KG-CRuSE: Recurrent Walks over Knowledge Graph for Explainable Conversation Reasoning using Semantic Embeddings

Rajdeep Sarkar[1], **Mihael Arcan**[2] and **John P. McCrae**[1,2]

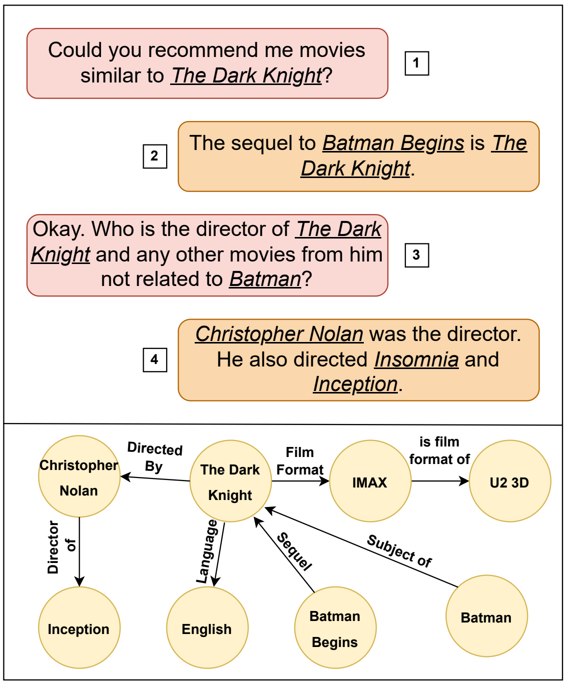
[1] Data Science Institute, National University of Ireland Galway, Ireland
[2] Insight SFI Research Centre for Data Analytics, National University of Ireland Galway, Ireland
{r.sarkar1, mihael.arcan, john.mccrae}@nuigalway.ie

Abstract

Knowledge-grounded dialogue systems utilise external knowledge such as knowledge graphs to generate informative and appropriate responses. A crucial challenge of such systems is to select facts from a knowledge graph pertinent to the dialogue context for response generation. This fact selection can be formulated as path traversal over a knowledge graph conditioned on the dialogue context. Such paths can originate from facts mentioned in the dialogue history and terminate at the facts to be mentioned in the response. These walks, in turn, provide an explanation of the flow of the conversation. This work proposes KG-CRuSE, a simple, yet effective LSTM based decoder that utilises the semantic information in the dialogue history and the knowledge graph elements to generate such paths for effective conversation explanation. Extensive evaluations showed that our model outperforms the state-of-the-art models on the OpenDialKG dataset on multiple metrics.

1 Introduction

Inducing factual information during response generation has garnered a lot of attention in dialogue systems research. While language models (Zhao et al., 2020; Zheng et al., 2020) have been shown to generate responses akin to the dialogue history, they seldom contain factual information, leading to a bland conversation with the agent. Knowledge-grounded dialogue systems focus on leveraging external knowledge to generate coherent responses. Knowledge Graphs (KGs) are a rich source of factual information and can be combined with an utterance generator for a natural and informative conversational flow.



Could you recommend me movies similar to *The Dark Knight*? 1

2 The sequel to *Batman Begins* is *The Dark Knight*.

Okay. Who is the director of *The Dark Knight* and any other movies from him not related to *Batman*? 3

4 *Christopher Nolan* was the director. He also directed *Insomnia* and *Inception*.

Figure 1: An example conversation wherein the agent utilises relevant information from the KG while generating responses. The agent generates facts about "Christopher Nolan" in utterance 4 while utilising the semantic information in the dialogue history and the KG.

Zhou et al. (2018) showed that utilising KGs in dialogue systems improves the appropriateness and informativeness of the conversation. Augmenting utterances in a dialogue with the KG information guides the conversational agent to include relevant entities and facts in the response. For example, Figure 1 shows an example conversation where a user is interacting with a dialogue agent about movies. The agent has access to a KG that aids in suggesting relevant facts during the dialogue flow. When responding to utterance 3, the agent can utilise information from the KG and produce relevant facts about "Christopher Nolan". This information would be more engaging than responding with information about "Batman" or "Batman Begins".

While KGs have been used extensively to include relevant facts in a dialogue, the explicability of such systems is limited. Naturally, this fostered

Proceedings of the 4th Workshop on NLP for Conversational AI, pages 98 - 107
May 27, 2022 ©2022 Association for Computational Linguistics

research on developing models for explainable conversation reasoning. Moon et al. (2019) addressed this problem by inducing KG paths for conversation explainability. They posited a dialogue-KG path aligned corpus wherein utterances are augmented with a KG path to denote fact transitions in the dialogue. The KG paths emanate from entities or facts mentioned in the dialogue history and terminate at the entity to be mentioned in the response text. Such paths form a sequence of entities and relations and aid the dialogue agent in introducing appropriate knowledge to the dialogue. In addition to this, they proposed an attention-based recurrent decoder over the KG to generate entity paths. Jung et al. (2020) designed a novel dialogue-context infused graph neural network to propagate attention scores over the knowledge graph entities for KG path generation. While such approaches have their inherent strengths, their limitations are manifold.

Given a dialogue context, it is desirable to generate paths that results in a natural dialogue flow. Therefore it is essential to capture the semantic information in the dialogue context as well as the KG elements. Transformer based models (Devlin et al., 2019; Lan et al., 2020; Liu et al., 2019a) have enabled the capture of contextual relationships between different words in a sentence. Textual representations from such models have been successfully adapted for the dialogue conditioned KG reasoning task (Jung et al., 2020). However, prior works use the embedding of the $[CLS]$ token to encode the dialogue history and the KG elements. Reimers and Gurevych (2019) demonstrated that such sentence embeddings are sub-optimal and lead to degraded performance in downstream application tasks. Sentence-transformers (Reimers and Gurevych, 2019) are strong tools for capturing the semantic information of a sentence into a fixed-size vector. As KG elements can be long phrases, KG-CRUSE uses the Sentence-BERT (SBERT) model to encode both the dialogue history and the KG elements for capturing their semantic information.

As a result of the long tailed distribution of node neighbors in a KG, it can become difficult to generate relevant paths over the KG for explainable conversation. Given the dialogue history, it is desirable to traverse paths that are semantically relevant. KG-CRUSE utilises the rich sequential information in the dialogue history and the path history to sample the top-k semantically similar neighbors for extending its walk over the KG.

We show that our KG-CRUSE improves upon the current state-of-the-art on multiple metrics, demonstrating the effectiveness of KG-CRUSE for explainable conversation reasoning.

To summarise, our contributions are as follows:

- We propose a KG-CRUSE , a LSTM based decoder leveraging Sentence-Transformer (S-BERT) embedding to reason KG paths for explainable conversation.
- We show the efficacy of our model by improving the current state-of-the-art performances over multiple metrics on the OpenDialKG (Moon et al., 2019) dataset. Additionally, we conduct extensive empirical analysis to emphasise the effectiveness of KG-CRUSE for the reasoning task.
- We release[1] our system and baseline systems as an open-source toolkit to allow reproducibility and future comparison on this task.

2 Related Work

The use of external knowledge in dialogue agents has become commonplace, owing to the rich heterogeneous information contained in them. He et al. (2017) addressed the knowledge-grounded conversation task by iteratively updating the knowledge base embeddings to generate informative responses. Following this, knowledge-based dialogue systems have been studied extensively including the collection of new knowledge-grounded datasets (Ghazvininejad et al., 2018; Qin et al., 2019; Zhang et al., 2018) and developing knowledge-centric dialogue systems (Liu et al., 2018; Parthasarathi and Pineau, 2018a; Zhang et al., 2020).

Young et al. (2018a) attempted to integrate a large scale KG into an end-to-end dialogue system. Other similar works (Chen et al., 2019; Zhou et al., 2020; Sarkar et al., 2020) leveraged graph neural networks and KG embeddings to recommend relevant products in conversational recommender systems. Though successful in retrieving suitable entities or facts from the KG, these systems fail to provide explainability to the recommendations.

Such limitations encouraged explainable conversation reasoning using external knowledge. Liu et al. (2019b) develop the problem as a Partially

[1] https://github.com/rajbsk/kg-cruse

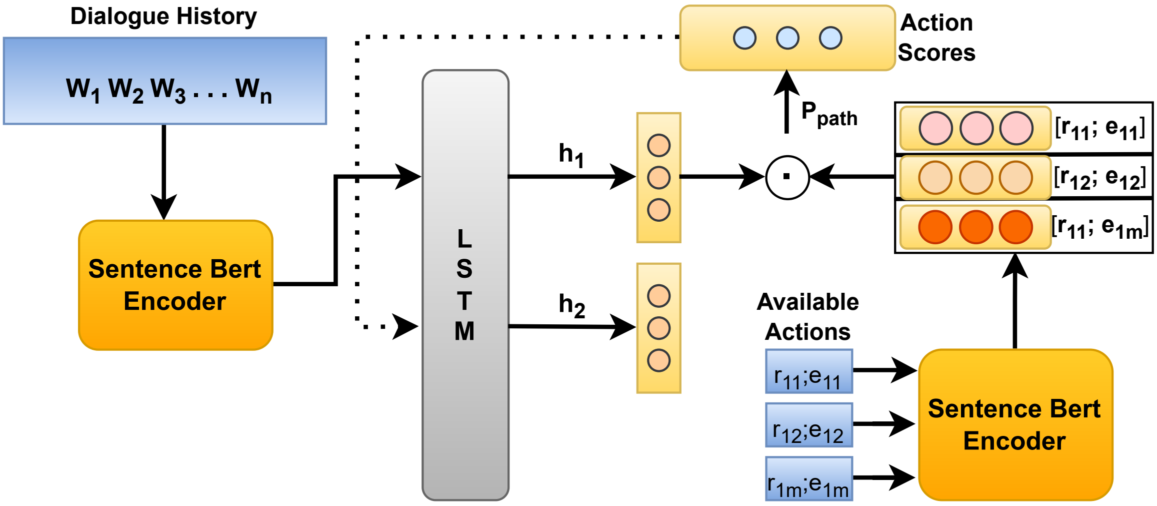


Figure 2: Modular overview of KG-CRUSE architecture. KG-CRUSE utilises the SBERT architecture to encode the dialogue history and the KG elements. To generate walk paths over the KG, KG-CRUSE leverages an LSTM network to model the temporal information. To generate the path at timestep t, the LSTM takes as input ($\mathbf{D}$, ($\mathbf{r_1}; \mathbf{e_1}$), ..., ($\mathbf{r_{t-1}}; \mathbf{e_{t-1}}$)) and outputs the hidden state representation $\mathbf{h_t}$ of the step t. KG-CRUSE then computes dot-product of $\mathbf{h_t}$ with the embeddings of the actions available at timestep t ([$\mathbf{r_{t,1}}; \mathbf{e_{t,1}}$], [$\mathbf{r_{t,2}}; \mathbf{e_{t,2}}$], ...,[$\mathbf{r_{t,m}}; \mathbf{e_{t,m}}$]) followed by a softmax layer to compute the probability of each available action.

Observable Markov Decision Process and use policy gradient for training the agent to generate KG paths. Moon et al. (2019) posited a KG path-parallel-dialogue corpus along with DialKG Walker (DKGW) model, a recurrent decoder model to generate the KG path for a response entity selection. Jung et al. (2020) suggested the use of graph neural networks using attention flow to generate KG entity paths. While novel, DKGW does not explicitly utilise the graph structure during model training. On the other hand, the performance of AttnIO (Jung et al., 2020) relies on the node sampler during training. AttnIO becomes computationally expensive due to dialogue specific graph neural network (both during training and inference) as the model concatenates the dialogue embedding to the node embeddings while propagating attention scores. To counter these issues, we design a very simple, lightweight, yet efficient LSTM network leveraging the dialogue and path history to extend the path over the KG.

While, DKGW uses TransE (Bordes et al., 2013) for encoding the elements of the KG, such translation embeddings have weak representation capacity. On the other hand, Jung et al. (2020) utilise the ALBERT (Lan et al., 2020) representation of sentence to encode the dialogue history and the KG elements. They use the $[CLS]$ token representation of the text sequence as the sentence representation. Reimers and Gurevych (2019) suggested Sentence-Transformers for encoding sentences. We encode the dialogue history and the KG elements using Sentence-Transformers to capture rich semantic similarities between the dialogue history and the KG elements.

The processing of semantically rich sequential information using a lightweight LSTM model makes KG-CRUSE ideal for generating walks over a KG for explainable conversation.

3 Methodology

In the following sections, we begin with formally introducing the problem statement. We then outline the embeddings used in KG-CRUSE. Following this, we discuss the architecture of KG-CRUSE as illustrated in Figure 2. Finally, we describe decoding process used by KG-CRUSE during the inference step.

3.1 Formal Problem Definition

We describe the problem statement similar to Moon et al. (2019). The KG is defined as $\mathcal{G} = \mathcal{V}_{\mathcal{KG}} \times \mathcal{R}_{\mathcal{KG}} \times \mathcal{V}_{\mathcal{KG}}$, where $\mathcal{V}_{\mathcal{KG}}$ is set of entities and $\mathcal{R}_{\mathcal{KG}}$ is set of relations in the KG. Facts in the KG are denoted by triples, and each has the form (*e, r, e'*) where *e, e'* $\in \mathcal{V}_{\mathcal{KG}}$ are entities and *r* $\in \mathcal{R}_{\mathcal{KG}}$ is the relation connecting them.

In addition to the KG, each input contains a dialogue D $\in \mathcal{D}$, represented as a sequence of utterances D = $\{s_1, ..., s_n\}$, and the set of entities $\mathrm{x_e} = \{\mathrm{x}_e^{(i)}\}$ occurring in the user's last utterance s_n, where $\mathrm{x}_e^{(i)} \in \mathcal{V}_{\mathcal{KG}}$. The output is represented as $\mathrm{y} = \{\mathrm{y_e}, \mathrm{y_r}\}$, where y_e is a set of entity paths $\mathrm{y_e} = \{\mathrm{y}_e^{(i)}\}$, with each element $\mathrm{y}_e^{(i)} = \{\mathrm{y}_{e,t}^{(i)}\}_{t=1}^{T}$ denoting an entity path connecting $\mathrm{x}_e^{(i)}$ to the response entity $\mathrm{y}_{e,T}^{(i)}$. Likewise, $\mathrm{y_r} = \{\mathrm{y}_r^{(i)}\}$ is a set of relation paths, where $\mathrm{y}_r^{(i)} \in \mathcal{R}_{\mathcal{KG}}$. The element $\mathrm{y}_r^{(i)} = \{\mathrm{y}_{r,t}^{(i)}\}_{t=1}^{T}$ is a sequence of relations from the KG connecting $\mathrm{x}_e^{(i)}$ and $\{\mathrm{x}_{e,t}^{(i)}\}_{t=1}^{T}$.

3.2 Dialogue and KG Representation

Capturing the semantic information in the dialogue context is an important component of our model. SBERT is a contextual sentence encoder that captures the semantic information of a sentence in a fixed-size vector. We encode pieces of text using Equation 1. The text is first sent though a pretrained BERT model to obtain the contextual representation of its tokens. The sentence embedding is computed by taking a mean-pool of the contextual token representations. The dialogue context is constructed by concatenating a maximum of three previous utterances and is then passed through SBERT encoder to obtain a fixed-size contextual dialogue representation.

$$\mathbf{S} = \mathrm{MeanPooling}(\mathrm{BERT}(S)) \qquad (1)$$

In order to align the semantic vector space of the dialogue representations and the KG representations, we use SBERT to encode the KG elements. As KG entities and relations can be words or phrases, SBERT can effectively capture their semantic information. We use the publicly available SBERT-BERT-BASE-NLI[2] model with mean-pooling as our SBERT encoder.

3.3 KG-CRUSE Architecture

KG-CRUSE learns to traverse a path on the KG by learning a function π_θ that calculates the probability of an action $a_t \in \mathcal{A}_t$ given the state s_t. The state s_t contains the dialogue history and entities already traversed by KG-CRUSE while decoding the paths, while a_t is the set of edges from the KG available to KG-CRUSE for extending its path.

The state s_t at step t is defined as a tuple (D, (r_1, e_1, ..., r_{t-1}, e_{t-1})), where D is the dialogue context and $r_i, e_i (i < t)$ are the relation and entity already decoded by KG-CRUSE at step i. The initial state s_0 is denoted as (D, $\varnothing$), where $\varnothing$ is the empty set.

At step t, an action has the form $a_t = (r_t, e_t) \in \mathcal{A}_t$, where $\mathcal{A}_t$ is the set of all possible actions available to the model at step t. $\mathcal{A}_t$ includes all outgoing edges of e_{t-1} in the KG $\mathcal{G}$, i.e. $\mathcal{A}_t$ is the set of all the outgoing edges of the entity decoded by KG-CRUSE at timestep $t - 1$. To let the agent terminate the search process, we add self-loop edges to every entity node in the graph denoting no operation ("self-loop"). The action a_t is represented as a concatenation of the relation and entity embedding $\mathbf{a}_t = [\mathbf{r}_t; \mathbf{e}_t]$, where $\mathbf{r} \in \mathbb{R}^{d_r}$, $\mathbf{e} \in \mathbb{R}^{d_e}$ and $\mathbb{R}^{d_e}$, $\mathbb{R}^{d_r}$ are the size of the entity embedding and relation embedding respectively. At step 1, KG-CRUSE chooses between the entities mentioned in s_n for path traversal. The relation associated with action at step 1 is the zero vector. As mentioned, the state s_t contains the dialogue context and action history (path history). This sequential information in s_t is modelled using an LSTM:

$$\mathbf{d} = \mathbf{W}_d\mathbf{D} \qquad (2)$$

$$\mathbf{h}_0 = \mathrm{LSTM}(\mathbf{0}, \mathbf{d}) \qquad (3)$$

$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{h}_{t-1}, \mathbf{a}_{t-1}), t > 0 \qquad (4)$$

where $\mathbf{D}$ is the contextual dialogue embedding obtained using Equation 1 and $\mathbf{W}_d$ is a learnable matrix that maps the dialogue embedding to the LSTM input dimension. Given the hidden state representation $\mathbf{h}_t$ at time t, KG-CRUSE assigns a probability to each action using Equation 6.

$$\mathbf{x_t} = \mathbf{W}_{3,\theta}(\mathrm{ReLU}(\mathbf{W}_{2,\theta}\mathbf{h}_t^{\intercal})) \qquad (5)$$

$$\pi_\theta(a_t|s_t, \mathcal{A}_t) = \frac{\exp(\mathbf{a}_t \cdot \mathbf{x}_t)}{\sum_{a_i \in \mathcal{A}_t} \exp(\mathbf{a}_i \cdot \mathbf{x}_t)} \qquad (6)$$

The hidden state representation $\mathbf{h}_t$ is passed through a two-layered dense network with ReLU activation (Nair and Hinton, 2010) in the first layer. The LSTM weights, $\mathbf{W}_{2,\theta} \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_{3,\theta} \in \mathbb{R}^{(d_r+d_e) \times d_h}$ are the learnable parameters, and d_h is the LSTM hidden representation size.

3.3.1 Model Learning

We train KG-CRUSE by minimising the cross-entropy loss on the entities decoded at each timestep. Additionally, we train the model using

[2] https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

teacher forcing (Sutskever et al., 2014), wherein the model makes each action conditioned on the gold history of the target path. To prevent overfitting, we add L_2 regularisation to the parameters of the model. During training, we do not fine-tune the SBERT architectures, but back-propagate the gradients to the entity and relation embeddings.

3.3.2 KG-CRUSE Path Generation

Once the model is trained, KG-CRUSE takes the dialogue history and the entities mentioned in the current utterance as input, a horizon T and outputs a set of entity paths, relations paths of length T along with the probability score of each path. During inference, we remove self-loops from the KG except for the self-loop with label "self-loop" introduced in section 3.3. We do so to allow the agent traverse diverse paths rather than staying at entities mentioned in the dialogue history.

4 Experimental Setup

This section presents the dataset used, the baselines compared with and the description of the model settings of KG-CRUSE along with the metrics the models have been evaluated on.

4.1 Dataset

We evaluate our proposed framework on the OpenDialKG dataset (Moon et al., 2019). The dataset has 91,209 turns spread over 15,673 dialogues in the form of either task-oriented dialogues (recommendations) or chit-chat conversations on a given topic. Each turn is annotated with a KG path to represent fact transitions in the conversation. The KG is a subset of the Freebase KG (Bollacker et al., 2008), which has 1,190,658 fact triples, 100,813 entities and 1,358 relations. Following Moon et al. (2019), we split the dataset randomly into 70%, 15% and 15% for training, testing and validation.

4.2 Baselines and Evaluation Metrics

We compare KG-CRUSE against the following baseline models suggested by Moon et al. (2019) and Jung et al. (2020):

- Tri-LSTM (Young et al., 2018b): The model encodes each utterance along with facts from the KG within 1-hop distance from the entities mentioned in the current utterance. This is used to retrieve facts from the KG for dialogue explanation.
- Ext-ED (Parthasarathi and Pineau, 2018b): Moon et al. (2019) conditioned the response generation with external knowledge vector input to generate response entity token at the final softmax layer, without using the structural information from the KG.
- Seq2Path (Jung et al., 2020): An attention based Seq2Seq model is modified to generate entity paths by masking out unreachable nodes at each decoding step.
- Seq2Seq: An LSTM based seq2seq (Sutskever et al., 2014) model where the decoder is modified to generate entity paths. Similar to DKGW (Moon et al., 2019) model, we use modality attention as the output of the encoder. Following Moon et al. (2019), we replace the softmax layer in the decoder with a zero-shot learning layer in the KG embedding space.
- DKGW (Moon et al., 2019): A model to generate KG paths using domain-agnostic, attention-based recurrent graph decoder reinforced with a zero-shot learning layer over the KG embedding space.
- AttnIO (Jung et al., 2020): A dialogue conditioned KG path traversal leveraging attention flow using graph neural networks.

Since the authors of OpenDialKG and AttnIO have not released their implementations, we report their performance on our re-implementations. We note that for most systems, our implementation is similar or better than the reported results. Regarding AttnIO, we were not able to reproduce the results although we note that errors in the implementation of the node sampler or leakage of the test dataset into the training dataset can easily lead to overestimation of the accuracy. The code and dataset used for re-implementation as well as our system is accessible at `https://github.com/rajbsk/kg-cruse`.

We evaluate our models on different recall@k metrics for entity and path retrieval. Path@k measures if the ground-truth path is present in the top-k paths with the highest probability searched by the agent. Similarly, tgt@k measures if the response entity is present in the top-k entities retrieved by the agent. In situations where multiple paths point to the same response entity, we consider the path with the highest score for entity retrieval.

Model	Recall@k							
	path@1	path@5	path@10	path@25	tgt@1	tgt@5	tgt@10	tgt@25
Tri-LSTM	3.2	22.6	36.3	56.2	-	-	-	-
Ext-ED	1.9	9.0	13.3	19.0	-	-	-	-
Seq2Path	14.92	31.1	38.68	48.15	15.65	33.86	42.52	53.28
Seq2Seq*	6.53±0.78	26.21±1.21	35.02±1.27	45.78±1.18	7.13±0.85	30.64±1.62	41.01±1.43	52.97±1.55
DKGW*	14.16±1.16	37.26±1.91	47.85±2.60	59.20±2.33	14.96±1.04	39.53±1.81	51.06±2.15	63.85±1.58
AttnIO*	19.08±1.19	38.49±0.79	43.99±1.10	48.94±0.55	**20.32±1.80**	45.90±0.93	52.82±0.65	55.17±0.96
KG-CRUSE	**19.59±0.43**	**44.62±1.08**	**56.16±1.21**	**70.59±0.38**	20.20±0.36	**47.76±0.62**	**60.11±0.92**	**75.30±0.57**

Table 1: Performance of KG-CRUSE in comparison with other baseline methods on different Recall@k metrics. The numbers reported are the mean values with the sample standard deviation (p=0.01). Results are statistically significant with p=0.01. Models with * denote our re-implementation.

4.3 Implementation Details

For the task, we set horizon T to 3. The dialogue, entity and relation embeddings are encoded using SBERT into a 768 dimensional vector. In KG-CRUSE, we consider 3 LSTM layers with $d_h = d_e + d_r = 1,536$. To prevent the agent from overfitting on the dataset, we add L_2 regularisation with a weight decay parameter of 1e-3.

Similar to Jung et al. (2020), we set the batch size to 8 and train the model with Adam optimiser (Kingma and Ba, 2015) with a learning rate of 1e-4 for 20 epochs. For models with re-implementations, we report the results on five different splits of the data. For Tri-LSTM and Ext-ED, we report the number reported by Moon et al. (2019), while for Seq2Path, the numbers are reported from the work of Jung et al. (2020). As entity occurrences in a dialogue dataset is sparse, it is desirable to report the performance on five different splits of the data rather than an assessment of five models on one split.

5 Results and Discussion

We begin with performing a quantitative evaluation of the models. Following this, we study the impact of our choice of sentence embeddings on the model performance. Then we analyse the impact of beam-width at each decoding step during inference. Finally, we provide insights of examples where the results of KG-CRUSE are not consistent with the ground truth paths.

5.1 Quantitative Analysis

In this section, we compare the performance of our proposed approach against the different baselines. From Table 1, it can be observed that KG-CRUSE performs better than the different baseline models on the OpenDialKG dataset. For entity and path accuracy, AttnIO has the closest performance compared to our model, with the latter being 2.7% relatively better on both path@1 and entity@1 metrics. On increasing k of recall@k, we find KG-CRUSE has at least 10% relative improvement over the baseline models. It is interesting to notice that on increasing the value “k”, KG-CRUSE performs relative better than other models. KG-CRUSE identifies paths semantically relevant to the dialogue context although different from the gold-label paths as discussed in Section 5.6. The huge gain on the path@25 metric advocates for this hypothesis. It is worthwhile to notice that although AttnIO has the closest performance for path@1 and entity@1 to KG-CRUSE, the performance degrades when “k” increases in path@k and entity@k. This might be due to the fact that the beam-width reported by the authors is not expressive enough to capture semantically relevant paths or entities.

Model	P@1	P@25	E@1	E@25	Rel@1
BERT	12.74	66.72	12.98	72.14	39.37
ALBERT	13.42	65.67	13.96	72.23	**40.93**
SRoBERTa	17.17	68.04	17.65	73.34	40.71
SBERT	**19.52**	**70.72**	**20.20**	**75.72**	40.02

Table 2: Influence of sentence embeddings on KG-CRUSE performance. Comparison of different embedding methods.

5.2 Effectiveness of Sentence Embeddings

In our framework, we utilise sentence SBERT embeddings to encode dialogue context and KG elements. In this section, we conduct an ablation study on the efficacy of such embeddings. We replace the SBERT embeddings with the *[CLS]*

SBERT Fine-tuned	Aligned KG	P@1	P@25	E@1	E@25	Rel@1
Yes	No	17.82	69.47	18.21	74.47	40.24
Yes	Yes	18.46	69.93	19.00	74.75	40.47
No	No	18.00	62.01	18.52	74.54	38.48
No	Yes	**19.52**	**70.72**	**20.20**	**75.72**	**40.02**

Table 3: Results on fine-tuning the SBERT architecture used for encoding the dialogue history. Additionally, the table reports the results of initialising the KG elements with random initialisation.

token representation of BERT[3] (Devlin et al., 2019) and ALBERT model[4] (Lan et al., 2020) in KG-CRUSE. Additionally, we consider an instance wherein the elements are encoded using Sentence-RoBERTa (SRoBERTa)[5] (Reimers and Gurevych, 2019). The results in Table 2 demonstrates the strength of our embedding choices wherein SBERT and SRoBERTa outperforms the BERT and ALBERT embeddings. Both the sentence embedding models models are pre-trained on NLI datasets, which allows them to capture rich semantic information for textual similarity. These embeddings have demonstrated strong performances in the task of semantic search using cosine-similarity (Reimers and Gurevych, 2019). It should be noted that before the softmax layer in KG-CRUSE, we compute the dot product of the LSTM layer hidden representation with that of the relation-entity embeddings available at the given timestep. As a result of this step, we expect the performance of SBERT and SRoBERTa embeddings to be better that BERT and ALBERT embeddings.

Additionally, we see from the Table 2 that the relation accuracy of different models is higher than path accuracy. This is due to the outgoing edges of a node (from the dialogue history) sharing similar features if they are connected using the same relation. Thus multiple entities can fit our choice of the response entity given the dialogue context.

5.3 Impact of KG Embedding Alignment and SBERT Fine-tuning

In this section, we study the impact of encoding KG elements with SBERT embeddings. Additionally, we analyse if fine-tuning the SBERT architecture used for encoding the dialogue history is beneficial for explainability.

Size	P@1	P@25	E@1	E@25
2, 5, 50	**19.59**	56.26	**20.09**	62.19
2, 10, 25	19.55	64.93	20.04	70.16
2, 10, 50	19.55	64.93	20.04	70.18
2, 25, 10	19.52	69.75	20.02	74.57
2, 25, 25	19.52	**70.72**	20.02	75.72
2, 25, 50	19.52	**70.72**	20.02	**75.75**
2, 50, 5	19.52	68.46	20.02	72.53
2, 50, 25	19.52	70.56	20.02	75.43

Table 4: Impact of the beam width at different timesteps on the model performance. The results are reported on one of the dataset split. Best results are shown in bold, while the results on the default setting of KG-CRUSE are underlined. All numbers are in percentage.

Model	GPU	Train Time	Test Time
Seq2Seq	Nvidia 1080Ti	≈8 mins	≈1 mins
DKGW	Nvidia 1080Ti	≈4 mins	≈8 mins
AttnIO	Tesla V100	≈38 mins	≈82 mins
KG-CRUSE	Nvidia 1080Ti	≈7 mins	≈8 mins

Table 5: Analysis of the time required by different models for training and inference on the OpenDialKG dataset. The numbers in the third column denote per epoch train time.

Table 3 outlines four situations, where in two cases we fine-tune the SBERT architecture used for encoding the dialogue history. We also consider two cases where the embeddings of the KG elements are initialised with values drawn from a normal distribution with mean 0 and standard deviation 1, corresponding to the value "No" in the second column. It should be noted that we never consider fine-tuning the SBERT architecture used for encoding the KG elements.

We see from the Table 3 that in cases when the KG elements are not encoded with SBERT embeddings, their performance drops as compared to cases when we use SBERT embeddings. Additionally, we find that fine-tuning SBERT leads to a decrease in the performance of KG-CRUSE. This can be attributed to the change in semantic space of the dialogue embeddings and the KG embeddings during fine-tuning. Hence, we do not finetune the SBERT architecture in the default setting of KG-CRUSE.

5.4 Impact of Beam-Width on Path Reasoning

In this experiment, we study the influence of beam-width at different timesteps on the model perfor-

[3] https://huggingface.co/bert-base-uncased

[4] https://huggingface.co/albert-base-v2

[5] https://huggingface.co/sentence-transformers/roberta-base-nli-mean-tokens

Dialogue	Model	Walk Path
A: Could you recommend movies similar to Kung Fu Panda? B: [response]	KG-CRUSE Ground Truth	Kung Fu Panda→written by→Cyrus Voris Kung Fu Panda→directed by→Mark Osborne→wrote→Monsters vs. Aliens
A: Oh cool, I also read Wocket in my Pocket! But sure, what else is there? B: Cool! Yertle the Turtle and Horton Hears a Who! are also written by Dr. Seuss. A: That first one is really old right? I think it was released in 1958. // B: [response]	KG-CRUSE Ground Truth	1958→released year→Tom's Midnight Garden Garden→has genre→Children's literature 1958→released year→Have Space Suit - Will Travel→written by →Robert A. Heinlein
A:Could you recommend a book by Jeffrey Zaslow? B: [response]	KG-CRUSE Ground Truth	Jeffrey Zaslow→wrote→The Last Lecture Jeffrey Zaslow→wrote→Last Lecture →has genre→Non-fiction

Table 6: Examples where KG-CRUSE generates path different from the true paths.

mance. The first column of Table 4 lists the tuples (K_1, K_2, K_3) where each K_i denotes the top-K_i edges sampled at timestep i.

We conduct this analysis on a single split of the dataset keeping all other parameters of the model constant. We consider a diverse set of values for each K_i. From Table 4, we find that although the tuples (2, 5, 50), (2, 10, 25), (2, 25, 10) and (2, 50, 5) have an equal number of sampled paths, tuple (2, 25, 10) performs better than others. Interestingly, we observe that the sampling sizes at the second timestep play a significant role in finding optimal paths for KG-CRUSE. The first two sets of fact selection (i.e. during timesteps 1 and 2) largely determine the facts reachable by KG-CRUSE. Sampling more samples during the initial timesteps enables the agent to explore diverse paths initially and KG-CRUSE then makes an optimal selection of facts dependent on the dialogue information.

5.5 Analysis of Computational Requirements

In this study, we conduct an analysis of the time required for training the model. We also compare the performance of different architectures with regards to the inference speed.

Table 5 shows that while DKGW has a better train time per epoch than KG-CRUSE and Seq2Seq has a better inference speed than KG-CRUSE, we can observe from Table 1 that our model achieves better performance compared to these models. It is worthwhile to mention that while AttnIO achieves the closest performance to KG-CRUSE as shown in Table 1, it requires roughly six times more training time and is ten times slower during inference. This clearly indicates the benefits of using KG-CRUSE for explainable conversation using KGs.

5.6 Qualitative Analysis

This section highlights three scenarios showcasing the underlying working of KG-CRUSE. Table 6 displays three examples where KG-CRUSE generates paths different from the gold KG paths. In the first example, it can be observed that KG-CRUSE identifies a path that is not sufficient to answer the given question. This can be due to the limited dialogue context provided. Choosing this fact might lead to a dialogue with the agent, however, the user query is not answer with the path chosen by KG-CRUSE.

In the second example, the relation traversed by KG-CRUSE is correct, however as the dialogue context is not specific, it decodes a path that might potentially fit the dialogue context but is different from the gold path. However, in the third example, even with limited context, KG-CRUSE identifies a path relevant to the context, however the final entity differs from the gold path. Such paths are admissible as all of them fit the dialogue context appropriately.

6 Conclusion

In this work, we propose KG-CRUSE, an LSTM based lightweight framework for explainable conversational reasoning. We utilise SBERT embeddings to capture the rich semantic information in the dialogue history and the KG elements. We conduct an extensive evaluation to demonstrate that our framework outperforms several baseline models on both explainability and response entity retrieval. As annotating ground truth paths is expensive, we plan on extending this model to scenarios when ground truth paths are not available involving the generation of knowledge-conditioned dialogue.

7 Acknowledgements

This work is supported by a grant from The Government of Ireland Postgraduate Fellowship, Irish Research Council under project ID GOIPG/2019/3480. The work is also co-supported by Science Foundation Ireland under grant number SFI/12/RC/2289 2 (Insight).

References

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1803–1813.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5110–5117.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1766–1776.

Jaehun Jung, Bokyung Son, and Sungwon Lyu. 2020. Attnio: Knowledge graph exploration with in-and-out attention flow for knowledge-grounded dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3484–3497.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1489–1498. Association for Computational Linguistics.

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

Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019b. Knowledge aware conversation generation with explainable reasoning over augmented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1782–1792.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 845–854.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress.

Prasanna Parthasarathi and Joelle Pineau. 2018a. Extending neural generative conversational model using external knowledge sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 690–695.

Prasanna Parthasarathi and Joelle Pineau. 2018b. Extending neural generative conversational model using external knowledge sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 690–695. Association for Computational Linguistics.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5427–5436.

Nils Reimers and Iryna Gurevych. 2019. 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*, pages 3980–3990.

Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John P. McCrae. 2020. Suggest me a movie for tonight: Leveraging knowledge graphs for conversational recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4179–4189.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018a. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4970–4977.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018b. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4970–4977. AAAI Press.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2031–2043. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213.

Xiangyu Zhao, Longbiao Wang, Ruifang He, Ting Yang, Jinxin Chang, and Ruifang Wang. 2020. Multiple knowledge syncretic transformer for natural dialogue generation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 752–762. ACM / IW3C2.

Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020. A pre-training based personalized dialogue generation model with persona-sparse data. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9693–9700. AAAI Press.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4623–4629. ijcai.org.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1006–1014.