model that is orders of magnitude larger. Specifically, we focus on the task of generating *generics* – i.e. statements of commonsense knowledge about everyday concepts. While generics express general truths (e.g. "birds can fly"), exceptions abound (e.g. penguins do not fly nor do sleeping or injured birds). Nonetheless, generics form the basis of how we express our commonsense about the world (Hampton, 2012; Leslie, 2014).

We present I2D2, a new framework for generating generic statements from GPT-2 (depicted in

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9614–9630
July 9-14, 2023

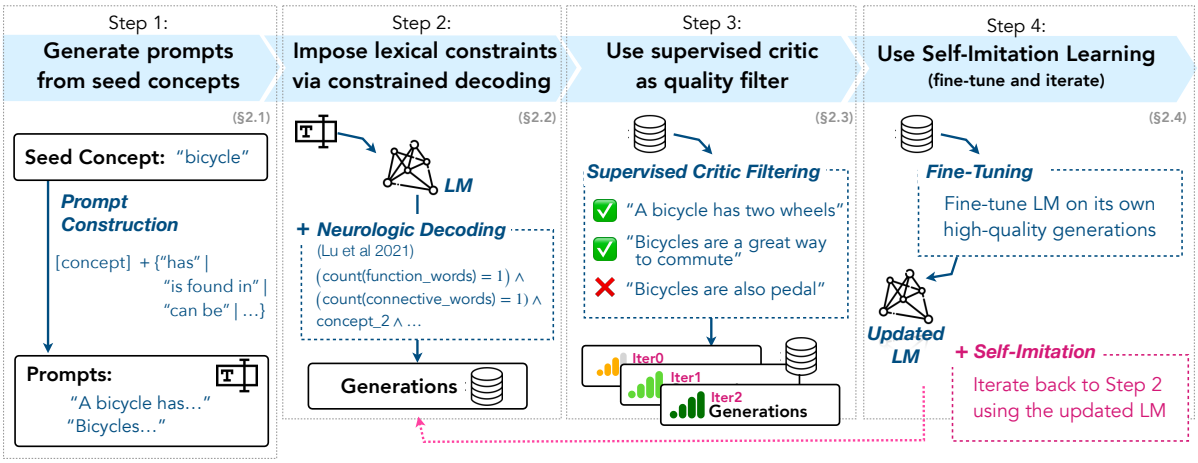

Figure 2: I2D2 is specifically designed to elicit *generics*—general statements about the world. I2D2 works by collecting a list of concepts and generates generics using Neurologic Decoding to constrain generations at decoding time. To ensure quality, I2D2 includes the use of a supervised critic to filter out false generations. The quality of the generations is further improved via iterative self-imitation learning whereby the language model is finetuned on the high-quality generics selected by the critic.

Fig. 2).[1] Out of the box, GPT-2 generations are anything but valid generics – often being repetitive, trivial, or resembling narratives. The key breakthrough for overcoming this challenge comes from (i) **constrained decoding**: in which generations are controlled to satisfy manually constructed lexico-syntactic constraints using Neurologic Decoding (Lu et al., 2021), and (ii) **self-imitation learning**: in which GPT-2 is iteratively fine-tuned on its own high-quality generations, automatically identified using a supervised critic model.

The marked disparity in scale makes the comparison between I2D2 and GPT-3 seem like an impossible match. However, constrained decoding and self-imitation enable I2D2 to overcome this limitation and even surpass the quality of knowledge generated by GPT-3. We formulate a binary-classification task on a human-annotated test set of generic statements and compare the precision-recall trade-off between I2D2 and Instruct-GPT-3 by ranking statements using their critic and perplexity scores, respectively.[2] I2D2 achieves an average precision of 0.92 and outperforms Instruct-GPT-3, which has an average precision of 0.82. Next, we show that iterative self-imitation learning dramatically improves the accuracy of generations from GPT-2 XL, even before applying the critic; increasing from 45% → 58% → 62% over three iterations. Finally, we construct `Gen-A-tomic` – a knowledge resource of generic statements generated by applying I2D2 to 40K everyday concepts. Compared to GenericsKB (Bhakthavatsalam et al., 2020), `Gen-A-tomic` is judged by humans to be more accurate (75% GenericsKB vs. 90% I2D2) while being larger (over 2X) in scale. Unlike GenericsKB, which was created through information extraction over text, I2D2 can provide commonsense knowledge for unseen concepts on-demand.

2 The I2D2 Framework

[Generally|Typically|Usually]? [a|an|the]?
`<noun_phrase>` `<relational phrase>`

Table 1: Template for automatically constructing morpho-syntactically varying prompts. '?' denotes the group of words is optional and '|' denotes the logical OR operator

I2D2 is a new framework for automatically generating generic statements using pretrained language models. Our language model of choice is GPT-2 XL. However, any auto-regressive language model can be used within I2D2.[3]

I2D2 generates generics in four stages. First, in **prompt construction**, we collect seed concepts (e.g. *bicycle*) and automatically construct several morpho-syntactically varying prompts (e.g. *"A bicycle has ..."*) (§2.1) for each concept. The

[1] I2D2: **I**terative **I**mitation and **D**ecoding for **D**istillation

[2] We use Instruct-GPT 3's `text-davinci-001` model in our experiments. In the rest of this paper, GPT-3 refers to this model, unless stated otherwise.

[3] In the rest of the paper, I2D2 refers to I2D2 using GPT-2 XL.

prompts are used as inputs to I2D2. Second, we employ **constrained generation** to control the style of text generated from the pre-trained LM at to mimic the style of generic statements(§2.2). Third, a supervised critic is used to **filter** out false and ill-formed generations (§2.3). Finally, the language model is finetuned on its own high-quality generations selected by the critic in an **iterative self-imitation learning** setup (§2.4). Figure 2 illustrates the overall framework.

2.1 Prompt Construction

Source of seed concepts: Our first set of concepts for generating generic knowledge is common noun phrases (e.g. "fruits"), selected from two resources: GenericsKB (Bhakthavatsalam et al., 2020) and ConceptNet (Speer et al., 2017). From GenericsKB, we retrieve all noun phrases for which there are at least five generic statements in the resource, resulting in a total of 8.5K seed concepts.[4] From ConceptNet, we retrieve noun phrases associated with the types `artefact` and `human`, identified based on hypernymy relationships to the corresponding WordNet senses. These lists are then manually vetted for validity to compile a shortlist totaling 1.4K seed concepts.[5]

Our second set of seed concepts is high-level human goals (e.g. "get better at chess") obtained from two sources: ProScript (Sakaguchi et al., 2021) and ATOMIC (Sap et al., 2019). We extract all `goals` that appear in the ProScript training data. From ATOMIC, we extract all `base events` and filter out hypothetical ones (e.g. "PersonX expects to win") based on an exclusion list (Appendix A.1).

To scale the number of seed concepts we prompt GPT-3 (Brown et al., 2020) with a *set-expansion template*, which is a prompt template for GPT-3 to generate items similar to a given set of items; see more details in Appendix A.1.1. Overall, after GPT-3 based expansion, we have 39K seed concepts, consisting of 26K noun phrases and 13K goals. Note that GPT-3 is only used for seed expansion and not for the generics generation.

[4]GenericsKB was found to consist of uncommon or specialized terminology (e.g. orpiment) that are not conducive for commonsense generation. Therefore, we select nouns with at least five generic statements so that the collected nouns are those that are capable of forming commonsense generics

[5]We choose `human` and `artifact` as much commonsense knowledge centers around these types. The list of concepts can be extended to other types as well (e.g. animals, natural phenomena) in the future.

Morpho-Syntactically Varying Prompts: We programmatically construct a large number of morpho-syntactically divergent prompts for each concept to facilitate the generation of a diverse set of generics. Prompts for noun phrases are constructed based on the template shown in Table 1.

Each concept is paired with a *relational phrase*, e.g. "can be", "is found in", from a manually constructed list; Appendix A.1.2 presents more details. Inspired by Leslie (2008), we prefix adverbs (such as "generally", "usually", and "typically") to the prompts. We find, empirically, that these prefixes encourage the language model to generate general statements, instead of long-form, narrative-like text. An article is optionally prefixed before the concept for grammaticality. For a given (concept, relational phrase) pair, we construct all prompt combinations according to the template above and choose the one with the lowest PLM (GPT-2 XL in our experiments) perplexity. For the goal seed concepts, from each goal we create four separate prompts by prepending each of these prefixes: "In order to", "Before you", "After you", and "While you".

Source of related concepts: NLP applications often require knowledge that connects two concepts together in some given context. For example, to solve a QA problem, it might be important to have background knowledge about the relationship between a "hotel" and a "credit card", e.g. "At a hotel, credit cards can be used to make a payment". We obtain concepts related to a seed concept from GPT-3 using a custom template; see details in Appendix A.1.3. In Section 2.2, we describe how I2D2 is able to generate such generic statements.

Finally, we filter out all prompts whose per-word perplexity under GPT-2 XL is above a threshold of 250. This allows us to apriori filter out ill-formed prompts such as "Typically, a hall are planted at . . . ". This results in a total of 1.6M prompts.

2.2 Constrained Generation using NeuroLogic Decoding

Why Constrained Decoding: Small language models like GPT-2 XL struggle with text degeneration (Holtzman et al., 2019). Text generated can be trivial, repetitive, or long-winded resembling a narrative. In contrast, generic statements are simple and short (Tessler and Goodman, 2016). The main challenge is to generate statements consistent with the linguistic style of generics, while using an inherently weak language model. To address this, we

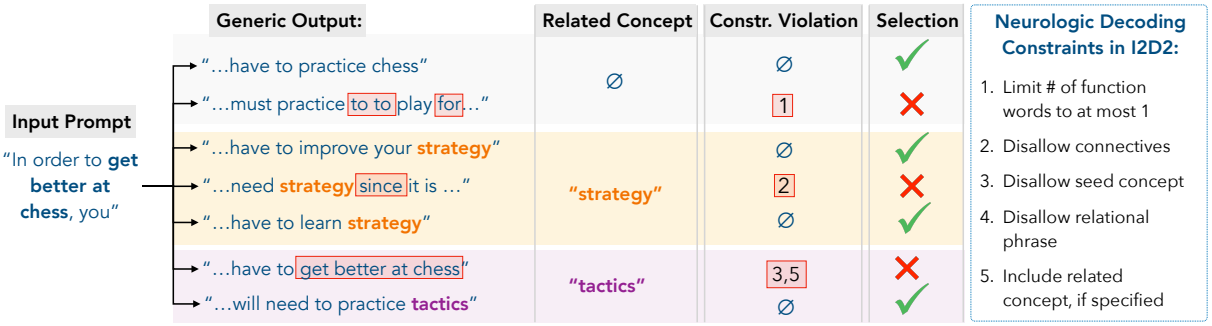

Figure 3: Example outputs of I2D2 for the concept "get better at chess". We add constraints to our constrained generation algorithm to include the related concept.

could either adapt the model to our task, through fine-tuning or apply novel decoding algorithms to substantially improve the generation quality. As the only resource of generic statements, GenericsKB (Bhakthavatsalam et al., 2020) could be used for fine-tuning. But it primarily focuses on scientific concepts and, as we show in §3, lacks diversity and scale. Crowdsourcing a new dataset from scratch is resource intensive. Thus, we focus on better decoding methods instead of relying on the standard top-p, top-k, or beam search algorithms.

What is NeuroLogic Decoding: NeuroLogic Decoding (Lu et al., 2021) enforces satisfaction of given constraints in generated text. It can handle any constraints—*positive* (a given word must be included in the generation) or *negative* (the given word must not be generated)—which can be expressed in conjunctive normal form. The constraint satisfaction problem is solved approximately using beam-search by introducing a high-penalty term for violating constraints.

NeuroLogic Decoding in I2D2 Our work is the first to use NeuroLogic Decoding for knowledge generation. The application of NeuroLogic to our problem is based on two key observations. First, we find that limiting the number of function words (e.g., "in", "on", "of") in a sentence implicitly controls its length. Next, excluding connective words (e.g., "although", "since", "furthermore") can make generations short and succinct.

These logical constraints can be enforced at decoding time to steer the model toward desired text using NeuroLogic Decoding. We devise the following set of constraints, represented in CNF. Constraints are exemplified in Figure 3 and further detailed in A.1.4.

$$
\begin{aligned}
&\left(count(\texttt{function\_words}) \leq 1\right) \\
&\wedge \left(count(\texttt{connective\_words}) = 0\right) \\
&\wedge \neg \texttt{source\_concept} \\
&\wedge \neg \texttt{relational\_phrase}
\end{aligned}
$$

Given the 1.6M programmatically constructed prompts and their associated constraints, we generate ten generations for each prompt using NeuroLogic Decoding applied to GPT-2 XL. Overall, we generate about 16M statements which must now be filtered to preserve quality.

2.3 Supervised Critic

LMs can generate hallucinations and false statements about the world (Ji et al., 2022). We similarly observe invalid or false statements output by our constrained decoding method. To address this, we train a supervised critic model to predict the veracity of a generation. We create a training set of ∼12K statements, with up to four sampled generations for each concept from a held-out set of ∼3K concepts. The labels for each generation are collected using the same procedure as the evaluation data, which is described in Section 3.2. We train a RoBERTa-Large (Liu et al., 2019) classifier as our critic model to identify valid generic statements.

2.4 Self-Imitation Learning

Why Self-Imitation: NeuroLogic Decoding allows I2D2 to generate statements in the style of generics. But the deficiencies of using a weak language model are still apparent as the critic model has to discard a majority of the candidate statements due to their low quality. Intuitively, using a better language model should make it more likely for NeuroLogic to find higher-quality candidates. We posit that fine-tuning the language model on

Concept: **"board games"**		Concept: **"friendship"**	
Gen-a-tomic	GPT2-xl (OTS)	**Gen-a-tomic**	GPT2-xl (OTS)
A board game is fun, if you're good at it. Board games have been around for decades. A board game should have at least two players. A board game may have fun components. A board game can be fun.	Board games can be found in your local game store, but […] Board games can be placed in one of three categories: Board Games […] Board games can be attached to one of the following categories: Board Games […]	A friendship can last for years. A friendship can be built by loyalty. A friendship has an emotional support system. A friendship is defined by a relationship of mutual respect. Friendships are built over time.	Friendships can be found in all walks of life, from the most intimate to […] A friendship can consist of one or more of the following types of […] A friendship can be taken away from you in one of three ways: You […]
GenericsKB	GPT-3 (instruct)	GenericsKB	GPT-3 (instruct)
Board games are used for play. Board games are used for teaching. Board games are located in cupboards. Board games are located in dens. Board games are fun.	Board games can be attached to the wall. Board games are played with a set of game pieces and a game board. A board games may have 2 to 4 players Board games are for entertainment.	Friendships are relationships. Friendship is the hidden energy calling nations to justice. Friendship means helping ease the loneliness in life. Friendship is also synonymous with fidelity. Friendship is a subdivision of faith.	A friendship lasts longer when both people are interested in each other. A friendship can be taken away from someone if they hurt the other person. A friendship lasts about 7 years. Friendships are able to last many years.

Figure 4: Examples of generics in `Gen-A-tomic` and GenericsKB, and those generated by off-the-shelf GPT2-xl and GPT-3 instruct. Examples in green are good generics, red are bad generics, and orange are questionable ones.

Algorithm 1 The I2D2 framework

Input: P_0: a pre-trained language model
$\mathbf{X} = \{(x_i, \mathcal{C}_i)\}$: Set of prompts and constraints
$\Omega(\cdot)$: a critic model
N: number of iterations
δ: classification threshold

for $k = 0, 1, \ldots, N-1$ **do**
  $\mathcal{D}_k \leftarrow \{\}$
  for $(x_i, \mathcal{C}_i) \in \mathbf{X}$ **do**
    $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} P_k(\mathbf{y}|\mathbf{x}) - \lambda \sum_{c \in \mathcal{C}_i} \big(1 - c(\mathbf{y})\big)$
    ▷ Constrained Decoding
    $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \{\hat{\mathbf{y}}\}$ ▷ Add generations to data pool
  Train Ω on annotated samples of $\mathcal{D}_k$ ▷ Train Critic
  $\tilde{\mathcal{D}}_k \leftarrow \{\mathbf{y} | \mathbf{y} \in \mathcal{D}_k \text{ and } \Omega(\mathbf{y}) > \delta\}$ ▷ Apply Critic
  $P_{k+1} \leftarrow \arg\max_\theta \mathbb{E} \log P_k(y_n | y_1, \ldots y_{n-1}); \mathbf{y} \sim \tilde{\mathcal{D}}_k$
  ▷ Train LM on high-quality generations

its own high-quality generations can make it better suited for knowledge generation by steering its distribution towards higher-quality samples.

What is Self-Imitation: In the reinforcement learning literature, self-imitation learning (Oh et al., 2018) is an actor-critic algorithm for learning to reproduce past *good* actions of an agent in an environment. ⟨`State`, `action`, `reward`⟩ triples from past experience are stored in memory and an action taken in the past is chosen only when that action resulted in higher reward than expected.

Self-Imitation in I2D2: Our method closely follows self-imitation of (Oh et al., 2018), but uses a pre-trained language model as the *'actor'* and a trained classifier as the *'critic'* models. Moreover, we update the language model using the standard conditional language modeling objective, maximum likelihood. I2D2 is formally described in Algorithm 1.

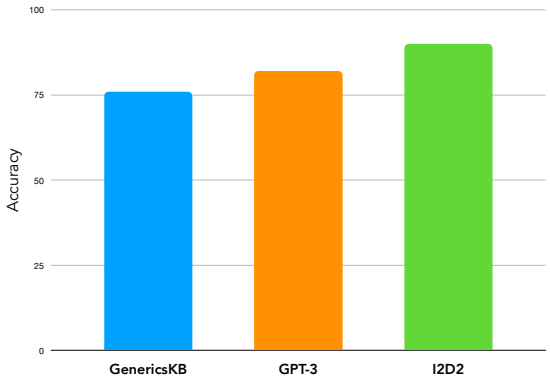

Figure 5: The accuracy of I2D2 generations is higher than GPT-3 (a 100× larger model) and GenericsKB

3 Experiments and Results

We describe results from our experiments comparing I2D2 with GPT-3, GPT-2 XL and GenericsKB in more detail below. Figure 4 shows outputs sampled from these sources.

3.1 I2D2's generations are more accurate than GPT-3 and GenericsKB

We compare the accuracy of generations from I2D2, GPT-3, and GenericsKB (see Figure 5). The best accuracy achieved by GPT-3 in our experiments is **82%**. GenericsKB (Bhakthavatsalam et al., 2020) is a static resource of generic knowledge created through information extraction over three large text corpora: the Waterloo corpus, SimpleWikipedia, and the ARC corpus. This work released a large-scale dataset of 14M generations and a high-quality subset of 1M generic statements. We compare GenericsKB's best 1M against our corpus. We randomly sample 1K generic statements from GenericsKB and I2D2 and ask annotators on Amazon Mechanical Turk (MTurk) to rate the validity of the generic statement. We find that while

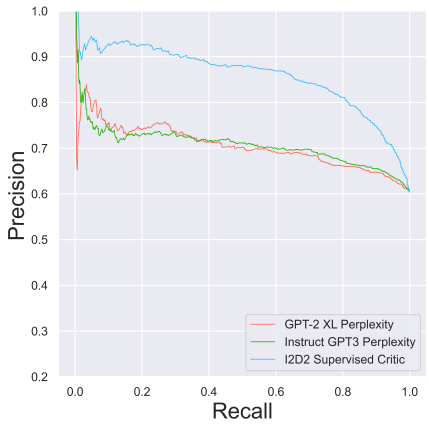

Figure 6: Comparing the PR Curve of I2D2 Critic and the language model perplexities based on GPT2-XL and GPT-3.

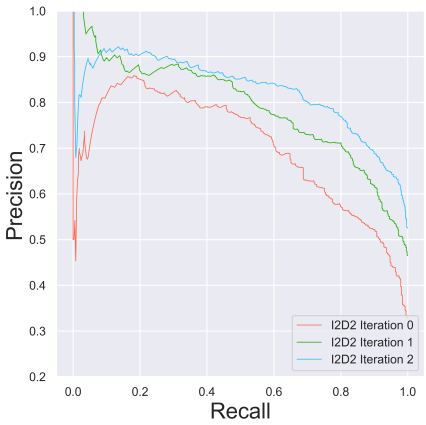

Figure 7: Comparing the PR Curve of I2D2 iterations 0 through 2

only **76%** of statements in GenericsKB were annotated as accurate, over **90%** of statements in I2D2 were judged as valid. The results show that I2D2 is more accurate than GenericsKB, while being larger. I2D2 is also more accurate than GPT-3, while using 100× fewer parameters in its model.

3.2 I2D2 results in better generics than GPT-3

Systems We wish to compare how GPT-3, given the same set of prompts as our approach, can generate and identify valid generics. For a given prompt, we generate ten generations from each system. GPT-3 is prompted in a few-shot manner with an instruction and six examples. We use different sets of few-shot examples for noun phrases and goals. Appendix A.1.6 further details the instruction and in-context examples provided to GPT-3. I2D2, using a supervised critic, assigns a score to each generated statement. For GPT-3, we use the perplexity assigned to a generation as an indicator of validity. As an additional baseline, we also compute perplexity under off-the-shelf GPT-2 XL.

Evaluation Data We set aside 300 concepts for evaluation. Each concept is associated with several prompts (on average 40). We generate ten generic statements for each prompt from I2D2 and GPT-3. Next, from all generations for a concept, we randomly sample four statements generated by each system. A generic statement is considered valid if it is a generally true statement about the world. Three annotators on MTurk rate the validity of each generated statement.[6] Annotation template and instructions are detailed in Appendix A.1.5. At least two out of three annotators agreed on a label 92.5% of the time over all 4 statements.[7]

Metrics Given the human-annotated test set of generics, we compare the precision-recall trade-off between I2D2 and GPT-3. Each system assigns a score to each generic statement, allowing us to rank the statements from most to least likely to be a generic. Combined with the human annotations of the validity of a statement, we plot a precision-recall (PR) curve. It allows us to evaluate the accuracy of each system as the number of statements it outputs varies, which is important since different tradeoffs between quantity and quality of output may be desired for different application settings.

Results Figure 6 shows the impact of including a supervised critic to identify valid generic statements. We find that GPT-3, while impressive, lags significantly behind our supervised critic in identifying which generic statements are valid. The off-the-shelf GPT-2 XL model is the worst at identifying valid generic statements. Perplexity alone is not a good indicator of what a valid generic is.

I2D2 uses both a generator and a discriminator. To evaluate the generator, we sample from its generations over the test set of prompts. For a given set of generations, human annotators judge whether

[6] Annotators select one of four choices: {true, false, don't know, garbled output}.

[7] We provide pairwise annotation agreement. Since our generations should ideally be valid, we produce a skew towards a single label, problematic for κ (Feinstein and Cicchetti, 1990).

the statement is true or false. We compute accuracy against human labels and use that as a metric to measure the quality of the generator.

The cautions against GPT-3 comparison There are growing concerns in the research community about the lack of open availability of GPT-3. Several versions of GPT-3 are available through an API, but the details of the training data used for each version are largely unavailable or underspecified. Direct comparison with GPT-3 is, therefore, becoming increasingly challenging. In this work, we compare against the 'text-davinci-001' version of the GPT-3 model and note that newer models might do better. However, extracting the best performance from GPT-3 is beside the point of our work. We believe that as a community, we must investigate alternative approaches that do not just rely on scale. Case in point, our results in §3.5 demonstrate that the smaller `curie` version of GPT-3 outperforms the much larger `davinci` version, through better training.

3.3 I2D2 gets better through iterative self-imitation learning

Systems For self-imitation learning, we generate a large corpus of generations and filter out invalid statements using the supervised critic to yield a "purified" subset. We compare generations from I2D2 using off-the-shelf GPT-2 XL and outputs from two additional iterations of fine-tuning.

Evaluation Data We use the same held-out test set of prompts for this experiment.

Metrics Here, we evaluate the accuracy of the generations before applying the supervised critic.

Results We show that a language model gets iteratively better as it gets finetuned on its own high-quality generations over each iteration. The raw accuracy of the generations, before applying the critic, improves from 45% $\rightarrow$ 58 % $\rightarrow$ 62% over three iterations. We also compare the precision-recall trade-off between the three iterations. Figure 7 shows the effectiveness of self-imitation learning over three iterations.

3.4 `Gen-A-tomic` is more diverse than GenericsKB

`Gen-A-tomic` is a large set of generic statements, but some of these may be semantically equivalent

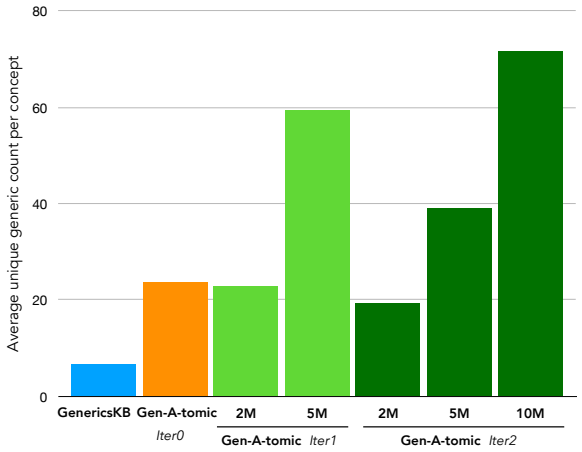

Figure 8: Compared to GenericsKB, the estimated average unique number of generics per concept is higher for any version of `Gen-A-tomic`.

to one another. Since exact quantification of semantically distinct statements in the dataset is intractable, we employ a survey method called Mark and Recapture (MnR) (Seber et al., 1982; The U.S. Geological Survey, 2018) commonly used by ecologists to estimate a large population size via sampling. This method captures individuals of a population in two (or more) stages. In the first capture, the generics capture (i.e., sampled) are *marked* and released. At a later capture, the number of *recaptured* generics[8] are counted and the population size estimated. Then, we employ the Chapman estimator for MnR (Brittain and Böhning, 2009; Chapman, 1951) to estimate the population size of unique generics in the dataset. More details can be found in Appendix A.1.7.

We compare the estimated *per concept* average count of unique generics for GenericsKB and `Gen-A-tomic`. Overall, we find that `Gen-A-tomic` includes at least triple the amount of generics per concept compared to GenericsKB. We also observe that the estimated unique generics per concept is higher for the best cuts of the `Gen-A-tomic` dataset. Experiments with embedding-based similarity methods yielded similar results.

3.5 Smaller, better-trained versions of GPT-3 outperform larger ones

We compare three versions of the GPT-3 model available on the OpenAI API: `davinci`, `curie-instruct` and `davinci-instruct` (Ouyang et al., 2022; Brown et al., 2020). Interestingly, we

[8] A recapture is determined by the second sample's BLEU score with respect to the already captured.

find that the curie-instruct model, despite being a much smaller model, generates more valid generic statements compared to the much larger davinci model. The instruct models (including curie-instruct) were trained using reinforcement learning on human feedback. The accuracy (validity) of statements generated by the three GPT-3 models on the same set of test prompts are 53.3% (davinci), 60.6% (curie-instruct), and 81.9% (davinci-instruct). These results further demonstrate that better training can result in smaller models performing better than larger models.

Our work adds to the growing body of evidence from recent work that large language models have not been trained optimally (Kaplan et al., 2020) and it would be worthwhile to look for better training strategies to achieve high performance using smaller, affordable, greener models.

4 Related Work

Generics Generics like "dogs are friendly" describe observed "truths" or defaults about the world for which exceptions can be found (e.g., not all dogs are friendly in practice). Generics have been studied quite extensively in philosophy, linguistics, and psychology. While they are clearly important to human reasoning, in particular, to non-monotonic reasoning (Carlson and Pelletier, 1995; Pelletier and Asher, 1997), they have also been long debated for their puzzling properties which renders them difficult to formally analyze (Leslie, 2012, 2008; Hampton, 2012; Liebesman, 2011). Bhakthavatsalam et al. (2020) demonstrated the usefulness of generics in language understanding by providing generic statements to text models and showing improvement on question-answering and explanation generation. However, being a static resource, GenericsKB cannot provide knowledge for unseen concepts. To be useful across a wide range of tasks and datasets, a more comprehensive resource of generics is required. I2D2 can generate generics for arbitrary concepts and even generics relating two concepts—a feature unique to I2D2. I2D2 can is easily extensible temporal ("during a cold night, people need a blanket") or comparative ("a tennis ball is smaller than an office chair") generic knowledge, leading to a more comprehensive commonsense knowledge model.

Commonsense Knowledge Various methods for representing commonsense knowledge have been proposed in the literature. ConceptNet (Speer et al., 2017) focused on the conceptual commonsense relationship among various concepts and entities in their knowledge graph. Atomic (Sap et al., 2019) and Atomic2020 (Hwang et al., 2021) have offered symbolic commonsense knowledge graphs representing relational inference focusing on the "If-Then" (cause-effect) reasoning. Fine-tuned on Atomic, Comet (Bosselut et al., 2019) has offered a neural knowledge model that can reason about situations beyond the symbolic knowledge graphs. Unlike our current framework, however, previous commonsense knowledge models typically only handled data in the form of structured triples and were predominantly focused on commonsense about events. I2D2 is the first knowledge model focused on generic knowledge expressed in natural language. Uniquely, we also provide a critic model that can filter invalid or ill-formed generations.

Symbolic Knowledge Distillation Collecting high-quality knowledge at scale has been a long-standing challenge. The traditional way is to collect by human annotation (Speer et al., 2017; Sap et al., 2019), which can be time-consuming and expensive. Bhakthavatsalam et al. (2020) extracted generics by filtering and cleaning based on 1.7B sentences from three large text corpora. However, manually constructed resources and resources extracted from large corpora can be difficult to extend. Recent works showed that pre-trained language models can be a good source of knowledge (West et al., 2022; Zhang et al., 2022). Symbolic knowledge distillation (SKD) (West et al., 2022), for instance, has generated even-centric inferential knowledge from GPT-3 and distills it into GPT-2. While these methods present promising results, they primarily rely on using GPT-3 and only handle knowledge about events in a structured triple format. I2D2, on the other hand, relies only on GPT-2's own generations to improve itself and generates knowledge in natural language.

Self-Imitation Learning Self-imitation learning (Oh et al., 2018) was proposed as a reinforcement learning method in which an agent learns to replicate past good actions. More recently, a similar approach was applied in dialog models (Thoppilan et al., 2022; Xu et al., 2022) and code generation (Haluptzok et al., 2022). However, recent applications have relied on models much larger than GPT-2 XL used in I2D2. Moreover, while (Haluptzok et al., 2022) have explored the idea of self-imitation

learning in language models, their method relies on a compiler that is, by definition, 100% accurate. Instead, the supervised critic in I2D2 can be noisy, especially for identifying generics, which have paradoxical properties that make its formalization very difficult (Mari et al., 2012). We also show that self-imitation learning is beneficial when done over multiple iterations. In principle, I2D2 could be improved iteratively through a life-long learning process. But, under what conditions would the performance gains plateau is an interesting open future research question.

5 Conclusion

We present I2D2— a novel framework for generating generic knowledge from language models using constrained decoding and self-imitation learning. I2D2, while using orders of magnitude fewer parameters, can still outperform GPT-3 at the task of generating high-quality generic statements. We also show that `Gen-A-tomic` is higher-quality, larger-scale, and more diverse than the static GenericsKB dataset. I2D2 provides on-demand access to generic knowledge that can bridge the gap in commonsense knowledge, often observed in even the largest LMs available today.

6 Acknowledgements

We thank our colleagues on the Beaker Team at the Allen Institute for AI for helping with the compute infrastructure. This work was supported in-part by DARPA MCS program through NIWC Pacific (N66001-19-2-4031). We thank the reviewers and ACL area chairs for their valuable feedback that made our work better.

Limitations

Comparison with GPT-3: There are growing concerns in the research community about the lack of open availability of GPT-3. There are several versions of the model and the details of the training data used for each version are largely unavailable. Direct comparison with GPT-3 is, therefore, becoming increasingly challenging. In this work, we compare against the 'text-davinci-001' version of the GPT-3 model and note that newer models might do better. However, extracting the best performance from GPT-3 is beside the point of our work. We believe that as a community, we must investigate alternative approaches that do not only rely on scale.

Undesirable Generations: Language models, large and small, have been shown to be prone to generating toxic text (Gehman et al., 2020). I2D2 relies on GPT-2 XL could also potentially generate toxic statements. While the trained critic model is able to filter out most toxic generations, we estimate the proportion of undesirable generations using the Delphi (Jiang et al., 2021) model. We find that $\sim 1.3\%$ of the generations may not be morally acceptable, either because the statements are not accurate, not verifiable, too restrictive, or they are potentially toxic.

Self-Imitation Iterations: In this work, we only try two iterations of self-imitation due to resource constraints. Exploring the effects of more self-imitation iterations is left for future work. But, based on the performance improvements we observed after two iterations, we hypothesize that the improvements could diminish with each future iteration.

Runtime Efficiency A batch of 32 generations from I2D2 takes 3mins on a single RTX A6000 GPU. NeuroLogic Decoding is the most computationally expensive component. As constrained decoding methods become more efficient, the runtime of I2D2 will also improve. Our focus in this work is to study the quality of generations and we leave runtime efficiency improvements to future work.

Ethical Statement

Crowdsourcing: Annotations were conducted on Amazon Mechanical Turk. For this project, we obtained an exemption through our institution's internal IRB. We do not retain nor publish deanonymizing information such as MTurk IDs. Throughout the project, we maintain an average hourly rate of $15/hour for all our evaluations. More detail on annotation is available in Appendix A.1.5.

Intended Use: The framework I2D2 is intended to enable further research in knowledge generation using a smaller and openly available language model like GPT-2. As discussed towards the end in §3, large language models like GPT-3 are indeed more capable of generating commonsense knowledge than off-the-shelf GPT-2, but they as unavailable for open use. This work seeks to expedite a more sustainable yet high-quality generation using smaller models that are accessible to all.

Gen-A-tomic can be used as a resource of static knowledge for downstream applications in NLP. As discussed in the Limitations section above, there may exist a small number of generations that may be considered toxic and harmful for use. Therefore, we emphasize that the dataset should be used for research purposes only. Moreover, because the dataset has been vetted by crowdworkers originating from North America, the knowledge of the retained generics in Gen-A-tomic is most strongly representative of generalizations or 'truths' of the English-speaking Western, specifically North American cultures. Extending it to encompass a more diverse set of world knowledge is a topic of our future research.

References

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *arXiv preprint arXiv:2005.00660*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Ronald J. Brachman and Hector J. Levesque. 2021. Toward a new science of common sense.

Sarah Brittain and Dankmar Böhning. 2009. Estimators in capture–recapture studies with two sources. *AStA Advances in Statistical Analysis*, 93(1):23–47.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Gregory N Carlson and Francis Jeffry Pelletier. 1995. *The generic book*. University of Chicago Press.

Douglas George Chapman. 1951. Some properties of the hypergeometric distribution with applications to zoological sample censuses. berkeley. *Calif: University of Catifomia Publications in Statistics*, 195(1).

Alvan R Feinstein and Domenic V Cicchetti. 1990. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *CoRR*, abs/2009.11462.

Patrick Haluptzok, Matthew Bowers, and Adam Tauman Kalai. 2022. Language models can teach themselves to program better. *arXiv preprint arXiv:2207.14502*.

James A Hampton. 2012. Generics as reflecting conceptual knowledge. *Recherches linguistiques de Vincennes*, pages 9–24.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation.

Liwei Jiang, Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Maxwell Forbes, Jon Borchardt, Jenny Liang, Oren Etzioni, Maarten Sap, and Yejin Choi. 2021. Delphi: Towards machine ethics and norms. *arXiv preprint arXiv:2110.07574*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Sarah-Jane Leslie. 2008. Generics: Cognition and acquisition. *Philosophical Review*, 117(1):1–47.

Sarah-Jane Leslie. 2012. Generics. In Gillian Russell and Delia Fara, editors, *Routledge Handbook of Philosophy of Language*, pages 355–366. Routledge.

Sarah-Jane Leslie. 2014. Carving up the social world with generics. *Oxford studies in experimental philosophy*, 1.

David Liebesman. 2011. Simple generics. *Noûs*, 45(3):409–442.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*.

Alda Mari, Claire Beyssade, and Fabio Del Prete. 2012. *Genericity*, volume 43. OUP Oxford.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. 2018. Self-imitation learning.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Francis Jeffry Pelletier and Nicholas Asher. 1997. Generics and defaults. In *Handbook of Logic and Language*.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. 2021. proscript: Partially ordered scripts generation via pre-trained language models. In *Empirical Methods in Natural Language Processing, Findings of EMNLP*.

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

George Arthur Frederick Seber et al. 1982. The estimation of animal abundance and related parameters.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Michael Henry Tessler and Noah D. Goodman. 2016. A pragmatic theory of generic language. *ArXiv*, abs/1608.02926.

The U.S. Geological Survey. 2018. Capture-mark-recapture science. https://www.usgs.gov/centers/eesc/science/capture-mark-recapture-science, Last accessed on 2022-09-17.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239.

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

Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2022. Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback. *arXiv preprint arXiv:2208.03270*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

A Appendix

A.1 Selection of Concepts and Goals

ConceptNet concept selection To select ConceptNet concepts, we first build a list of `artefact` and `human` terms from WordNet by hierarchically traversing the hypernymy hierarchy (by depth) starting from the *artefact%1:03:00* and *person%1:03:00*, respectively. We then select ConceptNet concepts that belong to the lists in the same order as the WordNet list. The concepts in the list are sequentially evaluated manually to build a total of 1K `artefact` and 400k `human` terms. The totaling 1.4k concepts are then used as seed ConceptNet concepts.

ATOMIC goal selection To select goals from ATOMIC, we obtain the complete list of `base events` (e.g. "PersonX adopts a cat"). We drop the "PersonX" prefixation and all mentions of "Person" (e.g. "PersonY", "PersonZ"). Additionally, because we want to select for goals that are achievable, we remove all irrealis or hypothetical situations (the described situation or event has not taken place). More specifically, we filter out all events with the verbs 'need', 'want' 'wish', 'hope', 'dream', 'expect', 'imagine', 'mean', and 'plan'; negated events (e.g. "PersonX does not get the job"); and events modified by modals that indicate permission or obligation (e.g. 'should'). In this manner we arrive at a list of 8.5K goals from ATOMIC.

A.1.1 GPT3 for Set Expansion

We develop a template for set expansion based on GPT3.

> Generate more concepts.
> 1: <sampled concept 1>
> 2: <sampled concept 2>
> 3: <sampled concept 3>
> 4: <sampled concept 4>
> 5: <sampled concept 5>
> 6:

Set expansion is done in several iterations. We define K as the number of new concepts to be found in each iteration. We construct a prompt as shown above by sampling five concepts. We get five outputs for each prompt. We skip concepts whose generation perplexity is lower than a set threshold (8 in our experiments). Thus, at most five new concepts are found with each call to the OpenAI API. At the end of each iteration, newly found concepts are added to the seed list of concepts. This iterative process allows us to slowly expand the original list with new related concepts.

A.1.2 List of relational templates

Noun phrases are combined with one of the following verb phrases if they are obtained from GenericsKB:

> are
> is
> have
> can
> has
> should
> produces
> may have
> may be

If a noun phrase is obtained from ConceptNet, we expand the templates available in the (file "templates.txt" attached in supplementary material.

A.1.3 Template for obtaining related concept

Related concepts for a given concept are also obtained from GPT3. We use the following prompt:

> Generate five words that are related to the given word.
>
> Word: hotel
> Related Words:
> 1: Credit card
> 2: Fee
> 3: Resort
> 4: Parking lot
> 5: Reception
> ==
> Word: <given concept>
> Related Words:

GPT-3 generates five related concepts for each given word.

A.1.4 Constraints for Neurologic Decoding

We use four sets of constraints for Neurologic Decoding:

$$\begin{aligned}&\big(count(\texttt{function\_words}) \leq 1\big)\\ &\wedge \big(count(\texttt{connective\_words}) = 0\big)\\ &\wedge \neg\texttt{source\_concept}\\ &\wedge \neg\texttt{relational\_phrase}\end{aligned}$$

`function_words` comprises of {``in", ``on", ``of", ``for", ``of", ``at", ``in", ``anybody", ``it", ``one", ``the", ``a", ``that", ``or", ``got", ``do"}.

`connective_words` comprises of {``without", ``between", ``he", ``they", ``she", ``my", ``more",

```
``much", ``either", ``neither",
``and", ``when", ``while", ``although",
``am", ``no", ``nor", ``not", ``as",
``because", ``since", ``although",
``finally", ``however", ``therefore",
``because", ``consequently",
``furthermore", ``nonetheless",
``moreover", ``alternatively",
``henceforward", ``nevertheless",
``whereas", ``meanwhile", ``this",
``there", ``here", ``same", ``few", ``1",
``2", ``3", ``4", ``5", ``6", ``7", ``8",
``9", ``0", ``similar", ``the following",
``by now", ``into"}
```

We additionally add the `source_concept` and the associated `relational_phrase` that were used to compose the prompt.

A.1.5 Human Evaluation

All human evaluations were conducted through the Amazon Mechanical Turk (IDs) platform. We sourced our annotators from a pool of 168 crowdworkers manually selected from best performing workers in a round of open paid qualification. For the evaluation, the workers asked to rank model predictions on a 4-point validity likert scale. The Figure 9 shows a screenshot of the annotation template with the full set of instructions used for collecting the training set and for evaluation of generic statements. Throughout the entirety project, we maintained an average of $15/hour pay rate.

We obtained IRB exemption for our evaluation from our institution's internal institutional review and ethics board. We did not collect any deanonymizing information nor do we publish with our dataset sensitive information such as MTurk IDs in full compliance to the exemption clauses found in 45 CFR 46.104(d)(2,3). Additionally, the extent of the crowdsourcing for the present work is limited to judgments based on world knowledge, we have no reason to believe that our crowdsourcing set up posed harm or discomfort beyond the minimal risk as defined by 45 CFR 46.102(i). Our exempted status does not require for us to use consent forms with our crowdsourcing.

As shown in the screenshot, the evaluations were conducted in English. Although we did not collect demographic information from the crowdworkers, our previous internal study from 2020 tells us that over 90% of our annotators are English speakers from the US. Thus, the evaluation received as to the validity of the generic statements most strongly reflect the North American

A.1.6 GPT-3 generics generation template

Generics are generated from GPT-3 using the following template:

> Generate statements that are generally true in the real world.
> An apple is a fruit.
> Violins are used for music.
> Aardvarks are mammals.
> Accidents cause injuries.
> Protein is made of amino acids.
> Apples can be red or green.
> $<$ test prompt $>$

We generate ten continuations for the prompt above.

A.1.7 Mark-and-Recapture Details

We use MnR to estimate the *unique population size* of our large datasets thereby gauging the diversity of the dataset. For our implementation of MnR, we perform two random captures using a sample size of 30% (of the total dataset size) at each capture. A generic in the second capture is considered a *recapture* (i.e., individual seen in the first capture) if it exceeds a textual similarity threshold (BLEU score > 0.85) with the *generics of the same concept* from the first capture as the reference. The threshold was determined via several rounds of experimentation and manual evaluation to determine a reasonable level of textual similarity. Then, we employ the Chapman estimator for MnR (Brittain and Böhning, 2009; Chapman, 1951) to estimate the population size of unique generics in the dataset.

A.1.8 Categories of generated Generics

In our preliminary experiments, we collected crowdsourced annotations to label generated generics with categories derived primarily from (Leslie, 2008). We found that the task was extremely challenging for non-expert crowdworkers. For example, recognizing "mosquitoes carry the West Nile virus" as a *striking* generic requires a domain knowledge that often falls outside common knowledge. As a result, we encountered low inter-annotator agreement scores leading us to not include them in the main discussion. However, based on samples from the first iteration of I2D2, we observed the following distribution of categories of generics:

1. semi-definitional (e.g., "laser produces a beam of light"): 45

2. characterizing: 35

3. striking or majority: 20

A.2 Regarding License for I2D2 and Gen-A-tomic

The codebase for I2D2 will be licensed and released under the Apache License 2.0. The Gen-A-tomic will be licensed under CC-BY.

A.3 Responsible AI Checklist

Number of parameters used I2D2 mainly uses two models: GPT-2 XL with 1.5B parameters and RoBERTa-large with 354M parameters.

Total Computation Cost We use Nvidia A6000 GPUs (with 48G RAM) in our experiments. The bulk of the computation cost is in executing constrained decoding over a large number of prompts to create Gen-A-tomic. We can generate 10 generations each for 32 prompts in 2 mins. Overall, to generate 16M generic statements, we need about $\sim$1500 GPU hours. That said, creation of the large corpus is a one-time cost. I2D2 is readily applicable as a knowledge model that can be queried on-the-fly. Retraining the language model takes $\sim$24 GPU hours.

Hyperparameters We use the following hyperparameters for different components:

For constrained decoding:

batch size 32,
beam size 10,
max generation length 30,
min generation length 2,
length penalty 0.1

For training the critic model:

batch size 64,
learning rate $1e-4$,
training epochs 5,

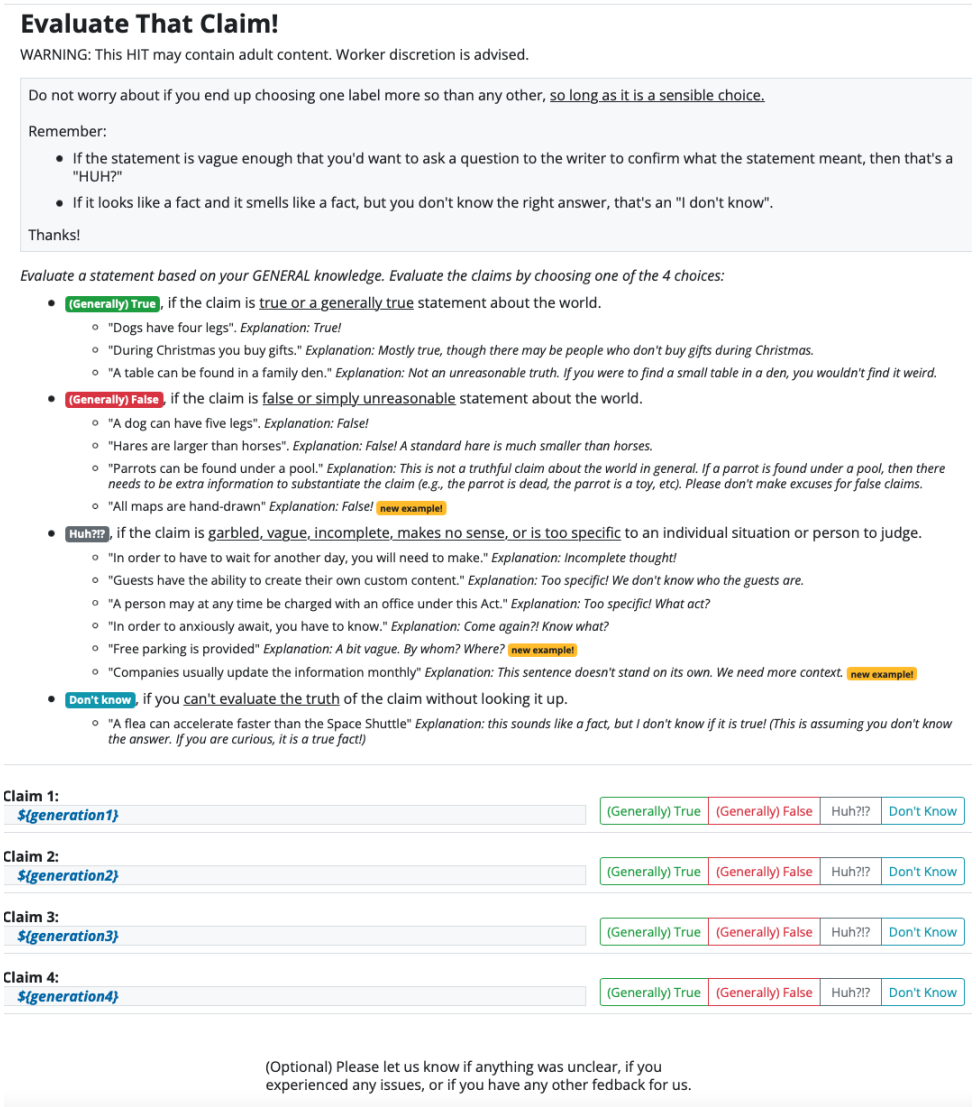

Figure 9: A screenshot of the template used for obtaining annotations on the Amazon Mechanical Turk platform.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section following the Conclusion

☑ A2. Did you discuss any potential risks of your work?
Ethics and Limitations sections following the Conclusion

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 2.

☑ B1. Did you cite the creators of artifacts you used?
Section 2.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A.2

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Ethics section following Conclusion

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Ethics section following Conclusion with further detail in A.1.5

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
A.1.5 and briefly under Ethics section following Conclusion.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.1

C ☑ Did you run computational experiments?

Approach and Methods in Section 2; Results in Section 3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A.3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In Section 2 as well as Appendix A.3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
The full implementation details about existing packages employed in the work will be included in the code release.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 3.1 and Ethics section

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Figure 8

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Appendix A.1.5

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No consent was required. Appendix A.1.5 details it

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Appendix A.1.5

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We did not collect any demographic information. We do have some generalizations based on past experience. This is detailed in Appendix A.1.5.