Dynamic Knowledge Graph Construction for Zero-shot Commonsense Question Answering

Antoine Bosselut **Yejin Choi**
Paul G. Allen School of Computer Science & Engineering
Allen Institute for Artificial Intelligence
Seattle, Washington, USA
{antoineb,yejin}@cs.washington.edu

Abstract

Understanding narratives requires dynamically reasoning about the implicit causes, effects, and states of the situations described in text, which in turn requires understanding rich background knowledge about how the social and physical world works. 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 QA by formulating the task as probabilistic 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 knowledge on demand using *generative neural commonsense knowledge models*.

Empirical results on the SOCIALIQA and STORYCOMMONSENSE datasets in a zero-shot setting demonstrate that using commonsense knowledge models to dynamically construct and reason over knowledge graphs achieves performance boosts over pre-trained language models and using knowledge models to directly evaluate answers.

1 Introduction

Understanding narratives requires reasoning about all the implicit, but trivially inferable details about 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: that they had to get dressed, that they were going dancing, that 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.

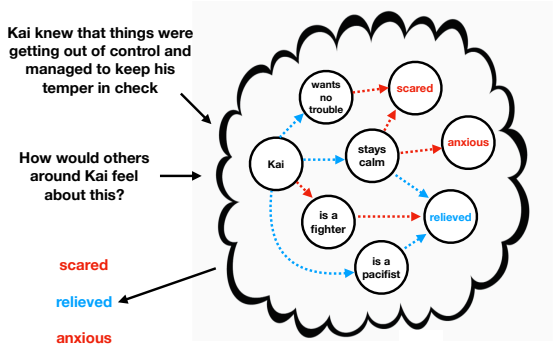

Figure 1: Agents must reason about the commonsense inferences underlying situations depicted in text

In response, recent work has investigated augmenting deep learning models with retrieval mechanisms over large-scale commonsense knowledge graphs (Mihaylov and Frank, 2018; Bauer et al., 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 can often retrieve semantically irrelevant knowledge (e.g., the fact that a club is a blunt weapon is irrelevant to the earlier statement).

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 en-

tity that can be encoded by the neural model. With no need to canonicalize context entities for linking with a static knowledge graph, the knowledge model can be queried directly with complex compositional structures, and even full narrative contexts.

In this work, we use COMET to 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 contexts to score answer candidates or to generate even more inferences. By generating new inferences and connecting them to the raw context and answers, COMET dynamically constructs a knowledge graph of commonsense. The raw context is the root node, answer choices are leaf nodes and generated commonsense inferences provide reasoning paths between the context and answers. Moreover, edges between these nodes are weighted by COMET's knowledge model scores. Using exact probabilistic inference, we can reason over the generated graph to identify the most likely answer to the question about the context.

We evaluate our approach in a zeroshot setting on the SOCIALIQA (Sap et al., 2019b) benchmark, a question answering dataset for evaluating social commonsense, and the STORYCOMMONSENSE benchmark (Rashkin et al., 2018), a story understanding dataset. Empirical results show that the approach outperforms large-scale pretrained language models (Radford et al., 2018, 2019) and COMET models that evaluate QA examples directly without dynamically generating an intermediate commonsense knowledge graph (i.e., reasoning with COMET with no inference hops). Ablations indicate that our approach is agnostic to the decoding algorithm used to generate commonsense inferences for constructing the knowledge graph, but that the pretrained language model used to seed COMET does impact the downstream performance.

2 Neural Representations of Knowledge Graphs for Question Answering

Task Overview Formally, we assume of 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 $\mathcal{A} = \{a^0, ..., a^n\}$ to that question. Each answer is made up of multiple tokens $X^a = \{x_0, ..., x_{|a|}\}$.

COMET is originally trained to generate the tokens of the target entity e_2 from a knowledge base tuple (e_1, r, e_2) given a seed entity e_1 and a relation r. Being architected as a conditional language model, it can also be used to evaluate tuples by aggregating scores for each token in e_2 based on their conditional loglikelihood. In this work, we exploit this property to perform zero-shot evaluation of each answer candidate $a \in \mathcal{A}$. For a given example, we map the context c to be equivalent to e_1 for COMET, the question q to be equivalent to r, and each answer candidate a to e_2. As a result, COMET is able to evaluate each answer candidate for the multiple-choice question according to the implicit knowledge it neurally encodes.

Computing Answer Scores For each answer $a \in \mathcal{A}$, we define a score proportional to its probability of being correct based on each token's conditional loglikelihood as computed by COMET:

$$\phi_a = \frac{1}{|a|} \sum_{t=1}^{|a|} \log P(x_t | x_{<t}, q, c) \quad (1)$$

where x_t corresponds to the token in a at time step t, $x_{<t}$ is all the tokens preceding x_t in a, $|a|$ is the total number of tokens making up a, and:

$$P(x_t | x_{<t}, q, c) = \text{COMET}(c, q, x_{<t}) \quad (2)$$

where the tokens of c and q are concatenated with the tokens $x_{<t}$ to be input to COMET. The most likely answer can then be computed as:

$$\hat{a} = \arg\max_{a \in \mathcal{A}} \phi_a \quad (3)$$

where $\hat{a}$ is the predicted answer from directly evaluating answer candidates using COMET with respect to the context.

3 Dynamic Construction of Intermediate Knowledge Graphs

Apart from evaluating answer candidates, COMET can also be used to generate commonsense inferences about a situation. Being originally trained as a transfer learning engine from large-scale pretrained language models (Radford et al., 2018) to knowledge graphs (Sap et al., 2019a), a trained COMET model can generate social commonsense inferences about any situation by treating a textual context c as an input event to the *neural knowledge graph*. The generated inferences

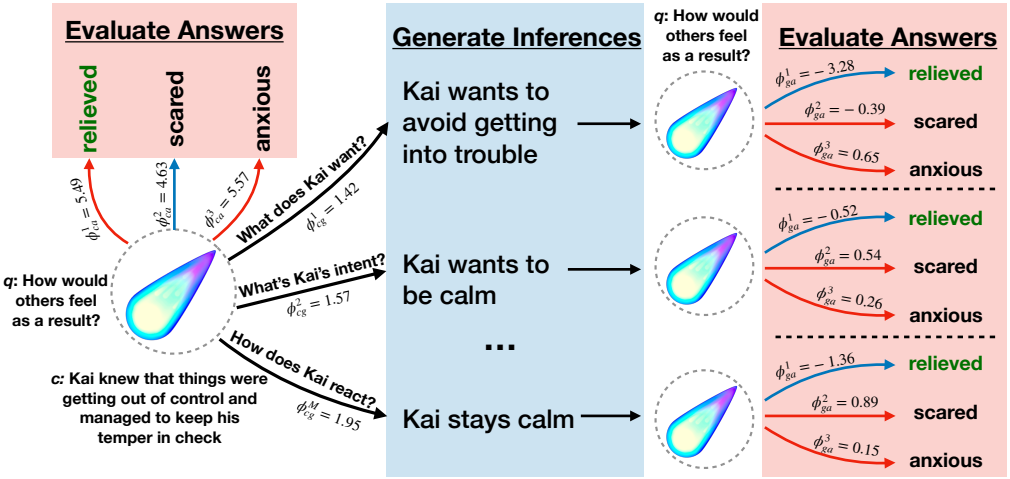

Figure 2: COMET receives the context c and question q. COMET can evaluate answer candidates directly and generate new commonsense inferences related to the context. By generating new inferences, it constructs a graph that can be reasoned over to select the best answer to the question.

can then be used as additional paths for reasoning about the question q and answer set a.

In Figure 2, for example, COMET selects *scared* as the most likely answer to the question given the context. However, after generating intermediate commonsense inferences such as "Kai wants to be calm," the inference algorithm rescores the answers to identify *relieved* as more likely. Below, we outline how intermediate situational inferences can be generated, thereby dynamically building out a knowledge graph of commonsense related to the context, and how COMET can then score answers relative to its own inferences.

3.1 Building the Graph

Generating COMET Inferences We generate commonsense inferences about 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:

$$\phi_g = \frac{1}{|g|} \sum_{s=1}^{|g|} \log P(y_s | y_{<s}, c, r) \quad (4)$$

where y_s are the tokens of g, $|g|$ is the token length of g and r is an arbitrary commonsense relation type for which COMET can generate inferences. Any generation $g \in \mathcal{G}$ that was conditioned on c can be seen as a 1-hop 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:

$$\phi_g^\ell = \phi_g^{\ell-1} + \frac{1}{|g^\ell|} \sum_{s=1}^{|g^\ell|} \log P(y_s | y_{<s}, g^{\ell-1}, r) \quad (5)$$

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. We set $g^0 = c$ and $\phi_g^0 = 0$ because the original context c is not probabilistic.

Computing Answer Scores Without loss of generality across ℓ, for any generation $g \in \mathcal{G}^\ell$, we can use Equation 1 to compute a score ϕ_a for each answer. Rather than using the example's original context as c, we condition on g and the question q to compute ϕ_a for each answer with COMET as an evaluator:

$$\phi_a = \frac{1}{|a|} \sum_{t=1}^{|a|} \log P(x_t | x_{<t}, q, g) \quad (6)$$

where x_t are the tokens in answer $a \in \mathcal{A}$ and $|a|$ is the length of a. Once ϕ_a is computed for answer a for every generated inference $g \in \mathcal{G}^\ell$, we can marginalize over inference scores:

$$\phi_{ga}^\ell = \frac{1}{|\mathcal{G}^\ell|} \sum_{m=1}^{|\mathcal{G}^\ell|} \gamma_g \phi_g^{m,\ell} + \gamma_a \phi_a^{m,\ell} \quad (7)$$

where $|\mathcal{G}^\ell|$ is the number of ℓ-hop inferences generated by COMET from generations at the previous level ($\mathcal{G}^{\ell-1}$), and $\phi_g^{m,\ell}$ (Eq. 5) and $\phi_a^{m,\ell}$ (Eq. 6) are the *path* and *answer* score, respectively, for generation $g_m^\ell \in \mathcal{G}^\ell$. γ_g and γ_a are hyperparameters balancing the contribution of both scores. At a high level, ϕ_{ga}^ℓ can be interpreted as approximating the likelihood of answer a given the reasoning path of $\{c \rightarrow g^1 \rightarrow \cdots \rightarrow g^\ell\}$.

As the number of levels L grows and the number generated inferences $|\mathcal{G}^\ell|$ grows for each level $\ell \in (0, L)$, it becomes intractable to compute ϕ_{ga} by marginalizing over all generated inferences, so we define a maximum likelihood estimator over the distribution of generated inferences $\mathcal{G}$:

$$\phi_{ga_{max}}^\ell = \max_{m \in [0, |\mathcal{G}^\ell|)} \gamma_g \phi_g^{m,\ell} + \gamma_a \phi_a^{m,\ell} \quad (8)$$

where $[0, |\mathcal{G}^\ell|)$ is the range of possible $g_m^\ell \in \mathcal{G}^\ell$ that can be indexed.

3.2 Evaluating the Graph

Probabilistic Reasoning Once the answer scores with respect to different levels in the graph are computed $\{\phi_{ga}^\ell\}_0^L$, the final score for each answer can be evaluated by marginalizing over the graph levels $\ell \in [0, L)$ and selecting the answer with the highest score:

$$\log P(a|q,c) \propto \phi_{ens} = \sum_{\ell=0}^{L} \beta_{ga}^\ell \phi_{ga}^\ell \quad (9)$$

$$\hat{a} = \arg\max_{a \in \mathcal{A}} \phi_{ens} \quad (10)$$

where L is the number of generation hops made by the COMET model (i.e., the number of levels in the graph), ϕ_{ga}^ℓ is the score that is propagated from each hop of the constructed knowledge graph, and β_{ga}^ℓ is hyperparameter scaling the contribution of each hop score. We note that ϕ_{ga}^0 is the result from evaluating the answer candidates directly against the original context c in Equation 1.

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_a (Eq. 6) in terms of the point-wise mutual information between the inference g and answer a:

$$\phi_a \propto \text{PMI}(a, g|q) \quad (11)$$

$$\phi_a = \frac{1}{|a|} \sum_{t=1}^{|a|} \Big(\log P(x_t|x_{<t}, q, g) - \log P(x_t|x_{<t}, q) \Big) \quad (12)$$

where $\log P(x_t|x_{<t}, q)$ is the conditional loglikelihood of each token in the answer given only the question and previous answer tokens.

4 Experimental Setup

We evaluate our method on two datasets: SOCIALIQA(Sap et al., 2019b), a social commonsense question answering dataset, and STORYCOMMONSENSE (Rashkin et al., 2018), a text classification dataset for identifying the motivations and emotions of characters in short stories. For each dataset, we use the development set and test set to report performance. Because our experiments are done in a zero-shot setting, we do not use training sets to update any model parameters. Furthermore, any result presented on the test set does not have hyperparameters tuned on the development set. As additional analysis, we show ablations with scores from the development sets of these datasets.

4.1 Datasets and Processing

SOCIALIQA The SOCIALIQA dataset evaluates a model's ability to understand the social dynamics underlying situations in short text snippets. Each example in the dataset consists of a context, a question about that context, and three candidate answers to the question. An example from the dataset is shown in Figure 2. When converting generated inferences to contexts for answer scoring (Eq. 6), we add a prefix that is specific to the inference type to the generated tokens (e.g., happy ⇒ Person is happy). The prefixes for each inference type can be found in Table 6 in Appendix A. Additionally, we prune the list of generated inferences that can be used for reasoning with rules that are presented in Appendix A. For our main models and ablations, names that appear in contexts and answers are anonymized.

STORYCOMMONSENSE The STORYCOMMONSENSE dataset consists of short 5-sentence stories with annotated motivations and emotional responses whose labels are drawn from classical theories of psychology: Plutchik's Wheel (Plutchik, 1980), Maslow's Hierarchy of Needs (Maslow,

Dataset	# dev	# test
SOCIALIQA	1952	2217
STORYCOMMONSENSE	202360	182160

Table 1: Dataset statistics for SOCIALIQA and STORYCOMMONSENSE

1943), Reiss Motives (Reiss, 2004). In this work, we use the Plutchik classification task to evaluate our method. We map the classification task to a question answering task by posing an individual question for each emotion (*disgust, surprise, fear, anger, trust, anticipation, sadness, joy*) that must be predicted for each example. As the answer text to score, we use formulations of the words that make up the classification label (e.g., *disgusted, surprised, afraid, angry, trusting, excited, sad, happy*). The total number of QA pairs extracted is outlined in Table 1. As question representations q to give to COMET, we use the relations from ATOMIC (Sap et al., 2019a), the knowledge graph on which COMET is trained, that correspond to reactions to events: `xReact` and `oReact`. When computing ϕ_a (Eq. 12, 6), we compute a score for $q =$ `xReact`, `oReact` and average them. Rules for pruning inferences are presented in Appendix A.

4.2 Experiment 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 than in ATOMIC. The number of levels in the graph L is set to 1, meaning we only do 1-hop of commonsense inferences. As we operate in the zero-shot setting, we do not tune inference hyperparameters. For the SOCIALIQA dataset, we set $\gamma_g = \gamma_a = 1.0$ and $\beta^\ell = 1.0\ \forall \ell$. For STORYCOMMONSENSE, we use the same hyperparameters except that $\gamma_g = 0$. Unless stated otherwise, we use argmax decoding to generate commonsense inferences from COMET.

Predictions To predict an answer on the SOCIALIQA dataset, we use Equation 10. Making predictions for STORYCOMMONSENSE is more complicated as the task is originally a binary classification task for eight different dimensions (i.e., emotional responses). To make a prediction, we treat ϕ_{ens} (Eq. 9) for each dimension independently and select an answer based on whether ϕ_{ens} is above a dimension-specific threshold:

$$\hat{a} = \mathbb{1}[\phi^n_{ens} > \kappa^n] \tag{13}$$

where ϕ^n_{ens} is the score from the model for dimension n and κ^n is the threshold for that dimension. Normally, we could tune these decision thresholds on the validation set, but this would violate the zero-shot setting. Instead, decision thresholds are chosen by producing a cumulative distribution function over model scores for the validation examples. Then, we select the threshold as the score at the percentile of the positive label distribution (i.e., if the *joy* emotion is present for 20% of examples, we set the score at the 20th percentile of the CDF as the threshold). Thresholds are reported in Table 7 of Appendix A for each label.

5 Experimental Results

5.1 SOCIALIQA

Baselines As baselines in the SOCIALIQA study, we use state-of-the-art 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. For example, a question such as "How does Alice feel after?" will be replaced by the template “Alice feels". Answers are appended to the end of the template, which is then concatenated to the context, and the language models score the answer words conditioned on the context and template. Table 9 in Appendix A provides the template for each question variety. We also report the results of a model that solely uses ϕ^0_{ga} to select answers (i.e., answers are evaluated directly against the context with no dynamic graph construction) and call this model COMET - CA. Finally, for comparison, we report the result of the random, supervised BERT (Devlin et al., 2018), and human baselines from Sap et al. (2019b).

Ablations We evaluate two ablation types. First, we investigate whether the algorithm for generating commonsense inferences from COMET affects the utility of the generated facts. We present the effect of the following candidate generation schemes: argmax greedy decoding, beam search with beam size $b = 5, 10$ and top-k sampling (Fan et al., 2018;

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
COMET - CA	48.7	49.0
COMET - CGA	**49.6**	**51.9**
BERT - Large (sup.)	66.0	66.4
Human	86.9	84.4

Table 2: Results on the development and test sets of SOCIALIQA. COMET - CGA is our model.

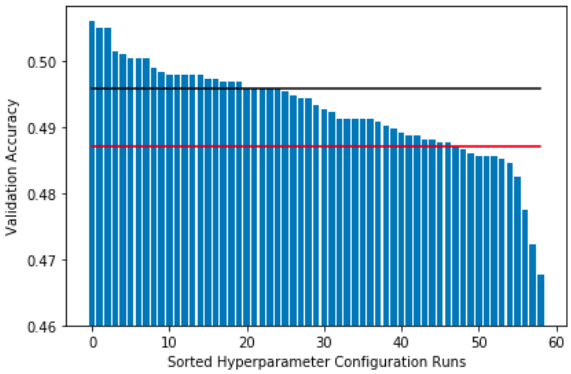

Figure 3: SOCIALIQAdevelopment set performance across different hyperparameter settings

Holtzman et al., 2018) with k = 5, 10 . For each decoding method, we provide the inference algorithm with every candidate produced by each strategy (e.g., argmax decoding produces a single candidate, top-10 sampling produces 10 candidates). Second, we test the effect the pretrained language model used to seed COMET. We train additional versions of the neural knowledge base from GPT, GPT2-117M, and GPT2-762M.

Overall performance We report the main results of our SOCIALIQA study in Table 2. Our model achieves an absolute improvement of 8.4% over the top performing language model baseline, showing the effectiveness of using a neural commonsense knowledge base. Additionally, our approach of dynamically constructing a knowledge graph *on demand* (COMET - CGA) performs better than using the neural knowledge base to directly evaluate answers (COMET - CA) by $\sim$ 3%. Figure 2 shows an example demonstrating how the knowledge graph can help re-score answer options.

We note, however, that the state-of-the-art performance of the supervised BERT model is 66.4%, indicating there is still much room for improvement in using neural knowledge bases for zero-shot question answering. One point of interest is that the performance of training the BERT model with only 5000 training examples (rather than the full 30k) is close (54%) to COMET - CGA, indicating that neural knowledge bases and joint neural-symbolic solutions are promising for studies in low-data regimes.

Tuning effects To evaluate how our results would have varied if we had tuned hyperparameters, we vary β_{ga}^{ℓ} $\forall \ell \in L$ by increments of 0.1 between 0 and 1 and report the results in Figure 3. Regardless of the values of these hyperparameters, COMET - CGA was superior to the pretrained language models in every configuration, and better than COMET - CA (depicted by red line) 80% of the time. The black line indicates the performance of the purely zero-shot approach, which is one of the better performing configurations, though better results are possible by varying these values. The best performing configurations often have $\beta_{ga}^{1} \sim 1.5\beta_{ga}^{0}$, highlighting the importance of the constructed graph.

Decoding Algorithm	CGA_{max}	CGA
Argmax Decoding	**49.6**	49.6
Beam Search - 5	49.1	**49.9**
Beam Search - 10	49.1	49.6
Top-5 sampling	49.0	49.7
Top-10 sampling	49.4	49.8

Table 3: Effect of the decoding algorithm for generating commonsense inferences from COMET. All results on SOCIALIQA development set.

Effect of decoding strategy Our results in Table 3 show that the performance of the graph construction algorithm is agnostic to the decoding strategy used to generate commonsense knowledge. This result is promising as it shows that the reasoning procedure is robust to variability in the candidate generations. However, it also depicts the weakness of using exact inference or maximum likelihood estimation for reasoning over the graph, as these approaches are not capable of leveraging larger candidate sets of commonsense knowledge to answer questions correctly. These results point to the need for future work in developing algorithms that can leverage diverse commonsense inference paths for reasoning over dynamically con-

Language Model	CA Acc.	CGA Acc.
GPT	**49.1**	49.1
GPT2 - 117M	44.9	46.6
GPT2 - 345M	48.7	**49.6**
GPT2 - 762M	41.1	42.7

Table 4: Effect of pretrained language model used to seed COMET. CGA Acc. is for exact inference (Eq. 7).

structed knowledge graphs.

Effect of COMET language model COMET is sensitive to the pretrained language model on which it is trained. In Table 4, we report the performance of training COMET on different seed language models. We see that the COMET model trained on OpenAI GPT (Radford et al., 2018) has the best performance when directly answering the questions without generating an intermediate knowledge graph. However, training COMET on GPT2-345M (Radford et al., 2019) yields better results when an intermediate knowledge graph is constructed. Most importantly, however, regardless of which language model seeds COMET, generating commonsense inferences for reasoning always produces superior performance on SOCIALIQA compared to answering questions directly. Interestingly, there is a drastic performance drop when training on COMET on GPT2-762M despite it being one of the better language models to evaluate directly on SOCIALIQA (as seen in Table 2). This result implies that the learned representations of GPT2-762M may not be as directly transferable as smaller versions of the model, though further analysis would be needed to confirm this hypothesis.

5.2 STORYCOMMONSENSE

Baselines As baselines, we report the performance of several models from Rashkin et al. (2018). Similar to our setup, these baselines can only access the sentence of the story for which the emotion classification must be made. These models use TF-IDF features, Glove embeddings (Pennington et al., 2014), LSTMs (Hochreiter and Schmidhuber, 1997), or CNNs (Kim, 2014), respectively, to encode the sentence. For each baseline, a multi-label linear classifier separately predicts each emotion label from this joint representation. As with SOCIALIQA, we also report the results of a baseline that only uses ϕ^0_{ga} to select answers (COMET - CA). Finally, we report the performance of a supervised GPT model that has access to preceding context sentences.

Model	P	R	F1
Random	10.4	50.0	17.2
Random (weighted)	12.3	11.8	12.0
Zero-shot	**No Training Data**		
COMET - CA	18.6	17.9	18.2
COMET - CGA	**19.9**	**18.8**	**19.3**
Tuned Hyperparameters	**No Training Data**		
COMET - CA	16.2	**60.3**	25.5
COMET - CGA	**18.6**	52.5	**27.5**
Supervised			
TF-IDF	20.1	24.1	21.9
Glove	15.2	30.6	20.3
LSTM	20.3	30.4	24.3
CNN	21.2	23.4	22.2
GPT	**41.6**	**50.2**	**45.5**

Table 5: Precision, Recall, and F1 of Plutchik emotion prediction on the STORYCOMMONSENSE dataset. The best model within the zero-shot, tuned, and supervised sections is **bolded**

Results Our results indicate that our zero-shot algorithm approaches the performance of the weaker supervised baselines for this task. This result is promising as no additional training is used to adapt the COMET model to a classification task. Instead, we use the learned neural knowledge graph to score the likelihood of tokens corresponding to emotional reactions for characters in the story. Importantly, once again, we see consistent improvement from dynamically generating a contextualized commonsense knowledge graph of facts rather than directly evaluating the answer choices with COMET. Our full approach yields higher precision, recall, F1, and accuracy than the COMET - CA baseline.

To evaluate the quality of our untuned thresholds from Section 4.2 based on the CDF of the model's scores, we also report the results of our approach if we are allowed to tune the κ thresholds on 20% of the development data (the same amount used for validation in Rashkin et al. (2018)). We see large gains in Recall from this process, causing our performance to exceed most supervised models. There is still much improvement that can be made, however, as new large-scale transformer models that are trained in a supervised manner are considerably better on this task.

6 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 et al., 2018; Weissenborn et al., 2017; Lin et al., 2019; Paul and Frank, 2019). Similar to our approach of discovering reasoning chains between contexts and answers, Paul and Frank (2019) extract multi-hop 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 work 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 dynamically constructing a unique symbolic structure for any example that can be reasoned over with probabilistic inference.

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 multiple facts without graph structure have been released in recent years (Weston et al., 2016; Welbl et al., 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 et al., 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 commonsense knowledge graphs and recent approaches have investigated how to mine commonsense knowledge from deep learning models. Work by 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 ConceptNet-style knowledge tuples rather than generating new ones. Jastrzębski et al. (2018) built on these approaches for evaluating novel commonsense knowledge mined from Wikipedia. More recent work mapped candidate commonsense tuples to natural language with templates and used pretrained transformer language models to validate them (Davison et al., 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 contextualized knowledge graphs *on demand* to provide context-dependent commonsense for downstream inference.

7 Conclusion

We use neural representations of large-scale commonsense knowledge graphs (COMET) to generate contextualized 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 conditions on it to evaluate answer options for a posed question. We use probabilistic inference to reason over the constructed graph to select the most likely answer to a question. Our approach exceeds the performance of large-scale pretrained language models at the zero-shot setting by 8.5% on the SOCIALIQA dataset. Furthermore, on both the SOCIALIQA and STORYCOMMONSENSE datasets, dynamically generating a contextualized commonsense knowledge graph performs better than using COMET to directly answer questions.

Acknowledgments

We thank Maarten Sap and Hannah Rashkin 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), and the Allen Institute for AI (AI2).

References

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *EMNLP*.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *Proceedings of the 6th International Conference on Learning Representations*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2018. Question answering by reasoning across documents with graph convolutional networks. In *NAACL-HLT*.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2019. Building dynamic knowledge graphs from text using machine reading comprehension. In *Proceedings of the 7th International Conference on Learning Representations*.

Joe Davison, Joshua Feldman, and Alexander Rush. 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)*, pages 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. In *NAACL-HLT*.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *ArXiv*, abs/1910.08435.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Stanislaw Jastrzębski, Dzmitry Bahdanau, Seyedarian Hosseini, Michael Noukhovitch, Yoshua Bengio, and Jackie Cheung. 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*, pages 8–16, New Orleans, Louisiana. Association for Computational Linguistics.

Yichen Jiang and Mohit Bansal. 2019. Self-assembling modular networks for interpretable multi-hop reasoning. In *EMNLP*.

Yichen Jiang, N. Joshi, Yen-Chun Chen, and Mohit Bansal. 2019. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. In *ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *ACL*, volume 1, pages 1445–1455.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *ArXiv*, abs/1909.02151.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2019. Exploiting structural and semantic context for commonsense knowledge base completion. *ArXiv*, abs/1910.02915.

Abraham H Maslow. 1943. A theory of human motivation. *Psychol. Rev.*, 50(4):370.

Tzvetan Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *ACL*.

Debjit Paul and Anette Frank. 2019. Ranking and selecting multi-hop knowledge paths to better predict human needs. *ArXiv*, abs/1904.00676.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 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)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Unpublished manuscript.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. In *ACL*.

Steven Reiss. 2004. Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Rev. Gen. Psychol.*, 8(3):179.

Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2018. Commonsense knowledge base completion and generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 141–150.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social iqa: Commonsense reasoning about social interactions. *ArXiv*, abs/1904.09728.

Alon Talmor and Jonathan Berant. 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)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Dirk Weissenborn, Tom'avs Kovcisk'y, and Chris Dyer. 2017. Dynamic integration of background knowledge in neural nlu systems. *CoRR*, abs/1706.02596.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *ICLR*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Victor Zhong, Caiming Xiong, Nitish Shirish Keskar, and Richard Socher. 2019. Coarse-grain fine-grain coattention network for multi-evidence question answering. In *ICLR*.

A Additional Experimental Settings

Rules for pruning generation sets

1. Any generation that is "none" is pruned
2. Any generation that is identical to a previous generation from the same inputs, but has added punctuation is pruned (e.g., to go to the mall vs. to go to the mall.)
3. Any generation for the following relations that mentions "PersonY" is removed: `oEffect, oReact, oWant`. These generations are untrustworthy as they are often impossible to resolve with an actual person in the context
4. Any generation for the following relations that does not have a token that is a verb is removed: `xEffect, oEffect`
5. In multiple candidate settings, if one of the candidates is "none," we prune all candidates with less likely scores
6. For the STORYCOMMONSENSE dataset, based on an inductive prior, we only generate inferences along the following ATOMIC relations: `xReact`, `oReact`, `xEffect`, `oEffect`, `xIntent`. The logic for pruning `xWant`, `oWant`, `xNeed`, `xAttr` inferences is that emotional reactions for these dimensions could be irrelevant to the context. For example, the emotional reaction to *getting into a car accident* is different from *needing to own a car* to do this. Emotional reactions to the kept relations are more likely to be fidelitous to the original context.

Relation	Prefix
xWant	PersonX wants
xReact	PersonX is
xNeed	PersonX needs
xIntent	PersonX wants
xAttr	PersonX is
xEffect	PersonX
oReact	PersonX is
oEffect	PersonX
oWant	PersonX wants

Table 6: Prefixes appended to COMET produced commonsense inferences for the evaluation step (Eq. 6)

Dimension	CA κ	CGA κ
disgust	0.3751	2.0287
surprise	-0.0329	1.2635
fear	0.1224	1.3517
anger	0.0041	1.1286
trust	-0.0144	1.1514
anticipation	0.0619	1.9843
sadness	0.0126	1.3311
joy	0.0210	1.2462

Table 7: Percentile thresholds κ for predicting an emotion for the COMET - CA and COMET - CGA models. Thresholds for COMET - CGA tend to be higher because the score is the sum of all graph level scores

Dimension	CA κ	CGA κ
disgust	0.1	0.1
surprise	1.0	2.1
fear	0.3	0.5
anger	0.2	0.3
trust	0.2	0.5
anticipation	1.4	2.7
sadness	0.2	0.1
joy	0.6	1.1

Table 8: Tuned thresholds κ for predicting an emotion for the COMET - CA and COMET - CGA models. κ is varied between 0 and 3 in increments of 0.1

Question	Template
What will happen to Others?	The effect on others will be ___
How would Others feel as a result?	Others feel ___
What will Others want to do next?	After, others will want to ___
How would you describe CHARACTER?	CHARACTER is ___
What will happen to CHARACTER?	The effect on CHARACTER will be ___
What does CHARACTER need to do before this?	Before, CHARACTER needs to ___
Why did CHARACTER do this?	CHARACTER did this because ___
How would CHARACTER feel afterwards?	CHARACTER feels ___
What will CHARACTER want to do next?	After, CHARACTER will want to ___

Table 9: Templates used to convert question answering pairs from SOCIALIQA to a format that can be evaluated by the baseline pretrained language models: GPT, GPT2-117M, GPT2-345M, and GPT2-762M.

Relation	Description	Example Completion:
		Event: Person X puts Person X's trust in Person Y
oEffect	The effect the event has on others besides Person X	is considered trustworthy is believed gains Person X's loyalty
oReact	The reaction of others besides Person X to the event	trusted honored trustworthy
oWant	What others besides Person X may want to do after the event	work with Person X partner with Person X to help Person X
xAttr	How Person X might be described given their part in the event	faithful hopeful trusting
xEffect	The effect that the event would have on Person X	gets relieved stays faithful Is betrayed
xIntent	The reason why X would cause the event	to be trusting his or her help/guidance/advice to be friends
xNeed	What Person X might need to do before the event	to be friends with Person Y to have heard a lot of good things about Person Y to get to know Person Y
xReact	The reaction that Person X would have to the event	trusting safe, not alone understood
xWant	What Person X may want to do after the event	to rely on Person Y to go into business with Person Y to make sure that their heart feeling is right

Table 10: Definitions of the relations in ATOMIC. Events in ATOMIC center around the personal situations of a central figure, Person X, with potentially more participants.