CoSe-Co: TEXT CONDITIONED GENERATIVE COMMONSENSE CONTEXTUALIZER

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-trained Language Models (PTLMs) have been shown to perform well on natural language tasks. Many prior works have attempted to leverage structured commonsense present in the form of entities linked through labeled relations in Knowledge Graphs (KGs) to assist PTLMs. Retrieval approaches use KG as a separate static module which limits coverage since KGs contain finite knowledge. Generative methods train PTLMs on KG triples to improve the scale at which knowledge can be obtained. However, training on symbolic KG entities limits their application in tasks involving natural language text where they ignore overall context. To mitigate this, we propose a task agnostic **Co**mmon**Se**nse **Co**ntextualizer (CoSe-Co) conditioned on sentences as input to make it generically usable in NLP tasks for generating contextually relevant knowledge. We propose a novel dataset comprising of sentence and commonsense path pairs to train CoSe-Co. The knowledge paths inferred by CoSe-Co are diverse, relevant and contain novel entities not present in the underlying KG. Additionally, we show CoSe-Co can be used for KG completion. We augment generated knowledge in Multi-Choice QA and Open-ended CommonSense Reasoning tasks leading to improvements over current best methods (upto $\sim$3% and $\sim$7% respectively) on CSQA, ARC, QASC and OBQA datasets. Further, improved performance is seen in low training data regimes which shows CoSe-Co knowledge helps in generalising better.

1 INTRODUCTION

Common sense, as defined by the Merriam-Webster dictionary, is "*sound and prudent judgment based on a simple perception of the situation or facts.*" While dealing with natural language text, common sense allows humans to expand salient concepts and infer additional information. For example, just by reading a sign like *Men at Work* on a road, we implicitly know to slow down our vehicles, look carefully for workers, etc. This implicit process of retrieving acquired common sense knowledge to reason and make logical inferences is critical to natural language understanding (Xie & Pu, 2021). A natural question to ask then is how we can incorporate common sense in now-ubiquitous language models (LMs) (Devlin et al., 2019; Radford et al., 2018b; Raffel et al., 2019).

There have been various efforts (Bao et al., 2016; Feng et al., 2020; Wang et al., 2020b) to leverage structured knowledge present in commonsense knowledge graphs (KGs)[1] (Xie & Pu, 2021) such as ConceptNet (Speer et al., 2017). Most of such works have primarily focused on either of two aspects - (i) training a LM over a KG to generate knowledge about entities extracted from input text, or (ii) solving a particular downstream task by retrieving knowledge from a KG. Generative methods learn commonsense knowledge through training on symbolic entities and relations between them in a KG. Such methods have either been designed for KG completion (Bosselut et al., 2019), i.e. generate tail entity of a KG triple given head entity and relation, or to generate commonsense paths connecting salient entities extracted from text ignoring overall context of the sentence (Wang et al., 2020b). Hence, applying such methods is sub-optimal since most NLP tasks comprise of sentences. Further, being trained on entities as input, applying them directly on sentences is infeasible and lead to train-inference input type mismatch. On the other hand, retrieval methods rely heavily on the structure of a downstream task like multi-choice question answering (QA) to leverage static knowledge in a KG (Yasunaga et al., 2021) and hence, are not applicable beyond a specific task.

[1] We use KG as a shorthand for Commonsense Knowledge Graph.

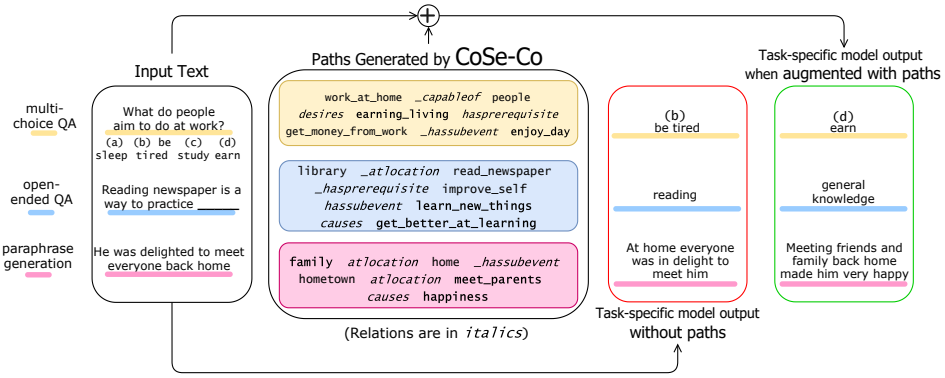


Figure 1: Commonsense knowledge paths generated by our task agnostic **CoSe-Co**. The generated knowledge can be augmented in downstream tasks to improve performance. For interpreting purposes, 'library _atlocation read_newspaper' is equivalent to 'read_newspaper atlocation library'

To address these limitations, we propose **Co**mmon**Se**nse **Co**ntextualizer - **CoSe-Co**, a generative framework which generates relevant commonsense knowledge given any natural language sentence as input. We condition it on sentences to make it generically applicable and enable it to dynamically select entities/phrases from an input sentence as well as generate novel yet relevant entities in the inferences being generated. Figure 1 shows how CoSe-Co could be used to generate commonsense inferences for a variety of tasks. We consider commonsense knowledge in the form of paths, i.e., sequence of entities connected through relations. Relation vocabulary is determined based on schema of KG used for training CoSe-Co. We first create sentence-path paired dataset by - 1) sampling paths from an underlying KG; 2) sampling a subset of entities from a path; and 3) retrieving & filtering sentences (from a sentence corpus) that are semantically similar to the path. The paired data is then used to train a generative language model to generate a path given a sentence as input. CoSe-Co is, thus, task agnostic and can be used directly without any pre-requisite steps.

To demonstrate the usefulness of generated commonsense paths, we augment them in various downstream tasks. The reasoning ability of NLP systems is commonly analysed using QA. Hence, we choose two such tasks: 1) **Multi-Choice QA** on the CSQA dataset (Talmor et al., 2019), where given a question and set of choices, the model has to identify the most appropriate choice. However, often more than one choice is a suitable answer. To mitigate this, 2) **OpenCSR** (Open-ended CommonSense Reasoning) (Lin et al., 2021a) was proposed, where each question is labeled with a set of answers which have to be generated without choices. We perform OpenCSR on ARC, QASC and OBQA datasets. We also show applicability of CoSe-Co in improving performance on paraphrase generation task (§4.5). Our contributions can be summarised as:

1. We propose a **Co**mmon**Se**nse **Co**ntextualizer trained on commonsense paths extracted from a KG to generate knowledge relevant to a given natural language text. CoSe-Co is conditioned on sentence as input to make it task agnostic and generically usable in NLP tasks.
2. We devise a method to create novel sentence-commonsense path data to train CoSe-Co (§3) as no such dataset exists. We release the dataset and trained CoSe-Co model here.
3. Since CoSe-Co is based on a generative LM, it infers relevant and diverse paths which contain novel entities not present in the underlying KG (§4.2). Further, we show that it can be used for the task of KG completion performing better than previous method.
4. We augment generated commonsense knowledge in Multi-Choice QA (§4.3) and OpenCSR (§4.4) tasks leading to significant improvements (upto $\sim$3% and $\sim$7% respectively) over current SOTA methods. Further, using CoSe-Co paths results in better performance in low training data regimes indicating that they help in generalizing better.

2 Related Work

Commonsense Knowledge Graphs (KGs) are structured knowledge sources comprising of entity nodes in the form of symbolic natural language phrases connected through relations (Speer et al.,

2017; Sap et al., 2019a; Ilievski et al., 2021; Zhang et al., 2020; Xie & Pu, 2021). The knowledge in KGs is leveraged to provide additional context in NLP tasks (Bao et al., 2016; Sun et al., 2018; Lin et al., 2019) and perform explainable structured reasoning (Ren* et al., 2020; Ren & Leskovec, 2020). Additionally, a variety of Natural Language Inference (NLI) and generation tasks requiring commonsense reasoning have been proposed over the years (Zellers et al., 2018; Talmor et al., 2019; Sap et al., 2019b; Lin et al., 2020; 2021a;b). Pre-trained language models (PTLMs) (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2019) trained over large text corpus have been shown to posses textual knowledge (Jiang et al., 2020; Petroni et al., 2019; Roberts et al., 2020) and semantic understanding (Li et al., 2021). Consequently, they have been used for reasoning where they perform well to some extent (Bhagavatula et al., 2020; Huang et al., 2019). However, it remains unclear whether this performance can be genuinely attributed to reasoning capability or if it is due to unknown data correlation (Mitra et al., 2019; Niven & Kao, 2019; Kassner & Schütze, 2020; Zhou et al., 2020).

Due to this, various LM + KG systems have been explored (Feng et al., 2020; Wang et al., 2019; Lv et al., 2020) to combine broad textual coverage of LMs with KG's structured reasoning capability. Early works on KG guided QA retrieve sub-graph relevant to question entities but suffer noise due to irrelevant nodes (Bao et al., 2016; Sun et al., 2018). Hybrid graph network based methods generate missing edges in the retrieved sub-graph while filtering out irrelevant edges (Yan et al., 2020). Graph Neural Networks (GNNs) have been used to model embeddings of KG nodes (Wang et al., 2020a). More recently, Yasunaga et al. (2021) proposed an improved framework (QA-GNN) leveraging a static KG by unifying GNN based KG entity embeddings with LM based QA representations. Although, such frameworks extract relevant evidence from a KG, it undesirably restricts knowledge that can be garnered since knowledge source is static and might lack coverage due to sparsity (Bordes et al., 2013; Guu et al., 2015). Contrarily, we train a generative model on a given KG to enable it to dynamically generate relevant commonsense inferences making it more generalizable and scalable.

Bosselut et al. (2019) cast commonsense acquisition by LMs as KG completion. They propose COMET, a GPT (Radford et al., 2018a) based framework to generate tail entity given head and relation in a KG triple as input. Owing to training on symbolic KG nodes, using COMET in downstream tasks involving natural language text is not straightforward. Specifically, it requires extracting entities from text as a prerequisite (Becker et al., 2021). Further, training on single triples makes its application in tasks requiring multi-hop reasoning challenging due to large relation search space (Bosselut et al., 2021). To address this, Path Generator (PGQA) was proposed to generate commonsense paths between entities pair (Wang et al., 2020b). Designed for multi-choice QA, they extract question entities and generate paths between each question entity and answer choice pair. Even though generated paths are multi-hop, training on entities limits applying it directly on sentences due to train-inference input type mismatch. Further, being conditioned only on question-choice entity pairs, paths are generated ignoring overall question context. To mitigate these limitations, we design CoSe-Co as a generic framework to dynamically generate multi-hop commonsense inference given natural language sentence as input and show a comparison with PGQA. Separately, retrieval methods have been explored to search relevant sentences to generate text corresponding to concepts (Wang et al., 2021). Different from this task, we retrieve sentences relevant to paths in a KG to create paired sentence-path data which is used to train CoSe-Co.

3 PROPOSED CoSe-Co FRAMEWORK

Problem Setting Given a commonsense knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ is the set of entity nodes and $\mathcal{R}$ is the set of labeled directed relational edges between entities, we aim to model a **Co**mmon**Se**nse **Co**ntextualizer (**CoSe-Co**) which generates a set of commonsense inferences in the form of paths derived using $\mathcal{G}$, that are relevant to a natural language text given as input. It is desirable that such a generative commonsense knowledge model should be generic, task agnostic, and takes into account the overall context of language input while generating commonsense. Since most tasks comprise of text in the form of sentences, we model the input to CoSe-Co as a sentence. In order to train such a model, a dataset is required which comprises of mappings of the form $\{(s_1, p_1), (s_2, p_2), ..., (s_N, p_N)\}$, where s_j and p_j are relevant sentence-commonsense inference path pair. However, no existing dataset consists of such mappings. To bridge this gap, we first devise a methodology to create a dataset $\mathcal{D}$ comprising of sentences paired with relevant commonsense inference paths. Broadly, we first extract a large corpus $\mathcal{C}$ constituting sentences $\{s_1, s_2, ..., s_{|C|}\}$. Subsequently, we sample a set of paths $\mathcal{P} = \{p_1, p_2, ..., p_{|\mathcal{P}|}\}$ from $\mathcal{G}$ such that each $p \in \mathcal{P}$ is of the form $p = \{e_1, r_1, e_2, r_2, ..., e_{|p|+1}\}$, where $e_i \in \mathcal{E}$ and $r_i \in \mathcal{R}$. For each $p \in \mathcal{P}$, a set of

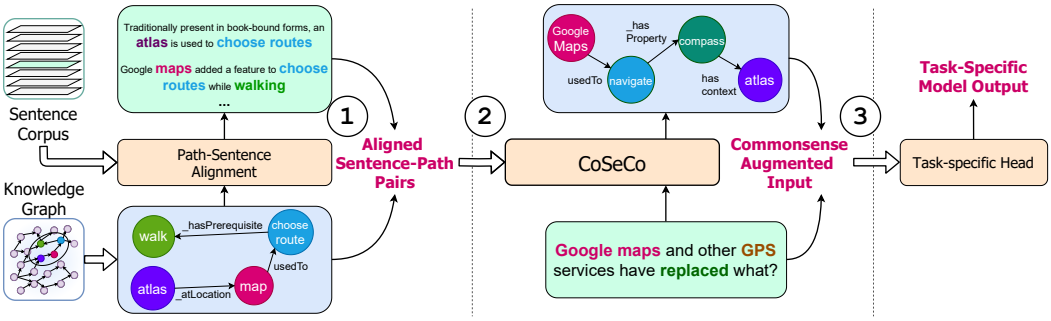


Figure 2: Our proposed approach consists of three main steps: (1) **Path to Sentence Alignment** to create the training data for CoSe-Co, (2) Training a **CommonSense Contextualizer** (CoSe-Co) to generate commonsense inferences relevant to a given natural language sentence, and (3) **Augmenting inferred knowledge** to the input as additional context in a downstream task.

contextually and semantically relevant sentences $S \subset \mathcal{C}$ is retrieved and mapped to p. We then train a generative LM based commonsense knowledge model using $\mathcal{D}$. During inference, given a sentence s', it generates commonsense paths of the form $p' = \{e'_1, r'_1, e'_2, r'_2, ..., e'_{|p'|+1}\}$ such that $e'_i \in \mathcal{E}'$ and $r'_i \in \mathcal{R}$. Here, $\mathcal{E}' = \mathcal{E} \cup \mathcal{E}_{novel}$ where $\mathcal{E}_{novel}$ are novel entities not present in $\mathcal{G}$. These include phrases present in an input sentence but not in $\mathcal{E}$ as well as entirely novel entities which the LM based knowledge model generates owing to a larger entity vocabulary than $\mathcal{G}$. The generated commonsense inference paths from CoSe-Co can then be used to augment context in downstream tasks. An overview of our framework is shown in Figure 2. We now explain each step in detail.

3.1 Sentence-Path Paired Dataset Creation

In order to train CoSe-Co, we create a novel dataset comprising of related sentence-commonsense path pairs. To obtain set $\mathcal{P}$, we perform random walk in $\mathcal{G}$ to extract multi-hop paths of the form $p = \{e_1, r_1, e_2, r_2, ..., e_{|p|+1}\}$, where the number of hops, denoted as path length $|p|$, is in range $[l_1, l_2]$. To avoid noisy paths which do not convey useful information, we employ relational heuristics in $\mathcal{P}$ (described in appendix F.1). Separately, the sentence corpus $\mathcal{C}$ is indexed using Apache Solr which is queried to retrieve sentences relevant to a path. We now explain this in detail.

Broadly, we map each path $p \in \mathcal{P}$ to a set of sentences $S \subset \mathcal{C}$ based on semantic similarity and overlap between entities in p and sentences. For this, consider a path $p = \{e_1, r_1, e_2, ..., e_{|p|+1}\}$. To ensure that retrieved sentences are similar to p, we devise two types of query templates - $Q1$ and $Q2$ which are used to create multiple queries per path while querying Solr. We design $Q1$ to capture relation information between entities in p in addition to entities themselves. Specifically, we extract non-contiguous entity-relation triples of the form $\{(e_i, r_i, e_{i+2})\}$ and $\{(e_i, r_{i+1}, e_{i+2})\}$. Here, we do not query entire path while retrieving sentences to ensure better coverage since it is unlikely that a sentence exists which contains all entities and relations present in a path. In $Q2$, we extract queries comprising of connected entities pairs $\{(e_i, e_{i+1})\}$. For each query q obtained from p according to $Q1$ and $Q2$, we query Solr and select sentences containing entities present in q. Subsequently, we rank retrieved sentences based on similarity between sentence embedding and embedded representation of the corresponding query q. The embeddings are obtained using SBERT (Reimers & Gurevych, 2019) since it is trained on siamese objective to learn semantically meaningful representations. Based on the ranking, we retain a maximum of top K' (= 10) sentences to ensure most semantically relevant sentences-path pairs are obtained and also to prevent CoSe-Co from getting biased towards generating particular paths. Figure 3 illustrates the entire sentence-path pairing process using an example from the dataset.

Using queries of type $Q1$ templates enables us to retrieve sentences that are relatively more semantically related to the overall path. For instance, consider a path 'violin *hasproperty* strings *_hasprerequisite* guitar *atlocation* concert'. Sentences retrieved using queries like {strings, *atlocation*, concert} (of the form (e_i, r_{i+1}, e_{i+2})) are more likely to be related to other entities in the path such as 'guitar'. Further, sentences that contain entities that are not directly connected in the corresponding path induce an inductive bias in CoSe-Co to generate paths that consist of intermediate entities which connect them. We perform ablations regarding the choice of query templates in §4.3.1.

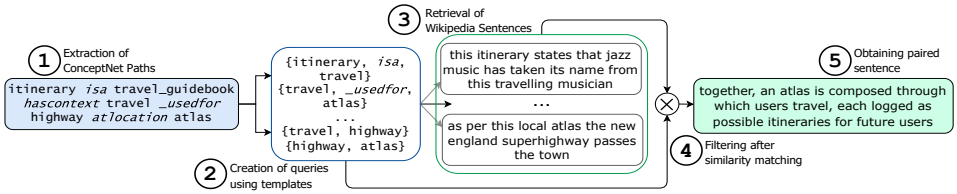


Figure 3: Obtaining the sentence-path paired dataset. We begin with paths from the knowledge graph and employ a two-step matching and filtering process to obtain relevant paired sentences from the given text corpora. Here we accompany each step with corresponding examples that we observed.

3.2 Sentence → Commonsense Path Generator

The sentence-commonsense path paired dataset $\mathcal{D}$ obtained in §3.1 is used to train a path generator model CoSe-Co$_\theta$ to generate commonsense inference path p relevant to the input sentence s. For this, we initialise the parameters θ of CoSe-Co with the weights of a generative pre-trained language model. Specifically, given a sentence $s = \{x^s_1, x^s_2, ..., x^s_{|s|}\}$ comprising of a sequence of tokens, we fine-tune the T5-base model (Raffel et al., 2019) to generate the corresponding path. Since T5 requires a prompt corresponding to the task to be performed, we prepend the string: '*convert sentence to path:*' to the tokens in s which is then processed by T5 encoder E_{θ_1} to give a sequence of outputs $O_E = \{o^E_1, o^E_2, ..., o^E_{|s|}\}$. T5 decoder D_{θ_2} is trained to sequentially generate the corresponding path tokens $p = \{x^p_1, x^p_2, ..., x^p_N\}$. During the decoding phase at time step t, D_{θ_2} is jointly conditioned on encoder outputs O_E and past tokens $x^p_{<t}$ in the path p while generating current path token x^p_t. E_{θ_1} and D_{θ_2}, where $\theta = \theta_1 \bigcup \theta_2$, are jointly optimized by minimizing loss $\mathcal{L}$:

$$\mathcal{L} = -\sum_{t=1}^{N} \log P(x^p_t | x^p_{<t}, O_E), \text{ where } P(x^p_t | x^p_{<t}, O_E) = \text{CoSe-Co}_\theta(s, x^p_{<t})$$

Separately, we design a variant where given a sentence-path pair, we randomly select an entity that co-occurs in the sentence as well as path and mask it in the sentence. The decision of whether a sentence will be masked during training is controlled by a probability p_{mask}. The model is then trained to generate path containing masked entity given the masked sentence as input. The intuition is to enforce CoSe-Co to capture context better through identifying masked entity while relating it with other entities during path generation. We discuss more and perform ablations to compare masked CoSe-Co with varying values of p_{mask} and unmasked variant in §4.3.1.

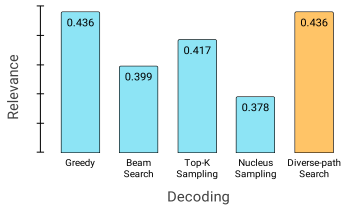
3.3 Path Decoding During Inference

As in most sequence generation tasks, teacher forcing is used to train the model, while a decoding strategy is used to generate diverse outputs during inference (Vaswani et al., 2017). To maximise contextual knowledge obtained from paths for each sentence in a downstream task, we generate multiple paths. In order to improve diversity between paths while not losing out relevance, we implement a path-specific variant of beam search, *diverse-path search*. Diversity is ensured in diverse-path search by sampling top-k most probable tokens at the first generation step followed by decoding the most probable sequence for each of them, thus returning k paths. This approach is motivated by the observation that when generating a path, the initial entity guide the overall decoding of the path.

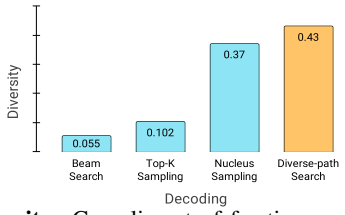
4 Experiments and Evaluation

4.1 Implementation Details

We choose Wikipedia as the sentence corpus $\mathcal{C}$, and ConceptNet (Speer et al., 2017) as the knowledge graph $\mathcal{G}$. The subset of Wikipedia that we use comprises of ∼5M articles, from which we extract ∼92.6M sentences. ConceptNet comprises of ∼8 million nodes as concepts linked through 34 unique commonsense relations with ∼21 million links in total. We sample ∼28M paths that have a length $|p|$ in the range $l_1 = 2$ and $l_2 = 5$. We obtain a total of ∼290K sentence-path pairs. CoSe-Co is trained until validation loss across an epoch does not increase, with maximum number of epochs $= 5$. p_{mask} is set to 0.33 based on tuning on CSQA dev set and number of paths per sentence $k = 5$ during inference. AdamW optimizer (Loshchilov & Hutter, 2017) is used to train parameters with a learning rate of $5e-4$, weight decay of 0.01 and epsilon of $1e-8$ using a single A-100 GPU with batch-size 8 and 4 gradient accumulation steps.



(a) **Relevance**: BLEU score of generated paths computed using ground truth paths



(b) **Diversity**: Compliment of fraction overlap between top-5 sampled paths.

Figure 4: Analysis and comparison of generated paths across different decoding strategies

Input	**CoSe-Co Outputs**
What do people typically do while playing guitar?	(playing_guitar *causes* singing *usedfor* people *capableof* feeling_sad) (playing_guitar *hassubevent* sing *_causesdesire* singing *_occupation* musician *genre* **folk_rock**) (play_guitar *_usedfor* guitar *atlocation* **symphony_halls**_or_musical_instruments_or_bands *_atlocation* people)
Where are you likely to find a hamburger?	(burger *_isa* hamburger *atlocation* fast_food_restaurant *usedfor* eating_food) (burger_king *_usedfor* hamburger *atlocation* fast_food_restaurant *isa* place *capableof* **take_car_for_drive**) (fast_food_restaurant *_isa* taco_bell *product* hamburger *madeof* **wheat_flour_and_salt**)
In what Spanish speaking North American country can you get a great cup of coffee?	(bretagne *partof* north_america *_atlocation* cup_of_coffee *hascontext* usa *isa* country) (**hot_beverage** *_isa* coffee *atlocation* cup_of_coffee *hascontext* north_america *_partof* grenada) (good_coffee *hasa* **caffiene_in_milk_and_sugar** *atlocation* in_ canada)

Table 1: Commonsense paths generated by CoSe-Co for questions in *CommonsenseQA* data. Potential answers observed in path itself are highlighted, context-enriching concepts are **coloured**.

4.2 Analysing Generated Paths

We analyse quality of generated paths on three aspects - *Relevance*, *Diversity* and *Novelty*, evaluated on test split of our sentence-path dataset. We estimate **Relevance** by treating each triple in generated and ground truth paths (for a given test sample) as one uni-gram followed by determining BLEU score (Papineni et al., 2002) between them. To estimate **Diversity**, we extract top-$k = 5$ paths for each sentence, consider each pair combination amongst them and calculate amount of fractional overlap (intersection over union of set of path entities) between them. Compliment of overlap $(1 - overlap)$ followed by mean over entire test split conveys how diverse paths are. Figure 4 shows corresponding results. It is observed that paths generated using nucleus sampling are diverse but lack relevance, while an opposite trend is observed for top-k sampling. *Diverse-path search* provides best balance between relevance (0.436) and diversity (0.43). We estimate **Novelty** by determining fraction of total entities in a generated path (for a test sentence) that are not present in any of the training paths followed by averaging over the entire test split. CoSe-Co attains a novelty of 23.28% which shows that good fraction of entities in generated path are novel. Table 1 shows qualitative examples of paths generated for few CSQA questions. CoSe-Co generates paths contextually relevant to question in addition to inferring novel entities.

Further, we perform KG completion (predicting tail entity given head entity and relation of a KG triple) using CoSe-Co since it generates paths which essentially comprise of triples. We compare the performance with COMET (Bosselut et al., 2019). We consider test split of sentence-path dataset comprising of $11,264$ paths and extract triples. We filter out triples appearing in training paths of CoSe-Co and train set triples of COMET yielding 717 test triples in total. CoSe-Co achieves an accuracy of 24.12% which is significantly better than COMET which provides accuracy of 9.76%.

4.3 Multi-Choice Question Answering

We perform multiple choice question answering on CSQA dataset (Talmor et al., 2019) which is the most commonly used benchmark for this task. Here, a question is given with 5 answer choices and the model has to predict the correct one. As an example, consider the question - 'Where could you see an advertisement while reading news?' with answer choices as - 'television, bus, email, web page, and la villa'. Here the correct answer is 'web page'. Following Wang et al. (2020b), we

Methods	20% Train IHtest (%)	60% Train IHtest (%)	100% Train IHdev (%)	100% Train IHtest (%)
T5-base (w/o KG) (Raffel et al., 2019)	–	–	61.88 (±0.08)	57.34 (±0.21)
T5-large (w/o KG) (Raffel et al., 2019)	–	–	69.81 (±1.02)	67.80 (±0.83)
RoBERTa-large (w/o KG)	46.25 (±0.63)	52.30 (±0.16)	73.07 (±0.45)	68.69 (±0.56)
+ RGCN (Schlichtkrull et al., 2018)	45.12 (±0.69)	54.71 (±0.37)	72.69 (±0.19)	68.41 (±0.66)
+ GconAttn (Wang et al., 2019)	47.95 (±0.11)	54.96 (±0.69)	72.61(±0.39)	68.59 (±0.96)
+ KagNet (Lin et al., 2019)	–	–	73.47 (±0.22)	69.01 (±0.76)
+ RN (Santoro et al., 2017)	45.12 (±0.69)	54.23 (±0.28)	74.57 (±0.91)	69.08 (±0.21)
+ MHGRN (Feng et al., 2020)	–	–	74.45 (±0.10)	71.11 (±0.81)
+ PGQA (Wang et al., 2020b)	58.25 (±0.43)	<u>69.66</u> (±0.97)	<u>77.53</u> (±0.47)^{q}	71.19 (±0.49)
+ QA-GNN (Yasunaga et al., 2021)	<u>59.08</u> (±1.25)	68.70 (±0.62)	75.54 (±0.42)	<u>72.29</u> (±0.43)^{p}
+ CoSe-Co (**Ours**)	**61.20** (±0.19)^{p,q}	**70.23** (±0.40)^{q}	**78.15** (±0.23)^{p,q}	**72.87** (±0.31)^{p,q}

Table 2: Performance comparison on in-house dev (**IHdev**) and test (**IHtest**) split of ***CommonsenseQA*** dataset (Talmor et al., 2019). All scores are averaged across 5 runs. First row depicts amount of training data used. The second-best number for each column is underlined while best is in bold. Superscripts 'p' and 'q' denote statistically significant differences (***p-value* of 0.05**) in comparison to two of our major baselines - PGQA and QA-GNN respectively.

use their approach for this task which embeds question + choices context using **RoBERTa** and use CLS token output (Liu et al., 2019) to perform attention over paths embeddings generated using their commonsense module. The output of attention module together with embedding of question and answer choices is used to predict the correct answer. We just **replace** their paths with the ones generated by CoSe-Co. We discuss the elaborated details of PGQA framework in the appendix D.

Table 2 shows results on CSQA which are usually averaged over 5 runs on this benchmark. We compare against several baselines broadly classified into ones using static KG such as MHGRN (Feng et al., 2020), QA-GNN (Yasunaga et al., 2021) etc. and others which train a dynamic path generator (PGQA) (Wang et al., 2020b) as commonsense module. When using entire training data, we observe that CoSe-Co performs better than all baselines[2] on test set. PGQA (Wang et al., 2020b) is most similar to our approach since we adapt their framework and just differ in how commonsense paths are generated. We outperform PGQA with a gain of 1.68% in accuracy on test split signifying the relevance and applicability of inferences generated by CoSe-Co. CoSe-Co performs better than QA-GNN (Yasunaga et al., 2021) also particularly in low training data regimes with performance gains of $\sim 2\%$ (and $\sim 3\%$ over PGQA) showing that while QA-GNN is more sensitive towards amount of training data used, CoSe-Co is more robust and helps in generalizing better. Qualitatively, consider the question - 'Where could you see an advertisement while reading news?' PGQA generates the path - 'read_news *hassubevent* read *relatedto* news *atlocation* television' ignoring the context that advertisement is being seen along with reading news and ends up predicting television as answer which is wrong. While CoSe-Co generates - 'spread_information *_capableof* advertisement *atlocation* web_page *usedfor* reading_news'. Here it can be seen that CoSe-Co identifies that seeing the advertisement and reading news is happening together and generates path accordingly to relate them with 'web page' which is the correct answer. Please refer to appendix A for human study and more qualitative comparison with baselines on CSQA dataset. We study effect of using a different generative LM (GPT-2 as used by PGQA) as backbone for CoSe-Co in appendix C and empirically establish that performance gains over PGQA are independent of which LM is used.

4.3.1 Ablation Study

Entity masking during training: As described in §3.2, probability p_{mask} is used to decide whether an input sentence will be masked to enable CoSe-Co to capture context better through identifying masked entity. We tune p_{mask} over CSQA dev set and determine 0.33 as optimal value. Table 3 shows comparison where masking during training works better than not masking ($p_{mask} = 0$). We show qualitative analysis of paths for different p_{mask} in appendix B. Thus, we perform further experiments keeping $p_{mask} = 0.33$ during training. Further, $0 < p_{mask} < 1$ ensures trained CoSe-Co can be used to generate paths for both masked and unmasked inputs during inference.

[2]Results for PGQA (Wang et al., 2020b) and QA-GNN (Yasunaga et al., 2021) are reproduced using their official implementations while numbers for other baselines have been taken from these two works.

Entity masking for training		Query template for path-sentence		Entity masking for inference	
p_{mask}	IHdev (%)	Query	IHdev (%)	Type	IHdev (%)
0.0	77.52 (±0.44)	$Q1$	77.69 (±0.43)	*Interrogative*	78.07 (±0.56)
0.50	77.38 (±0.40)				
0.67	77.61 (±0.79)	$Q2$	77.25 (±0.64)	*Random*	77.90 (±0.84)
1.0	77.71 (±1.17)				
0.33	**78.15** (±0.23)	**Q1 + Q2**	**78.15** (±0.23)	**No Masking**	**78.15** (±0.23)

Table 3: Studying the effect of ablation variants through comparison on *CommonsenseQA* dev set.

	ARC			**QASC**			**OBQA**		
Hits@K	H@10	H@30	H@50	H@10	H@30	H@50	H@10	H@30	H@50
DrFact (Lin et al., 2021a)	36.09	53.25	64.50	21.78	37.62	51.49	12.08	23.77	35.13
T5-base (Raffel et al., 2019)	49.70	**67.46**	71.01	33.66	47.52	53.47	17.42	29.55	37.88
+ CoSe-Co Paths	**50.89**	63.91	69.23	30.69	47.52	56.44	20.45	34.09	**45.45**
+ CoSe-Co Concepts	44.97	66.86	**73.37**	**35.64**	**47.52**	**57.43**	**21.21**	**35.61**	42.42
Recall@K	R@10	R@30	R@50	R@10	R@30	R@50	R@10	R@30	R@50
DrFact (Lin et al., 2021a)	12.60	21.05	27.27	12.38	22.28	29.70	6.12	11.85	16.51
T5-base (Raffel et al., 2019)	15.98	28.30	33.93	18.98	26.40	30.53	8.52	14.61	18.71
+ CoSe-Co Paths	**16.87**	27.45	33.73	17.49	28.05	**33.33**	9.90	16.53	**22.42**
+ CoSe-Co Concepts	15.12	**28.99**	**35.21**	**19.64**	**28.05**	33.00	**9.96**	**17.35**	21.10

Table 4: Performance comparison on Hits@K and Recall@K metrics for OpenCSR (Lin et al., 2021a) on ARC, QASC and OBQA datasets. DrFact is a BERT-based current state of the art method.

Path-sentence query templates: We obtain path to sentence mapping while creating training dataset by devising two query templates $Q1$ (includes relation information) and $Q2$ (does not capture relations) as described in §3.1. Table 3 shows a performance comparison of using different query templates on CSQA dev set. We observe that training CoSe-Co on dataset created using $Q1$ alone outperforms the one trained only on $Q2$, indicating the improvement due to relation information. It is to be noted that both these variants of the dataset are of the same size. We also observe that combining these two variants into a single dataset, $Q1 + Q2$, results in further improvement.

Entity masking during inference: Since CoSe-Co is given a masked sentence as input during training ($p_{mask} = 0.33$), we explore the effect of similar type of masking during inference. Specifically, certain parts of input sentence can be replaced with masked token to enable CoSe-Co to generate paths that lead towards filling the mask. As reported in Table 3, the variant where no masking is done performs marginally better than when *Interrogative* or *Random* tokens in sentence are masked. Thus, by default we do not perform masking during inference unless otherwise stated.

4.4 OpenCSR: Open-Ended CommonSense Reasoning

In CSQA, often multiple choices are appropriate and model gets penalised unfairly if it predicts suitable answer which does not match with single ground truth. To mitigate this, Lin et al. (2019) re-configured three multi-choice QA datasets for OpenCSR as a generative task where interrogative tokens are replaced with blanks ("_ _") and a set of singleton tokens is labeled as ground truth. To generate a set of paths P, we use inference masking variant of CoSe-Co since question contains a blank. Given a question q, blank ("_ _") is replaced with mask token. To inject our paths, we devise a supervised method where we adapt a separate T5-base model for OpenCSR such that concatenation of q and paths is given as input to T5 along with the prefix *'fill mask to answer question: '*. T5 is trained to generate one of the answers in ground truth set. During inference, top-K answers, determined on basis of generation likelihood from T5 decoder, are taken as answer candidates.

Table 4 shows comparison between DrFact[3] (Lin et al., 2021a) (current state-of-the-art based on BERT-base) and our supervised method which uses CoSe-Co's paths. Specifically, we evaluate - 1) 'Paths from CoSe-Co' where generated paths are concatenated; and 2) 'Concepts from CoSe-Co' where only entities in generated paths are appended. Since our supervised method is based on pre-trained T5, for fair comparison and to probe if performance changes are due to T5, we compare

[3]The authors communicated that the test set and leader board has not been released yet. Hence, we report results using the author provided code and validation set. Also, they run their models on single seed.

MRPC Paraphrase Generation					
	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
T5-base	43.10	36.10	61.80	36.33	47.10
+ CoSe-Co Paths	**44.50**	**36.70**	**62.50**	**37.34**	**48.50**
CommonGen					
T5-base	9.90	21.10	**36.70**	14.54	**44.70**
+ PGQA Paths	9.80	21.0	36.60	14.64	**44.70**
+ CoSe-Co Paths	**10.10**	**21.20**	**36.70**	**14.78**	**44.70**

Table 5: Using CoSe-Co Paths leads to improvements in MRPC paraphrase generation and no change in CommonGen task. Generative commonsense methods like PGQA which rely on answer choices cannot be applied in tasks like paraphrase generation where entities are not available.

against another baseline: T5-base fine tuned for OpenCSR without paths. We evaluate two metrics as used in Lin et al. (2021a): 1) **Hits@K**: Determined on basis of whether generated and ground truth answer sets have non-empty intersection; 2) **Recall@K**: Estimates how many predicted answers match at least one ground truth answer. We vary value of K to be {10, 30, 50}. We evaluate on three datasets - ARC (Clark et al., 2018), QASC (Khot et al., 2020), and OBQA (Mihaylov et al., 2018).

CoSe-Co performs significantly better than both baselines on all datasets uniformly. Specifically, 'Concepts from CoSe-Co' usually performs better which shows entities in paths generated by CoSe-Co are useful. Our approach provides performance gains of upto 8%, 6%, 10% in Hits@50 and 8%, 3%, 6% in Recall@50 over DrFact on ARC, QASC and OBQA respectively. Even though T5-base baseline performs better than DrFact, commonsense from CoSe-Co augmented with T5 achieves new state of the art on this task with performance gains upto 2.3%, 3.9%, 7.5% in Hits@50 and 1.2%, 2.5%, 3.7% in Recall@50 over T5-base on ARC, QASC and OBQA respectively.

4.5 Effect of Concatenating CoSe-Co paths in Generation Tasks

We explore augmenting CoSe-Co paths for text generation tasks where the aim is not to obtain SOTA results but to analyse if it improves performance of a base model. Specifically we study - 1) Paraphrase Generation: given a sentence, generate another sentence expressing same meaning using different words where commonsense is usually helpful in rephrasing, and 2) CommonGen: generate a sentence describing a scene using a concept-set requiring commonsense to compose concepts. Since T5 (Raffel et al., 2019) is designed for generation tasks, we fine-tune T5-base to generate annotated paraphrase given a sentence as input on MRPC dataset (Dolan & Brockett, 2005). For commongen (Lin et al., 2020), given concepts 'bird hold cup eat food', T5 is trained to generate sentence like 'The bird eats food from cup that its owner is holding'. Generated paths are appended as string to input. Please refer to appendix E for elaborated implementation details and discussion.

Table 5 summarises results evaluated through commonly used generation metrics - BLEU (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). Amongst these, SPICE is considered to correlate most with human judgement. Using CoSe-Co paths results in better paraphrase generation as indicated by $\sim$1-1.5% improvement in most generation metrics. For CommonGen, no significant improvements are observed and performance remains similar without any degradation.

5 Conclusion

We presented CoSe-Co, a task agnostic framework to generate contextually relevant commonsense inferences given natural language text as input. Unlike previous generative approaches, which ignore important contextual clues in the input text and operate on entities, CoSe-Co works directly on input sentences making it suitable for a wide variety of NLP tasks. We created a novel dataset of $<$ sentence, commonsense paths $>$ pairs for training CoSe-Co and make it available to the community. Empirical evaluation shows that commonsense inferences generated by CoSe-Co are relevant, diverse and also contain novel entities not present in the KG used to train CoSe-Co. Additionally, we show that CoSe-Co can be used for KG completion. We augment knowledge generated by CoSe-Co in commonsense tasks such as Multi-Choice QA, and Open-ended CommonSense Reasoning, achieving SoTA results for these tasks. Further, we also used CoSe-Co for NLP tasks such as paraphrase generation achieving improved performance. As future work, CoSe-Co can be enhanced by utilizing other commonsense KGs. Also, new ways to utilize the generated knowledge and augment commonsense paths in downstream tasks can be explored.

REPRODUCIBILITY STATEMENT

We devise a methodology to create a novel sentence-commonsense paths data to train CoSe-Co. We release the dataset and trained CoSe-Co model here. We provide detailed steps of our algorithm to create the dataset and training CoSe-Co model (§3) accompanied with the detailed implementation details (§4.1). For each downstream task where CoSe-Co is used to augment knowledge, we provide implementation details and datasets used in the main paper (§4.2, §4.3, §4.4 and §4.5) as well as further in-depth details in appendix F. Links to sources of baseline codes and different datasets used for each task are also provided in these sections.

REFERENCES

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pp. 382–398. Springer, 2016.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2503–2514, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1236.

Maria Becker, Katharina Korfhage, Debjit Paul, and Anette Frank. Co-nnect: A framework for revealing commonsense knowledge paths as explicitations of implicit knowledge in texts. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pp. 21–32, Groningen, The Netherlands (online), June 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.iwcs-1.3.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. Abductive commonsense reasoning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Byg1v1HKDB.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4762–4779, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1470. URL https://aclanthology.org/P19-1470.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 4923–4931. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16625.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1295–1309, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.99. URL https://aclanthology.org/2020.emnlp-main.99.

Kelvin Guu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 318–327, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1038. URL https://aclanthology.org/D15-1038.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. 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)*, pp. 2391–2401, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1243. URL https://aclanthology.org/D19-1243.

Filip Ilievski, Pedro Szekely, and Bin Zhang. Cskg: The commonsense knowledge graph. In *European Semantic Web Conference*, pp. 680–696. Springer, 2021.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020. URL https://www.aclweb.org/anthology/2020.tacl-1.28.pdf.

Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7811–7818, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.698. URL https://www.aclweb.org/anthology/2020.acl-main.698.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090, Apr. 2020. doi: 10.1609/aaai.v34i05.6319. URL https://ojs.aaai.org/index.php/AAAI/article/view/6319.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1813–1827, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL https://aclanthology.org/2021.acl-long.143.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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)*, pp. 2829–2839, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1282. URL https://aclanthology.org/D19-1282.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1823–1840, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.165. URL https://aclanthology.org/2020.findings-emnlp.165.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William Cohen. Differentiable open-ended commonsense reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4611–4625, Online, June 2021a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.naacl-main.366.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1504–1515, 2021b.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. URL http://arxiv.org/abs/1711.05101.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8449–8456, Apr. 2020. doi: 10.1609/aaai.v34i05.6364. URL https://ojs.aaai.org/index.php/AAAI/article/view/6364.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *arXiv preprint arXiv:1909.08855*, 2019.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4658–4664, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1459. URL https://aclanthology.org/P19-1459.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 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)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://www.aclweb.org/anthology/D19-1250.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference*

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128.

Adam Poliak. A survey on recognizing textual entailment as an nlp evaluation. *arXiv preprint arXiv:2010.03061*, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018a.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018b.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL http://arxiv.org/abs/1910.10683.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

Hongyu Ren and Jure Leskovec. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19716–19726. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e43739bba7cdb577e9e3e4e42447f5a5-Paper.pdf.

Hongyu Ren*, Weihua Hu*, and Jure Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgr4kSFDS.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://www.aclweb.org/anthology/2020.emnlp-main.437.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/e6acf4b0f69f6f6e60e9a815938aa1ff-Paper.pdf.

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

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. 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)*, pp. 4463–4473, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4231–4242, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1455. URL https://aclanthology.org/D18-1455.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

Han Wang, Yang Liu, Chenguang Zhu, Linjun Shou, Ming Gong, Yichong Xu, and Michael Zeng. Retrieval enhanced model for commonsense generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3056–3062, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.269. URL https://aclanthology.org/2021.findings-acl.269.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. Entity context and relational paths for knowledge graph completion. *arXiv preprint arXiv:2002.06757*, 2020a.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro A. Szekely, and Xiang Ren. Connecting the dots: A knowledgeable path generator for commonsense question answering. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 4129–4140. Association for Computational Linguistics, 2020b. doi: 10.18653/v1/2020.findings-emnlp.369. URL https://doi.org/10.18653/v1/2020.findings-emnlp.369.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. Improving natural language inference using external knowledge in the science questions domain. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7208–7215, Jul. 2019. doi: 10.1609/aaai.v33i01.33017208. URL https://ojs.aaai.org/index.php/AAAI/article/view/4705.

Yubo Xie and Pearl Pu. How commonsense knowledge helps with natural language tasks: A survey of recent resources and methodologies. *arXiv preprint arXiv:2108.04674*, 2021.

Jun Yan, Mrigank Raman, Aaron Chan, Tianyu Zhang, Ryan Rossi, Handong Zhao, Sungchul Kim, Nedim Lipka, and Xiang Ren. Learning contextualized knowledge structures for commonsense reasoning. *arXiv preprint arXiv:2010.12873*, 2020.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 535–546, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.45. URL https://aclanthology.org/2021.naacl-main.45.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 93–104, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1009. URL https://aclanthology.org/D18-1009.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. Aser: A large-scale eventuality knowledge graph. In *Proceedings of The Web Conference 2020*, pp. 201–211, 2020.

Pei Zhou, Rahul Khanna, Seyeon Lee, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. Rica: Evaluating robust inference capabilities based on commonsense axioms. *arXiv preprint arXiv:2005.00782*, 2020.

A Qualitative Comparison of CoSe-Co with Baselines on CSQA

Table 6 shows qualitative comparison between CoSe-Co and baselines on the CSQA dataset.

Question	Predictions			Generated Paths	
	PGQA	QA-GNN	Ours	PGQA	CoSe-Co
Where could you see an advert-isement while reading news?	television	web page	web page	(read_news *hassubevent* read *relatedto* news *atlocation* television) (read_news *hassubevent* read *relatedto* page)	(spread_information *_capableof* advertisement *atlocation* web_page *usedfor* reading_news) (news_article *isa* article *atlocation* web_page *_receivesaction* advertisement)
What can years of playing tennis lead to?	becoming tired	becoming tired	tennis elbow	(playing_tennis *causes* becoming_tired) (play *antonym* fun *usedfor* playing_tennis *causes* tennis_elbow)	(injury *_hassubevent* playing_tennis *hasprerequisite* practice_taking_care_of_sports_equipment) (playing_tennis *hassubevent* injury *hasprerequisite* practice *_hasfirstsubevent* be_better_at_new_things)
A person writes a check to a clerk, where does the clerk put them?	desk drawer	cash register	cash register	(put *relatedto* desk *partof* drawer) (check *relatedto* cash *relatedto* register) (write *relatedto* desk *partof* drawer)	(make_payments *_capableof* clerk desires check *_atlocation* cash_registers *_usedfor* to_pay_for_goods) (cash_registers *_usedfor* clerk *isa* person *desires* clean_house *hasprerequisite* put_things_into_places)
Where could you find some large pieces of paper that are not for sale?	office supply store	cabinet	artist's studio	(large *relatedto* note *relatedto* paper *relatedto* office_supply) (pieces *relatedto* part *relatedto* paper *relatedto* office_supply)	(shredded_paper *usedfor* sale *_hassubevent* buying_products *_nothasproperty* artist_studio) (write_letters *_usedfor* paper *receivesaction* sell_for_money *atlocation* store)
What do humans take in while breathing?	air	oxygen	oxygen	(humans *relatedto* air) (breathing *hassubevent* air) (human *relatedto* breathing *hassubevent* oxygen)	(breathing *hassubevent* inhale *motivatedbygoal* fresh_air *_atlocation* oxygen) (inhaling *_hassubevent* breathing *causes* life *_usedfor* living_life *hasprerequisite* good_health)

Table 6: Comparison between predictions made by PGQA (Wang et al., 2020b), QA-GNN (Yasunaga et al., 2021), and CoSe-Co on a subset of CSQA's in-house test set (Talmor et al., 2019). Commonsense paths that are responsible for the corresponding predictions are also given for both the path-based models. Underlined portions represent the meaningful path sub-structures which direct the overlying model towards the correct answer.

We conduct a **human study** wherein we presented evaluators with questions from CSQA dataset with corresponding commonsense paths generated by CoSe-Co and PGQA in an anonymized manner. We asked them to compare the paths based on their contextual relevance with the complete sentence and classify them into one of three categories - 1) 'CoSe-Co is better than PGQA', 2) 'PGQA is better than CoSe-Co', 3) 'Both are of the similar quality'. A total of 150 questions samples were randomly sampled from the test set and presented to 6 evaluators (25 samples each). Following are our observations:

Number of samples where CoSe-Co is better: 62 (41.33% of 150 samples)
Number of samples where PGQA is better: 38 (25.33% of 150 samples)
Number of samples where both are of similar quality: 50 (33.33% of 150 samples)

This shows that commonsense generated by CoSe-Co is found to be more relevant in human evaluation. Also, if we exclude neutral samples and consider the 100 samples where the path generated by one of either approach is found to be better, CoSe-Co's paths are found to be more relevant in 62 samples (62% of 100 samples) while PGQA's paths are more relevant in 38 samples (38% of 100 samples).

B ENTITY MASKING WHILE TRAINING CoSe-Co

Table 7 shows the various kinds of paths obtained from CoSe-Co when trained with different values of p_{mask}, across the same original question. A number of observations can be made. First, the paths obtained from the variant which is trained without any masking ($p_{mask} = 0.0$) produces inferences that enrich the overall context of certain entities in question but do not necessarily capture the interrelation between them and thus the overall intention of the question. With the configurations that are trained with $p_{mask} \neq 0$, the various paths capture the overall context in an answer-oriented manner. These configurations also allow us to mask concepts in the original question such that CoSe-Co can exploit the unmasked entities to direct its generated paths in a manner that best suit the blank. This is evident from the second half of Table 7. When the interrogative element is masked in the first example, the paths are directed towards actually finding the best answer, while when 'Google maps' is replaced in the third example, the paths are clearly focused on predicting concepts related to GPS systems.

Input	**CoSe-Co Variant**	**CoSe-Co Outputs**
Google maps and other GPS services have replaced what?	$p_{mask} = 0.0$	(orienteering *_usedfor* map *_isa* **navigation** *_hascontext* GPS *_product* gsrc) (satellite_gear *hascontext* maps *capableof* **help_find_places** *_usedfor* map_store_in_city) (similar_to_gps *_hasproperty* google *hascontext* maps *capableof* **show_live_places**) (geospatial *_isa* navigation *_usedfor* google *hascontext* maps *capableof* show_people_places)
Google maps and other GPS services have replaced what?	$p_{mask} = 1.0$	(gps *hascontext* **navigation** *_usedfor* satellite *atlocation* orbital_stations_in_space) (satellite_navigation *_usedfor* compass *capableof* **point_north** *_hassubevent* **driving_car**) (navigation *_usedfor* satellite *_isa* gps *hascontext* navigation *_entails* competition) (**location_where_go**_gallery *_definedas* navigation *_usedfor* satellite *_hascontext* gps)
Google maps and other GPS services have replaced what?	$p_{mask} = 0.5$	(compass *_isa* google_maps *usedfor* **navigation** *_hascontext* gps *isa* navigating_map) (**location_where_go_camping** *_isa* location *_usedfor* map _product google_maps) (satellite_gear *_isa* GPS *usedfor* navigation *_hascontext* gps *isa* navigating_map) (navigation_maps *_usedfor* map *_isa* navigation_map *hascontext* navigation)
Google maps and other GPS services have replaced what?	$p_{mask} = 0.33$	(orienteering *_usedfor* maps *_isa* google_maps *_hasprerequisite* looking_through_telescope) (**location_where_go_shopping** *_definedas* where_go_shopping *_usedfor* map) (navigation_maps *_isa* maps *_usedfor* satellite *locatednear* planet) (satellite_navigation *_usedfor* maps *_hascontext* google_maps *capableof* show_locations)
Google maps and other GPS services have replaced [MASK].	$p_{mask} = 0.33$	(gps *hascontext* maps *_usedfor* satellite *locatednear* planet) (navigation_maps *isa* navigation *_usedfor* compass *capableof* **point_north_handle**) (satellite_navigation *_usedfor* compass *capableof* point_north_or_south_hemispheres) (location_where_go_if_near_beach *_definedas* map *usedfor* navigation *_mannerof* sport)
Google maps and other GPS services have [MASK] what?	$p_{mask} = 0.33$	(orienteering *_usedfor* map *_isa* google_maps *_hascontext* gps) (**location_where_go_if_need_to** *_definedas* location *_isa* map *usedfor* **information**) (located_in_latin_america *_receivesaction* israel *_language* latin_america) (navigation_maps *usedfor* find_place *_hasprerequisite* go_to_market) (satellite_navigation *_usedfor* maps *capableof* **show_locations_and_routes**)
[MASK] and other GPS services have replaced what?	$p_{mask} = 0.33$	(navigation_system *_isa* GPS *hascontext* astronomy *_field* edmond_halley) (location_where_go_if_in_accident *_usedfor* map *_atlocation* GPS_systems) (**radio_frequency_messaging** *_isa* GPS *hasproperty* useful) (receiver *partof* radio *_isa* gps *hascontext* navigation *_usedfor* compass)

Table 7: Examples of commonsense inferences obtained for different input forms of the same question from CoSe-Co when trained with different values of p_{mask}. Potential answers which are observed in a path are highlighted, while context-enriching concepts are **coloured**.

Methods	100% Train IHdev (%)	IHtest (%)
RoBERTa-large (w/o KG)	73.07 (±0.45)	68.69 (±0.56)
+ PGQA w/ GPT-2	77.53 (±0.47)	71.19 (±0.49)
+ CoSe-Co w/ GPT-2	77.9 (±0.37)	72.67 (±0.18)
+ PGQA w/ T5-base	77.56 (±0.32)	71.31 (±0.44)
+ CoSe-Co w/ T5-base	78.15 (±0.23)	72.87 (±0.31)

Table 8: Performance comparison between using T5-base and GPT-2 as backbone language model for PGQA and CoSe-Co for multi-choice QA task on CSQA dataset.

C Comparison with GPT-2 as backbone language model

We decided to use T5-base as a design choice as we were required to train a text-to-text model where given a sentence as input, the model has to generate the relevant path as output. Since T5-base is a text-to-text generation language model, we felt that it is a suitable choice.

To empirically establish that improvements over PGQA are not due to using T5-base instead of GPT-2, we performed an experiment to replace T5-base with GPT-2 as the backbone language model of CoSe-Co. We train GPT-2 using the same sentence-path dataset as we used for T5-base by providing it as input the sentence followed by a [SEP] token and adapting GPT-2 to generate the corresponding path. Additionally, we also experiment with replacing the language model in PGQA from GPT-2 to T5-base. Table 8 summarises the results obtained for multi-choice QA on CSQA where it can be seen that using GPT-2 vs T5 does not lead to noticeable changes in the performance. The test accuracy attained by CoSe-Co with T5-base is 72.87% which is almost the same as for CoSe-Co with GPT-2: 72.67%. A similar observation is seen for PGQA where using T5-base backbone gives 71.31% and using GPT-2 gives 71.19%. Further, we would like to highlight that CoSe-Co with GPT-2 backbone attains 72.67% accuracy and performs better than PGQA with GPT-2 (71.19%).

Based on these observations, we can conclude that performance gains of CoSe-Co over PGQA are not due to using different backbone but because CoSe-Co is trained over semantically related sentence-commonsense pairs that enables it to generate contextually more relevant commonsense.

D Details of PGQA Baseline

PGQA (Wang et al., 2020b) leverages the commonsense paths generated by their path generator module along with the question and candidate answer choices to perform multi-choice QA on CSQA dataset (Talmor et al., 2019). Specifically, given a question q with corresponding candidate answer choices set $C = \{c_1, \ldots, c_n\}$, the PGQA framework generates commonsense inferences for each pair of answer choice c_i and entities extracted from q. A total of k paths corresponding to each answer choice c_i are obtained to get a resultant set of paths - P_{q-c_i}. Further, an average over the hidden representations corresponding to sequence of decoded tokens from the final layer of their path generator decoder are used as path embedding and combined as - $H_S \in R^{k \times h_D}$ to represent the paths in P_{q-c_i}. Following this, they augment the choice into q by replacing the interrogative phrase in q with c_i to obtain q'. For instance, given the question 'Google maps and other GPS services have replaced what?', the answer choice 'atlas' is augmented into the question as: 'Google maps and other GPS services have replaced *atlas*.'

To embed the augmented question and corresponding answer choice, they use a pre-trained LM encoder E (such as RoBERTa (Liu et al., 2019)) to embed the query - '[CLS] q' [SEP] c_i' corresponding to c_i. The representation corresponding to [CLS] token is extracted from the final hidden layer as $h_{US} \in R^{h_E}$. In order to leverage relevant knowledge from the generated commonsense inferences, the question and choice embeddings are used to attend over generated paths as:

$$\alpha_p = Softmax(tanh(H_S W^A) h_{US})$$

$$h_{S'} = \sum_{h \in H_S} \alpha_p^h \cdot h$$

MRPC Paraphrase Generation						CSQA	
	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE		Accuracy (%)
T5-base	43.10	36.10	61.80	36.33	47.10	T5-base	53.48
+ CoSe-Co Paths	**44.50**	**36.70**	**62.50**	**37.34**	**48.50**	+ CoSe-Co Paths	**55.11**
CommonGen							
T5-base	9.90	21.10	**36.70**	14.54	**44.70**	T5-large	55.69
+ CoSe-Co Paths	**10.10**	**21.20**	**36.70**	**14.78**	**44.70**	+ CoSe-Co Paths	**57.33**
+ PGQA Paths	9.80	21.0	36.60	14.64	**44.70**		

Table 9: Concatenating paths generated by CoSe-Co leads to improvements in MRPC paraphrase generation and CSQA (cast as generation task) without much performance change on CommonGen. The numbers for T5-base and T5-large might not match with the leaderboard and other prior implementations. Here we report numbers from our implementations, that weren't tuned for the best hyper-parameters. However, we use this same configuration uniformly across all experiments. Generative commonsense methods like PGQA which rely on answer choices cannot be applied in tasks like paraphrase generation where entities are not available.

where, $W^A \in R^{h_D \times h_E}$, $\alpha_p \in R^k$ and $h_{S'} \in R^{h_D}$. Finally, a linear layer is applied over the concatenation of $\{h_{US}, h_{S'}\}$ to project it as a scalar. A softmax is taken over concatenation of scalars obtained corresponding to each answer choice to obtain their likelihood followed by cross entropy loss for training.

E Effect of Concatenating CoSe-Co paths in Generation Tasks

In this section, we explore augmenting CoSe-Co paths for text generation. The aim is not to obtain SOTA results but to analyse if concatenating paths improves performance of a base model. Specifically we study - 1) Paraphrase Generation: given a sentence, generate another sentence expressing same meaning using different words where commonsense is usually helpful in rephrasing while retaining meaning, and 2) CommonGen: generate a sentence describing a scene using a concept-set requiring commonsense to compose concepts. Since T5 (Raffel et al., 2019) is designed for generation tasks, we fine-tune T5-base to generate annotated paraphrase given a sentence as input on MRPC dataset (Dolan & Brockett, 2005). For commongen data (Lin et al., 2020), given concepts 'bird hold cup eat food', T5 is trained to generate sentence like 'The bird eats food from cup that its owner is holding'. Paths generated by CoSe-Co are appended as a string to the input.

Table 9 summarises results evaluated through commonly used generation metrics - BLEU (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). Amongst these, SPICE is considered to correlate most with human judgement. We evaluate on test and dev set of MRPC and CommonGen respectively. Please note that performance might not match with leaderboard since our hyper-parameters might be different. Fine-tuning T5 with CoSe-Co paths results in better paraphrase generation as indicated by $\sim$1-1.5% improvement in most metrics. However for CommonGen, no significant improvements are observed and performance remains similar without any degradation, even though generated paths seem relevant (for concepts 'bird hold cup eat food', CoSe-Co generates 'bird_eating *isa* eating *hassubevent* hold_food *_usedfor* cup'). We believe this could be due to simple path concatenation instead of employing a trainable module to inject paths which paves scope for future work. Additionally, we cast CSQA as generation task where T5 is trained to generate correct answer belonging to choices which are given as input with question. As shown in Table 9, improvements ($\sim$1.5%) are observed on CSQA dev set for both T5-base and T5-large when CoSe-Co paths are concatenated. We would like to highlight that generative commonsense methods like PGQA which rely on answer choices cannot be applied in tasks like paraphrase generation where entities are not available.

For paraphrase generation on MRPC (Dolan & Brockett, 2005) dataset, we fine-tune T5-base (with and without CoSe-Co knowledge) at a learning rate of 5e-4 for 5 epochs with weight decay of 0.01 and 4 gradient accumulation steps using AdamW (Loshchilov & Hutter, 2017) optimizer. The training set of MRPC comprises of 2,661 paraphrases while the test set comprises of 1,088 paraphrases. The dataset has been downloaded from here.

For CommonGen (Lin et al., 2020), we fine-tune T5-base (with and without CoSe-Co knowledge) for 20 epochs at a learning rate of 5e-5, with weight decay of 0.01 on a batch size of 48 with 3

gradient accumulation steps using AdamW (Loshchilov & Hutter, 2017) optimizer. The dataset has been downloaded from here. The train set of commongen comprises of 67,388 concept-set and sentence pairs while the development set comprises of 4,017 concept-set and sentence pairs.

For CSQA cast as a generation task, T5-base and T5-large (with and without CoSe-Co knowledge) are trained for 20 epochs with a learning rate of 5e-4 at a batch size of 8 using AdamW optimizer. The accuracy is estimated by determining if the generated answer matches with the correct answer which is present amongst the choices given as input to the model.

F Further Implementation Details

F.1 Relation Heuristics

As mentioned in §3.1, we employ heuristics on the basis of contained relations to perform filtering of ConceptNet paths. Particularly, we use the following rules:

1. We discard any path that uses the same two relations to connect any three neighbouring entities occurring in it. That is, for any sub-path $\{e_i, r_i, e_{i+1}, r_{i+1}, e_{i+2}\}$ in a given path p, we only consider p as a part of our dataset if $r_i \neq r_{i+1}$.

2. Following (Wang et al., 2020b), we do not consider paths that contain any relations from the set {*HasContext*, *RelatedTo*, *Synonym*, *Antonym*, *DerivedFrom*, *FormOf*, *EtymologicallyDerivedFrom*, *EtymologicallyRelatedTo*}. We observed that entities connected through these relations were often largely dissimilar and thus not useful for our case.

F.2 KG Completion

As discussed in §4.2, we use CoSe-Co for the task of KG completion where given a test triple 'h, r and t', we give h and r as input to CoSe-Co and also condition the decoder with input 'h r' and then generate the next entity. To compute the accuracy, we perform matching between generated entity and ground truth tail entity in the triple. To perform comparison with COMET (Bosselut et al., 2019) we take their code and pre-trained model from here.

F.3 CommonSense QA

In §4.3, we discuss commonsense question answering task where we use framework developed by Wang et al. (2020b) and just replace the commonsense knowledge used by them with the paths generated by CoSe-Co. We use the same hyper-parameters as used by them and mention them here for reference. The model is trained on a batch size of 16, dropout of 0.1 for 15 epochs. A learning rate of 2e-6 is used for encoder LM (Roberta-large) used for embedding question and choice context and an lr of 1e-3 is used for remaining path attention and classification layer parameters. We perform the evaluation on CSQA (Talmor et al., 2019) dataset downloaded from here. The train split comprises of 8,500, dev split contains 1,221 and in-house test split contains 1,241 samples.

F.4 OpenCSR

In this section, we discuss the implementation details used for OpenCSR in §4.4. The dataset has been downloaded from here. The training splits of ARC, QASC, and OBQA datasets comprises of 5355, 6883, and 4199 samples respectively while the development split comprises of 562, 731, and 463 samples respectively. The test set is hidden and authors who proposed the task with reformulated dataset are yet to set up a leaderboard on the hidden test set. They run their proposed model DrFact (which is based on BERT-base and is the current state-of-the-art on this task) on a single seed which takes about $\sim$2-3 days to train one model on a given dataset. While fine-tuning T5-base (with and without CoSe-Co knowledge), we train the model for 5 epochs with a learning rate of 5e-4, weight decay of 0.01 and batch size 8 using AdamW optimizer (Loshchilov & Hutter, 2017).

Paraphrase Detection on MRPC		PIQA		RTE		WiC	
Model	Acc.	Model	Acc.	Model	Acc.	Model	Acc.
BERT-base	84.28	BERT-base	60.22	BERT-base	64.4	BERT-base	59.9
+ CoSe-Co paths	**84.4**	+ CoSe-Co paths	**61.03**	+ CoSe-Co paths	**64.54**	+ CoSe-Co paths	**60.1**

Table 10: Analysing effect of concatenating CoSe-Co paths on more tasks using BERT-base. While performance improvements are observed for PIQA requiring physical common sense, no significant changes are observed for the remaining three tasks.

G PATH AUGMENTATION IN MORE TASKS

In this section, we explore effect of augmenting paths generated by CoSe-Co on others tasks using BERT-base (Devlin et al., 2019). Specifically, we explore the tasks of 1) **Paraphrase Detection** on MRPC dataset (Dolan & Brockett, 2005): given a pair of sentences, the task is to predict whether they are paraphrases of each other; 2) **Physical Interaction QA** (PIQA) (Bisk et al., 2020): given a goal statement (such as 'Extend life of flowers in vase') and a pair of solutions to attain the goal (such as 'sol1: Add small amount of coffee in vase.' and 'sol2: Add small amount of water in vase.'), the model has to predict which solution can lead to the goal requiring physical commonsense; 3) **Recognising Textual Entailment** (RTE) (Poliak, 2020): Given a premise sentence and a hypothesis sentence, determine if hypothesis is implied by the premise or contradicts it; and 4) **The Word-In-Context** (WiC) (Pilehvar & Camacho-Collados, 2019): Given a pair of sentences having a common word, predict if the word has same context in both the sentences or not.

For each task, we fine-tune pre-trained BERT-base to predict the correct answer through binary classification (since each task requires a binary prediction). For all tasks except PIQA, the pair of sentences are concatenated with each other with '[SEP]' token in between, given to BERT as input and a classification layer is used over the output of '[CLS]' token. For PIQA, the goal statement is concatenated with each solution statement separately with '[SEP]' in between and processed through BERT in parallel. The corresponding '[CLS]' token outputs are concatenated and processed through a classification layer. To augment commonsense knowledge, we generate paths corresponding to one text input only (first sentence in case of paraphrase detection, the goal statement in case of PIQA, premise for RTE and first sentence in WiC) and append the paths with the corresponding sentence while giving as input to BERT. We fine-tune BERT with a batch size of [16, 64, 32, 32], for [5, 8, 5, 5] epochs, at a learning rate of [2e-5, 1e-5, 2e-5, 2e-5] for the 4 tasks respectively using AdamW optimizer (Loshchilov & Hutter, 2017).

Table 10 summarises the results obtained on the dev set for the 4 tasks discussed above. It can be seen that CoSe-Co paths lead to improvement of $\sim 0.8\%$ in PIQA task with performance remaining almost unchanged without degradation for the remaining three tasks. We believe that since PIQA requires commonsense reasoning, augmenting CoSe-Co paths lead to improvements which is not the case with other tasks. Even though there are no improvements for paraphrase detection, RTE and WiC, the performance remains same or improves marginally which means there is no degradation. We leave exploring how commonsense paths can be injected in an intelligent manner instead of simple concatenation to further boost performance in such tasks where common sense might not be explicitly helpful as future work.