Dynamic Neuro-Symbolic Knowledge Graph Construction for Zero-Shot Commonsense Question Answering

Antoine Bosselut,[1,2] **Ronan Le Bras,**[2] **Yejin Choi**[3,2]
[1]Stanford University
[2]Allen Institute for Artificial Intelligence
[3]Paul G. Allen School of Computer Science & Engineering, University of Washington
antoineb@cs.stanford.edu, {ronanlb,yejinc}@allenai.org

Abstract

Understanding narratives requires reasoning about implicit world knowledge related to the causes, effects, and states of situations described in text. At the core of this challenge is how to access contextually relevant knowledge on demand and reason over it.

In this paper, we present initial studies toward zero-shot commonsense question answering by formulating the task as inference over dynamically generated commonsense knowledge graphs. In contrast to previous studies for knowledge integration that rely on *retrieval* of existing knowledge from *static* knowledge graphs, our study requires commonsense knowledge integration where contextually relevant knowledge is often *not* present in existing knowledge bases. Therefore, we present a novel approach that *generates* contextually-relevant symbolic knowledge structures on demand using *generative neural commonsense knowledge models*.

Empirical results on two datasets demonstrate the efficacy of our neuro-symbolic approach for dynamically constructing knowledge graphs for reasoning. Our approach achieves significant performance boosts over pretrained language models and vanilla knowledge models, all while providing interpretable reasoning paths for its predictions.

Introduction

Understanding narratives requires reasoning about all the implicit, but trivially inferable, details of a situation based only on what is explicitly stated in text. A statement as simple as “they went to the club” instantly invokes a bank of commonsense expectations: they had to get dressed, they were going dancing, they likely had drinks, and so forth. These reasoning capabilities are missing in most existing neural language understanding models that learn task-specific representations without acquiring rich background knowledge about the social and physical world.

In response, recent work has investigated augmenting deep learning models with retrieval mechanisms over large-scale commonsense knowledge graphs (Mihaylov and Frank 2018; Bauer, Wang, and Bansal 2018; Paul and Frank 2019). However, these approaches assume an entity linking step between the written text and knowledge graph. By canonicalizing entities, they discard key context surrounding the input, and often retrieve semantically irrelevant knowledge (*e.g.*, a *club* being a blunt weapon is irrelevant to the earlier situation).

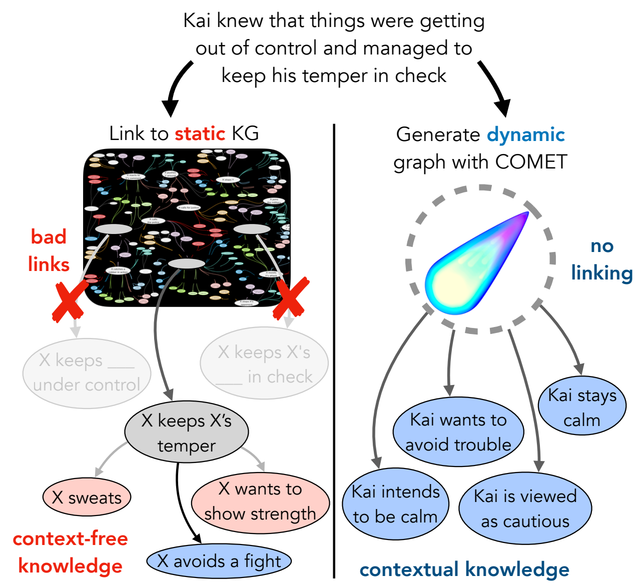


Figure 1: Previous approaches for accessing knowledge link situational contexts to static knowledge graphs. Our work generates knowledge dynamically from neural knowledge models.

In this paper, we propose to *generate* new knowledge that is contextually relevant instead of *retrieving* existing knowledge as is. Bosselut et al. (2019) recently introduced *Commonsense Transformers* (COMET), a new framework for training neural representations of knowledge graphs. This new class of neural *knowledge model* provides a powerful representational tool for connecting commonsense knowledge to downstream task models. Because COMET represents knowledge graphs neurally, it can generate commonsense inferences for any entity that can be encoded by the neural model (*i.e.*, described with language). With no need to canonicalize context entities to link to a static knowledge graph, the knowledge model can be queried directly with complex compositional structures, and even full narrative contexts. This flexibility has led them to be used out-of-the-box in a variety of settings requiring contextual knowledge, such as sarcastic comment generation (Chakrabarty et al. 2020), therapy chatbots (Kearns et al. 2020), and story plot generation (Ammanabrolu et al. 2020).

In this work, we use COMET to dynamically construct context-relevant knowledge graphs that can be reasoned over for commonsense question answering. Given a raw context, COMET generates commonsense inferences that provide world knowledge about the situation depicted in the context. These inferences can be used as additional context to score answer candidates or to generate additional inferences. By generating new inferences and connecting them to the raw context and answers, COMET dynamically constructs a symbolic graph of commonsense knowledge. The raw context is the root node, answer choices are leaf nodes and generated commonsense inferences provide intermediate nodes between them, instantiating different reasoning paths between the context and answers. Using COMET generated scores as factors weighting these paths, we propose new inference algorithms to reason over the generated graph and identify the most likely answers to questions about the situation.

We evaluate our approach in a *zero-shot* setting on the SocialIQa (Sap et al. 2019b) benchmark, a question answering dataset for evaluating social commonsense, and the StoryCS benchmark (Rashkin et al. 2018), a story understanding dataset. Empirical results show that our neuro-symbolic approach, COMET - DynaGen, outperforms purely neural large-scale pretrained language models (Radford et al. 2018, 2019) and knowledge models that evaluate QA examples directly without dynamically generating an intermediate symbolic commonsense knowledge graph (*i.e.*, reasoning with COMET with no inference hops).

Dynamic Knowledge Graph Construction for Question Answering

Our approach uses a knowledge model, COMET (Bosselut et al. 2019), to dynamically construct a context-relevant commonsense knowledge graph about a presented situation. COMET is trained using transfer learning from large-scale pretrained language models (Radford et al. 2018) to knowledge graphs. When trained on the Atomic knowledge graph (Sap et al. 2019a), it learns to generate social commonsense inferences of situations depicted in text. Importantly, unlike static knowledge graphs (*e.g.*, ConceptNet; Speer, Chin, and Havasi 2017), which require canonicalizing input entities to link to the graph, COMET represents knowledge neurally, allowing it to generate commonsense for arbitrary input forms.

In Figure 1, for example, the context "Kai knew things were getting out of control and managed to keep his temper in check" is unlikely to be found in any existing knowledge graph. It describes a very specific situation. However, COMET can parse this full context and generate commonsense knowledge about Kai's reactions and motivations, such as "Kai stays calm" or "Kai wants to avoid trouble," as downstream inferences. We exploit this generalization property of knowledge models to dynamically construct knowledge graphs for presented situations that can be reasoned over to answer commonsense questions about them.

Notation. Formally, we assume a dataset of examples, each with an associated context c describing a situation, a question q asked about that situation, and a set of n possible answers $\mathcal{A} = \{a^0, ..., a^{n-1}\}$ to that question. Each answer is composed of multiple tokens $Y^a = \{y_1, ..., y_{|a|}\}$.

Generating Commonsense Inferences. We generate commonsense inferences for a situational context c by concatenating the context with relation types from the Atomic knowledge graph and using COMET to produce candidates $\mathcal{G}$. Each candidate $g \in \mathcal{G}$ is associated with a score ϕ_g that approximates the model's confidence in the inference:

$$\phi_g = \frac{1}{|g|} \sum_{t=1}^{|g|} \log P(x_t | x_{<t}, c, r) \qquad (1)$$

where x_t are the tokens of g, $|g|$ is the token length of g, r is an arbitrary commonsense relation type for which COMET can generate inferences, and:

$$P(x_t | x_{<t}, c, r) = \mathbb{COMET}(c, r, x_{<t}) \qquad (2)$$

where the tokens of c and r are concatenated with the tokens $x_{<t}$ to be input to COMET. Any generation $g \in \mathcal{G}$ conditioned on c can be seen as a 1-hop commonsense inference of c.

Using a Markov assumption, we can generalize this approach by conditioning on generated commonsense inferences to generate $\mathcal{G}^\ell$, a set of ℓ-hop inferences from c:

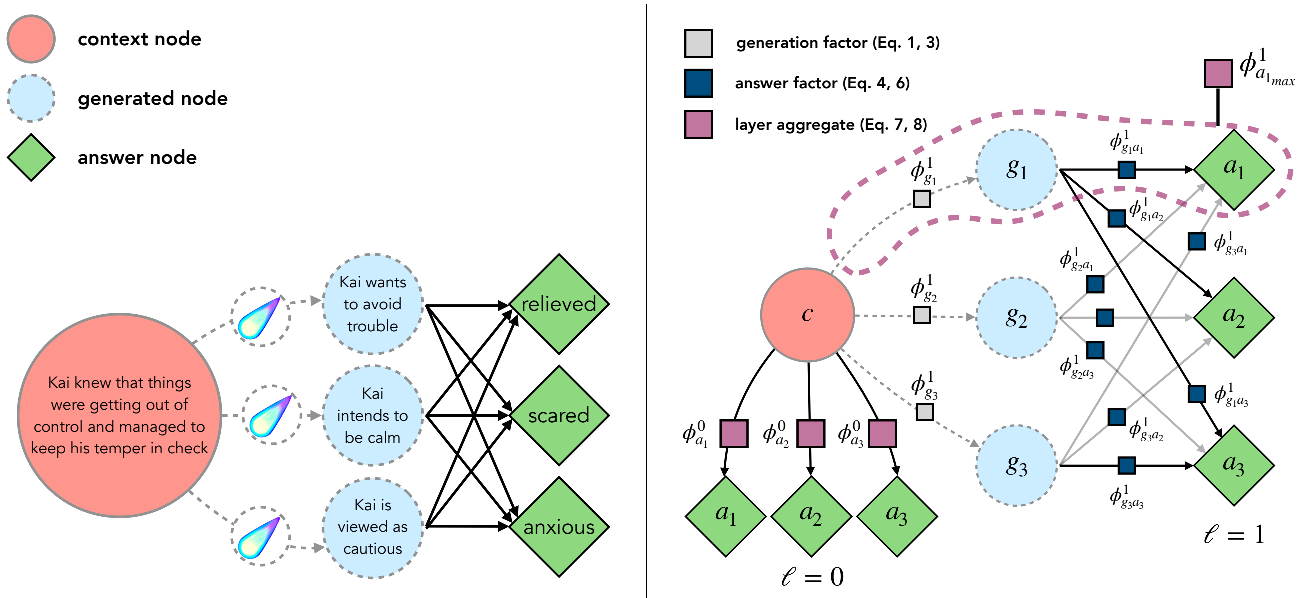
$$\phi_g^\ell = \phi_g^{\ell-1} + \frac{1}{|g^\ell|} \sum_{t=1}^{|g^\ell|} \log P(x_t | x_{<t}, g^{\ell-1}, r) \qquad (3)$$

where ϕ_g^ℓ is a generation score for any $g^\ell \in \mathcal{G}^\ell$, $g^{\ell-1}$ is an arbitrary inference from $\mathcal{G}^{\ell-1}$, the set of inferences of the previous hop, and $\phi_g^{\ell-1}$ is the generation score of that seed inference. Using this approach, we can use COMET to construct a graph where commonsense inferences g are nodes. For an arbitrary node g^ℓ, its parent is the node from the previous level $\mathcal{G}^{\ell-1}$ that COMET conditions on to generate g_ℓ. The children of g_ℓ are nodes generated when COMET conditions on g_ℓ to generate new commonsense inferences. We set $g^0 = c$ because the context is the root node of the graph, and $\phi_g^0 = 0$ because the original context c is deterministic.

Answers as Leaf Nodes. The final step in constructing the knowledge graph is to connect the answer choices $a \in \mathcal{A}$ to the generated commonsense inferences. We initialize a node in the graph for each answer choice a and connect it as a child node to each commonsense inference in the graph: $g \in \mathcal{G}^\ell$ for $\ell \in [0, L)$ where L is the number of levels in the final graph. In Figure 2b, we see that the answer choices $\mathcal{A}$ = {*relieved, scared, anxious*} are connected to the root node and each generated commonsense inference in the $L = 2$ level graph.

Knowledge Graph Reasoning

Being designed as a conditional language model, COMET can also be used to score candidate commonsense inferences. We use this property to score answer



(a) COMET receives the context c and generates new commonsense inferences to construct a local graph of knowledge about the situation (Section).

(b) Our inference algorithms reason over the graph by aggregating commonsense paths to answer questions about the situation (Section).

Figure 2: Our approach consists of dynamically constructing a local commonsense knowledge graph about a presented situation. This graph can be used to reason about the different questions about the situation.

candidates $a \in \mathcal{A}$ conditioned on the generated commonsense inferences $g \in \mathcal{G}$ that are connected to them. The scores from COMET are used to initialize factor nodes between each generated commonsense inference (at all levels of the graph) and each answer choice. Using these scores, and scores between commonsense inferences (Eqs. 1, 3), as a set of factors, our generated knowledge graph implicitly encodes a factor graph that can be reasoned over to evaluate each answer candidate.

Computing Answer Scores

COMET is originally trained to maximize the conditional log-likelihood of the tokens of a target entity e_2 from a knowledge graph tuple (e_1, r, e_2). As a result, the knowledge model can measure the log-likelihood of a candidate entity e_2 given a source entity e_1 and relation r. For a given example, we treat each answer candidate a as an e_2 candidate for COMET, map the parent nodes of a (*e.g.*, g nodes) to be equivalent to e_1, and set the question q as r, allowing COMET to evaluate each answer candidate according to its implicit knowledge representations. For each answer $a \in \mathcal{A}$, we define a factor based on each token's conditional log-likelihood as computed by COMET:

$$\phi_{ga} = \frac{1}{|a|} \sum_{s=1}^{|a|} \log P(y_s | y_{<s}, g, q) \tag{4}$$

where y_s corresponds to the token in a at time step s, $y_{<s}$ is all the tokens preceding y_s in a, and $|a|$ is the total number of tokens making up a. In this way, for any QA example, we define a set of factor nodes ϕ_{ga} connecting the answer candidates $a \in \mathcal{A}$ to the commonsense inferences $g \in \mathcal{G}$ generated by COMET about the situational context c.

Overcoming Answer Priors. Because certain answer candidates have a high probability of occurring for certain questions regardless of the context (*e.g.*, *happy* is a common answer for questions about emotional reactions), we redefine ϕ_{ga} (Eq. 4) in terms of the point-wise mutual information between the commonsense path g and answer a:

$$\phi_{ga} \propto \text{PMI}(a, g|q)$$

$$\phi_{ga} = \frac{1}{|a|} \sum_{s=1}^{|a|} \Big(\log P(y_s | y_{<s}, g, q) - \log P(y_s | y_{<s}, q) \Big) \tag{5}$$

where $\log P(y_s | y_{<s}, q)$ is the log-likelihood of each token in the answer given only the question and previous answer tokens. We describe our approximation of this distribution in Appendix B.

Inference

Each ϕ^ℓ_g scores a unique reasoning path at a particular depth ℓ in the graph. The composition $\gamma_g \phi^\ell_g + \gamma_{ga} \phi^\ell_{ga}$ can then be seen as scoring a path to a particular answer. To find the most likely answer, we marginalize over all paths to the answers at layer ℓ:

$$\phi_a^\ell = f(\{\gamma_g \phi_g^\ell + \gamma_{ga}\phi_{ga}^\ell : g \in \mathcal{G}^\ell\}) \tag{6}$$

where ϕ_g^ℓ (Eq. 3) and ϕ_{ga}^ℓ (Eq. 5) are the *path* and *answer* score, respectively, for generation $g \in \mathcal{G}^\ell$. γ_g and γ_{ga} are hyperparameters balancing the contribution of both scores. Because the path and answer scores are log-probabilities, we set f as the LogSumExp, yielding Eq. 6 as a variable elimination over $g \in \mathcal{G}^\ell$. We also define an extremum estimator over the distribution of generated inferences $\mathcal{G}^\ell$:

$$\phi_{a_{max}}^\ell = \max_{g \in \mathcal{G}^\ell} \gamma_g \phi_g^\ell + \gamma_{ga}\phi_{ga}^\ell \tag{7}$$

At a high level, $\phi_{a_{max}}^\ell$ can be interpreted as approximating the likelihood of answer a given a singular reasoning path: $\{c \rightarrow g^1 \rightarrow \cdots \rightarrow g^\ell \rightarrow a\}$, rather than by computing an aggregation of all paths in the graph to the answer (Eq. 6).

Once the answer scores at different levels in the graph are computed, $\{\phi_a^\ell\}_0^L$, the final score for each answer can be evaluated by averaging over the graph levels $\ell \in [0, L)$:

$$\log P(a|q,c) \propto \phi_a = \sum_{\ell=0}^{L} \beta^\ell \phi_a^\ell \tag{8}$$

$$\hat{a} = \arg\max_{a \in \mathcal{A}} \phi_a \tag{9}$$

where $\hat{a}$ is the selected best answer by the approach, L is the number of generation hops made by the COMET model (*i.e.*, the number of levels in the graph), ϕ_a^ℓ is the score that is propagated from each hop of the constructed knowledge graph, and β^ℓ is hyperparameter scaling the contribution of each hop score. We note that ϕ_a^0 is the result from evaluating the answer candidates directly against the original context c, and that ϕ_a^ℓ is replaced by $\phi_{a_{max}}^\ell$ if the extremum estimator (Eq. 7) is used instead of variable elimination (Eq. 6).

Experimental Setup

We evaluate our approach in a zero-shot experimental setting. It is a well-studied phenomenon that neural methods trained on crowdsourced data often learn to shortcut reasoning to arrive at a correct answer (Gururangan et al. 2018; Li and Gauthier 2017). We use a zero-shot setting to simulate the model having to reason about situations it has never encountered before, forcing it to construct reasoning graphs from explicit knowledge it can generate (*e.g.*, knowledge learned by COMET), and precluding it from learning dataset-specific artifacts. As such, we do not use training data to update model parameters. Furthermore, any result presented on the test set does not have hyperparameters tuned on the development set.

Datasets and Processing

We evaluate our method on two datasets: SocialIQa (Sap et al. 2019b) and StoryCS (Rashkin et al. 2018).

SocialIQa. The SocialIQa dataset evaluates a model's ability to understand the social dynamics underlying situations described in short text snippets. Each example in the dataset consists of a context, a question about that context, and three multiple choice answers. An example from the dataset is shown in Figure 2. We outline pre-processing steps for the data in Appendix A.

StoryCS. The StoryCS dataset consists of short 5-sentence stories with annotated motivations and emotional responses whose labels are drawn from classical theories of psychology (*e.g.*, Plutchik 1980). We map the emotion classification task to a QA task by posing an individual question for each emotion label (*disgust, surprise, fear, anger, trust, anticipation, sadness, joy*) that must be predicted for each example. We outline this procedure in Appendix B.

Experimental Settings

Hyperparameters. We use most of the same hyperparameters to train the COMET model on the Atomic knowledge graph as in Bosselut et al. (2019). However, we use GPT2-345M (Radford et al. 2019) as the pretrained language model that seeds COMET and freeze the position embeddings so we can generalize to longer contexts. We note that the SocialIQa dataset is partially derived from Atomic knowledge base tuples. However, we do not allow Atomic tuples used to seed SocialIQa evaluation examples to be used as training examples for COMET. We provide more details of this splitting in Appendix A. The number of levels in the graph L is set to 2. As we operate in the zero-shot setting, we do not tune hyperparameters. For the SocialIQa dataset, we set $\gamma_g = \gamma_{ga} = 1.0$ and $\beta^\ell = 1.0\ \forall \ell$. For StoryCS, we do the same except that $\gamma_g = 0$. Unless stated otherwise, we use argmax decoding to generate inferences from COMET, and use variable elimination over the graph to select answers.

Prediction. To predict an answer on the SocialIQa dataset, we use Equation 9. Prediction for StoryCS is less straightforward, as the task is originally binary multi-label classification. To make a prediction, we treat ϕ_a (Eq. 8) for each label j independently and select an answer based on whether $\phi_{a,j}$ is above a label-specific threshold, κ^j. To avoid violating the zero-shot setting (*i.e.*, tuning thresholds on the development set), we select the threshold using the score at the percentile of the positive label distribution (*e.g.*, if the *joy* emotion is present for 20% of examples, we set the threshold at the score of the 20th percentile of the CDF). Thresholds are reported in Appendix Table 10 for each label.

SocialIQa Study

Baselines. As baselines in the SocialIQa study, we use large-scale pretrained language models: GPT (Radford et al. 2018), GPT2-117M, GPT2-345M, and GPT2-762M (Radford et al. 2019). To adapt these language models optimally to the QA task, question-answer pairs are automatically converted to a templated form, a process we outline in Appendix B. We also report the results of a model, COMET - Direct, that only uses ϕ_a^0 to select answers (*i.e.*, answers are evaluated with respect to the context with no dynamic graph construction). Additionally, we compare against the Self-Talk model of Shwartz et al. (2020), which queries pretrained language models to generate additional details about a presented situ-

Situation	Most Contributing Paths in Graph	Answers
Jesse drove Ash to the airport and dropped them off at the airport with ease. How would Jesse feel afterwards?	**Jesse wants to go home**	**a)** ***drained*** ✓ b) went to the ticket counter c) dropped me off at the airport
	Jesse wanted to be helpful	a) drained b) *went to the ticket counter* ✗ c) dropped me off at the airport
After jumping off the roof of his house Quinn had trouble breathing. How would you describe Quinn?	**Quinn gets hurt**	**a)** ***foolish*** ✓ b) patient c) light-headed
	Quinn wants to get medical help	a) foolish b) patient c) *light-headed* ✗
Alex took notice of the children who were singing at the playground. What will happen to Alex?	**Alex is happy**	a) hurt the children **b)** ***joy*** ✓ c) tell the children to stop
	Alex wants to go home	a) hurt the children b) joy c) *tell the children to stop* ✗
Taylor was close to winning the game. Taylor ran straight for home plate. What will Taylor want to do next?	**Taylor wants to celebrate**	a) try to get over that they did win **b)** ***celebrate the win*** ✗ c) wanted to score
	Taylor wants to be home	a) try to get over that they did win b) celebrate the win c) *wanted to score* ✓

Table 1: Example contexts, paths, and answers for the COMET - DynaGen model on SocialIQa. We bold the predicted answer and its most contributing path. We *italicize* the most likely answer for each path. Incorrect high-scoring answers for a path are marked with ✗ and correct answers are marked with ✓. We only present a subset of the generated paths.

Model	Dev Acc.	Test Acc.
Random	33.3	33.3
GPT	41.8	41.7
GPT2 - 117M	40.7	41.5
GPT2 - 345M	41.5	42.5
GPT2 - 762M	42.5	42.4
Self-Talk	46.2	43.9
COMET - Direct	48.7	49.0
COMET - DynaGen	**50.1**	**52.6**
BERT-large (sup.)	66.0	63.5
RoBERTa-large (sup.)	**78.1**	**77.0**
Human	86.9	84.4

Table 2: Accuracy on the development and test sets of SocialIQa. COMET - DynaGen is our model.

ation and appends these to the original context. Finally, we report the result of supervised BERT (Devlin et al. 2018) and RoBERTa (Liu et al. 2019) models, and random and human baselines from Sap et al. (2019b).

Overall Performance. We report the main results of our SocialIQa study in Table 2. First, our approach achieves an absolute improvement of $\sim$10.2% over the top performing language model baseline, GPT2-762M, showing the importance of using knowledge models to represent commonsense. Additionally, our approach of dynamically constructing a knowledge graph *on demand* (COMET - DynaGen) performs better than using the knowledge model to directly evaluate answers (COMET - Direct) by $\sim$3.6%, highlighting the value in representing more complex reasoning paths. Finally, the improvement over Self-Talk depicts the benefit of using a structured graphical representation for reasoning compared to one that uses language models to generate additional situational context sentences for conditioning.

We note, however, that the state-of-the-art performance of the supervised BERT and RoBERTa models is significantly higher, meaning there is room for improvement in developing comparable zero-shot approaches to QA. However, one point of interest is that the performance of training BERT with only 5000 training examples (rather than the full 30k) is close (54%) to the performance of COMET - DynaGen, indicating that knowledge models and joint neuro-symbolic solutions are already promising in low-data regimes.

Qualitative Analysis. In Table 1, we present top reasoning paths from the graphs generated by COMET - DynaGen. The strength of our approach can be seen in the first example, where the correct answer, *drained*, is more likely to be a feeling associated with wanting "to go home," a post-condition in the graph generated by COMET - DynaGen. In the original context, this condition is implicit. This benefit to leveraging graph reasoning is also seen in the second example, where Quinn's *foolishness* is linked to "[getting] hurt." We note that COMET - Direct, RoBERTa-large, and GPT2-345M all

Algorithm	# nodes	# edges	ϕ_a^ℓ	$\phi_{a_{max}}^\ell$
Argmax Decoding	10.6	26.4	**50.1**	**49.6**
Beam Search - 5	43.2	156.8	49.5	49.1
Beam Search - 10	83.0	316.2	50.0	49.1
Top-5 sampling	32.0	111.9	49.0	49.0
Top-10 sampling	59.9	223.8	49.3	49.4

Table 3: Development set accuracy for different graph construction techniques. The average number of nodes and edges in the constructed graphs is presented.

answer this question incorrectly, reinforcing the importance of explicit reasoning graphs.

In the final two examples, we present uninteresting or failure cases. In the first, the model predicts that Alex will experience *joy* after reasoning through the path that he will be "happy," which, while correct, is merely leveraging synonymy. In the final example, we show a case where the model selects an incorrect answer by reasoning through an incorrect path. By recognizing that "Taylor wants to celebrate" as a likely post-condition of the context, the model selects an answer that is incorrect. An interesting secondary failure mode in this example is in the second path through the inference "Taylor wants to be home." While this path selects the correct answer, it would not be considered explanatory by humans. In general, we find these cases to be more common in multi-sentence situations. The compositionality of the context makes it more challenging to generate directed inferences, and the factor nodes become less reliable in the graph. We observe that performance on multi-sentence contexts drops by $\sim$5%.

Graph Construction Algorithm. As the quality of the reasoning paths is key to our approach, we investigate the effect of the inference generation algorithm. We evaluate the following generation algorithms: argmax decoding, beam search with beam size $b = 5, 10$ and top-k sampling (Fan, Lewis, and Dauphin 2018; Holtzman et al. 2018) with k = 5, 10. For each decoding method, we generate a graph using every candidate produced by the decoder (*e.g.*, argmax decoding produces 1 candidate, top-10 sampling produces 10).

Our results in Table 3 show that the performance COMET - DynaGen is not dependent on the decoding strategy used to dynamically generate the commonsense knowledge graph. This result is promising as it shows that the reasoning procedure is robust to variability in the candidate generations (larger graphs will be less precise). However, it also shows that the approach has difficulty using richer dynamically-generated commonsense knowledge representations to answer questions correctly. These results point to the need for future work in developing algorithms that can aggregate larger sets of commonsense inference paths.

StoryCS Study

Baselines. As with SocialIQa, we report the results of a random baseline, pretrained language models adapted to the task, and a model that only uses ϕ_a^0 to select answers (COMET - Direct). As supervised comparison models, we report the performance of several BERT-based models from Gaonkar et al. (2020) that are state-of-the-art for the task.

Model	P	R	F1
Zero-shot CDF-weighted	**No Training Data**		
Random	20.6	20.8	20.7
GPT	34.7	36.4	35.5
GPT2 - 117M	30.8	31.8	31.3
GPT2 - 345M	33.3	35.3	34.3
GPT2 - 762M	35.5	37.4	36.4
COMET - Direct	37.4	36.9	37.2
COMET - DynaGen	**38.9**	**39.3**	**39.1**
Supervised			
BERT	**65.6**	56.9	61.0
BERT + LE	63.1	61.7	62.4
BERT + SS	57.9	**76.4**	**65.9**

Table 4: Precision, Recall, F1 on the StoryCS dataset. Best models in different training settings are bolded

Overall Performance. Our results indicate that our zero-shot algorithm, COMET - DynaGen, significantly outperforms other zero-shot baselines such as language models, including models with twice the number of parameters. Importantly, again, we see consistent improvement from dynamically generating a contextual commonsense knowledge graph, rather than directly evaluating the answer choices with COMET - Direct. Our full approach yields higher precision, recall, and F1, than the COMET - Direct baseline.

Qualitative Analysis. We once again see the benefit of generating a reasoning graph in Table 5. COMET - DynaGen is able to select the two correct answers to "How does Daniel feel?" leveraging the path through the commonsense inference that "His Dad is helpful" to predict that Daniel is *trusting*, and the path through the commonsense inference "Daniel wants to try something new" to predict that Daniel is *excited*. However, there is still much room for improvement, as large-scale pretrained language models that are fine-tuned using supervised data perform considerably better on the task.

Few-shot Tuning. To evaluate the quality of our untuned thresholds from Section based on the label distribution threshold of the CDF of the model's scores (CDF-label in Table 6), we also report the results of our approach using different strategies to set thresholds κ. First, we explore the impact of tuning the κ thresholds on varying amounts of the development set data: 4 examples, 10 examples, 20 examples, and 20% of the development data (the same amount used for validation in Rashkin et al. 2018). In each of these settings, we run a study with 5 different randomly selected sets of examples, and report the average performance. We also report the performance of using the 50^{th} percentile score of the CDF as the threshold (CDF-50). In Table 6, we observe large recall gains from these tuning strategies at the expense of precision. However, tuning using merely 10 examples achieves higher F1 than the default strategy, showing the potential of relaxing to a few-shot setting when limited examples are available.

Situation	Most Contributing Paths in Graph	Answers
Daniel was excited to get a remote control boat for his birthday. He asked his dad to drive him to the lake to try it out. *How does Daniel feel?*	His dad is helpful	disgusted, angry, sad, afraid, happy, **trusting** ✓, excited, surprised
	Daniel wants to try something new	disgusted, angry, sad, afraid, happy, trusting , **excited** ✓, surprised

Table 5: Example StoryCS context, high-scoring paths, and answers for our approach. We show which emotions are predicted through which path by bolding them. Correct answers are bolded. As in Table 1, only a subset of paths in the graph generated by COMET - DynaGen are shown. Generated graphs for StoryCS have on average 8.8 nodes and 19.3 edges.

Model	P	R	F1
Zero-shot	**No Training Data**		
CDF-label	**39.5**	39.5	**39.5**
CDF-50	25.9	**75.0**	38.5
Few-shot Tuning			
Tuned from 4 examples	**31.1**	54.6	39.4
Tuned from 10 examples	30.2	64.3	41.0
Tuned from 20 examples	28.6	**73.5**	**41.1**
20% development tuning	31.2	65.1	42.2

Table 6: Development set Precision, Recall, and F1 of emotion prediction on the StoryCS dataset for different strategies for setting prediction thresholds.

Related Work

Question Answering with Knowledge Graphs Previous work has explored integrating reasoning over static knowledge graphs for question answering and story understanding. In general, these approaches extract knowledge tuples from the static KG by linking canonicalized entities to nodes and performing multi-hop inference along relation paths to form full tuples that can be encoded by a downstream neural architecture (Mihaylov and Frank 2018; Bauer, Wang, and Bansal 2018; Weissenborn, Kovcisk'y, and Dyer 2017; Lin et al. 2019; Paul and Frank 2019; Yu et al. 2019). Similar to our approach of discovering reasoning chains between contexts and answers, Paul and Frank (2019) extract reasoning paths in ConceptNet between normalized entities from the context answer candidates, but can only discover paths through nodes in the static knowledge graph. Finally, there exists works that also dynamically construct latent knowledge graphs (Das et al. 2019; Bosselut et al. 2018), but these works presuppose a fixed set of entities that can be KG nodes and then approximate graph edges with neural transformations. In contrast, our algorithm can generate arbitrary nodes, thereby constructing a unique graphical structure for any example.

Multi-hop Reading Comprehension Similar in spirit to reasoning over knowledge graphs for question answering is work in multi-hop reading comprehension. Many datasets for learning to aggregate facts without graph structure have been released in recent years (Weston et al. 2016; Welbl, Stenetorp, and Riedel 2018; Yang et al. 2018; Talmor and Berant 2018). Approaches designed for these resources generally use large-scale neural networks to attend over supporting facts across text (Zhong et al. 2019; Dhingra et al. 2018). Most similar to our work are approaches that construct real-time entity mention graphs as neural reasoning paths (Cao, Aziz, and Titov 2018; Jiang et al. 2019; Jiang and Bansal 2019; Fan et al. 2019). Our approach differs from these models in that we *generate* relevant supporting information rather than mining it from accompanying documents and conduct our study in a zero-shot setting with no additional training.

Automatic Commonsense KG Construction Multi-hop reasoning over commonsense inferences requires construction of knowledge resources and recent approaches have investigated how to mine commonsense knowledge from deep learning models. Sap et al. (2019a) investigated whether LSTM models could generate new tuples for the Atomic knowledge graph. Similarly, Li et al. (2016) and Saito et al. (2018) explored whether neural models could be used to validate proposed knowledge rather than generating it. Jastrzebski et al. (2018) built on these approaches for evaluating novel commonsense knowledge mined from Wikipedia. More recent work mapped commonsense tuples to natural language with templates and used pretrained language models to validate them (Davison, Feldman, and Rush 2019; Petroni et al. 2019). Concurrently, other research has explored using pretrained language models and adapting them as generative knowledge graph constructors (Bosselut et al. 2019; Malaviya et al. 2019). In contrast to these works that augment static knowledge graphs, our approach focuses on constructing knowledge graphs *on demand* to provide context-dependent commonsense for downstream inference.

Conclusion

Our neuro-symbolic approach uses neural representations of large-scale commonsense knowledge graphs (COMET) to generate contextual knowledge graphs *on demand* for zero-shot question answering. Our approach dynamically constructs a knowledge graph of commonsense inferences related to a presented context and uses it to evaluate answer options for a posed question. A novel inference algorithm reasons over the constructed graph to select the most likely answer to a question. Our approach shows promising results at answering questions without training on the end task on two datasets, SocialIQa and StoryCS, outperforming zero-shot pretrained language models. Finally, our analysis indicates that dynamically generating a contextualized commonsense knowledge graph for inference performs better than using vanilla knowledge models (COMET - Direct) to directly answer questions.

Acknowledgments

We thank Maarten Sap, Hannah Rashkin, Vered Shwartz, and Chandra Bhagavatula for helpful feedback. This research was supported in part by NSF (IIS-1524371, IIS-1714566), DARPA under the CwC program through the ARO (W911NF-15-1- 0543), DARPA under the MCS program through NIWC Pacific (N66001-19-2-4031), JD.com, and the Allen Institute for AI (AI2).

References

Ammanabrolu, P.; Cheung, W.; Broniec, W.; and Riedl, M. O. 2020. Automated Storytelling via Causal, Commonsense Plot Ordering. *arXiv preprint arXiv:2009.00829* .

Bauer, L.; Wang, Y.; and Bansal, M. 2018. Commonsense for Generative Multi-Hop Question Answering Tasks. In *EMNLP*.

Bosselut, A.; Levy, O.; Holtzman, A.; Ennis, C.; Fox, D.; and Choi, Y. 2018. Simulating Action Dynamics with Neural Process Networks. In *Proceedings of the 6th International Conference on Learning Representations*.

Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Çelikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Cao, N. D.; Aziz, W.; and Titov, I. 2018. Question Answering by Reasoning Across Documents with Graph Convolutional Networks. In *NAACL-HLT*.

Chakrabarty, T.; Ghosh, D.; Muresan, S.; and Peng, N. 2020. Rˆ3: Reverse, Retrieve, and Rank for Sarcasm Generation with Commonsense Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7976–7986. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.711. URL https://www.aclweb.org/anthology/2020.acl-main.711.

Das, R.; Munkhdalai, T.; Yuan, X.; Trischler, A.; and McCallum, A. 2019. Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. In *Proceedings of the 7th International Conference on Learning Representations*.

Davison, J.; Feldman, J.; and Rush, A. 2019. Commonsense Knowledge Mining from Pretrained Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1173–1178. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1109. URL https://www.aclweb.org/anthology/D19-1109.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Dhingra, B.; Jin, Q.; Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2018. Neural Models for Reasoning over Multiple Mentions Using Coreference. In *NAACL-HLT*.

Fan, A.; Gardent, C.; Braud, C.; and Bordes, A. 2019. Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs. *ArXiv* abs/1910.08435.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-1082. URL https://www.aclweb.org/anthology/P18-1082.

Gaonkar, R.; Kwon, H.; Bastan, M.; Balasubramanian, N.; and Chambers, N. 2020. Modeling Label Semantics for Predicting Emotional Reactions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4687–4692. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.426. URL https://www.aclweb.org/anthology/2020.acl-main.426.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 107–112. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-2017. URL https://www.aclweb.org/anthology/N18-2017.

Holtzman, A.; Buys, J.; Forbes, M.; Bosselut, A.; Golub, D.; and Choi, Y. 2018. Learning to Write with Cooperative Discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1638–1649. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-1152. URL https://www.aclweb.org/anthology/P18-1152.

Jastrzebski, S.; Bahdanau, D.; Hosseini, S.; Noukhovitch, M.; Bengio, Y.; and Cheung, J. 2018. Commonsense mining as knowledge base completion? A study on the impact of novelty. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, 8–16. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/W18-1002. URL https://www.aclweb.org/anthology/W18-1002.

Jiang, Y.; and Bansal, M. 2019. Self-Assembling Modular Networks for Interpretable Multi-Hop Reasoning. In *EMNLP*.

Jiang, Y.; Joshi, N.; Chen, Y.-C.; and Bansal, M. 2019. Explore, Propose, and Assemble: An Interpretable Model for Multi-Hop Reading Comprehension. In *ACL*.

Kearns, W. R.; Kaura, N.; Divina, M.; Vo, C. V.; Si, D.; Ward, T. M.; and Yuwen, W. 2020. A Wizard-of-Oz Interface and Persona-based Methodology for Collecting Health Counseling Dialog. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* .

Li, L.; and Gauthier, J. 2017. Are Distributional Representations Ready for the Real World? Evaluating Word Vectors for Grounded Perceptual Meaning. *ArXiv* abs/1705.11168.

Li, X.; Taheri, A.; Tu, L.; and Gimpel, K. 2016. Commonsense Knowledge Base Completion. In *ACL*, volume 1, 1445–1455.

Lin, B. Y.; Chen, X.; Chen, J.; and Ren, X. 2019. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. *ArXiv* abs/1909.02151.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Malaviya, C.; Bhagavatula, C.; Bosselut, A.; and Choi, Y. 2019. Exploiting Structural and Semantic Context for Commonsense Knowledge Base Completion. *ArXiv* abs/1910.02915.

Mihaylov, T.; and Frank, A. 2018. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In *ACL*.

Paul, D.; and Frank, A. 2019. Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs. *ArXiv* abs/1904.00676.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1250. URL https://www.aclweb.org/anthology/D19-1250.

Plutchik, R. 1980. A general psychoevolutionary theory of emotion. *Theories of emotion* 1(3-31): 4.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pretraining. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf* .

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8): 9.

Rashkin, H.; Bosselut, A.; Sap, M.; Knight, K.; and Choi, Y. 2018. Modeling Naive Psychology of Characters in Simple Commonsense Stories. In *ACL*.

Saito, I.; Nishida, K.; Asano, H.; and Tomita, J. 2018. Commonsense Knowledge Base Completion and Generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 141–150.

Sap, M.; Le Bras, R.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N. A.; and Choi, Y. 2019a. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In *AAAI*.

Sap, M.; Rashkin, H.; Chen, D.; Le Bras, R.; and Choi, Y. 2019b. Social IQA: Commonsense Reasoning about Social Interactions. *ArXiv* abs/1904.09728.

Shwartz, V.; West, P.; Le Bras, R.; Bhagavatula, C.; and Choi, Y. 2020. Unsupervised Commonsense Question Answering with Self-Talk. In *EMNLP*.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Talmor, A.; and Berant, J. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 641–651. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1059. URL https://www.aclweb.org/anthology/N18-1059.

Weissenborn, D.; Kovcisk'y, T.; and Dyer, C. 2017. Dynamic Integration of Background Knowledge in Neural NLU Systems. *CoRR* abs/1706.02596. URL http://arxiv.org/abs/1706.02596.

Welbl, J.; Stenetorp, P.; and Riedel, S. 2018. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. *Transactions of the Association for Computational Linguistics* 6.

Weston, J.; Bordes, A.; Chopra, S.; and Mikolov, T. 2016. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *ICLR*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *EMNLP*.

Yu, W.; Zhou, J.; Yu, W.; Liang, X.; and Xiao, N. 2019. Heterogeneous Graph Learning for Visual Commonsense Reasoning. In *NeurIPS*.

Zhong, V.; Xiong, C.; Keskar, N. S.; and Socher, R. 2019. Coarse-grain Fine-grain Coattention Network for Multi-evidence Question Answering. In *ICLR*.