Knowledge Graph Grounded Conversation Generation

Maria Diea
maria.diea@student.uva.nl
Student ID: 12593125

Diana Epureanu
dianamaria.epureanu@student.uva.nl
Student ID: 12710199

Teodora Stoenescu
teodora.stoenescu@student.uva.nl
Student ID: 12598291

ABSTRACT

In the interest of building human-like conversations, dialogue systems should show a proactive behavior, introducing new entities and attributes that are relevant to the conversation context. Knowledge graphs can be integrated into the response generation process to ensure topic exploration. In this paper, we address the problem of sampling relevant entities from a knowledge graph using the relationships between entities and the conversation history. We propose a classification model that learns to identify relevant entities from a pre-existent dialog dataset annotated with the entities present in the conversation and the relationships that connect them. Given a single-turn dialog line, the classifier predicts natural entities to be introduced which are then fed to a state-of-the-art transformer model used to generate a response. As several possible answers can be generated based on each predicted entity, a ranker model is used for extracting the best response. Automated and manual evaluations suggest that the proposed model obtains similar, even better results than the used baselines. While there is room for further development of the architecture, we show that transformers have great potential in enhancing prediction performance in conversational Q&A knowledge grounded dialogues.

1 INTRODUCTION

Conversational Artificial Intelligence necessitates the ability to have articulated and human-like conversations. Modern generative models such as the recently released OpenAI GPT-3 model [credit] are said to be so qualitative that the generated text is difficult to distinguish from that written by a human. Table 1 is a generation example from DialoGPT [9], a large-scale pretrained dialogue response generation model for multiturn conversations based on OpenAI GPT-2 [4].

User	Have you seen Iron Man?
Bot	I'm not sure what you're talking about. I'm pretty sure that's a woman.
User	I like Iron Man.
Bot	I like turtles.

Table 1: A conversation between a *User* and a *Bot* where the *Bot*'s responses are generated by DialoGPT.

The generated responses are comparable to human response quality, but the conversation falls flat. Dialog systems need to both understand the conversational context and to generate natural responses to keep the interaction engaging and coherent by introducing relevant new entities and attributes. For example, when talking about *Iron Man*, the conversation could naturally go towards *Marvel* or *Robert Downey Jr.*, because to us, as humans, the relationship between these topics is obvious and we know that it is suitable for the discussion to flow towards that direction. Dialog agents usually lack this explicit background knowledge, but this can be circumvented by enhancing the system with the vast information present in a knowledge graph (KG). Fundamentally, a KG encompasses data about entities and the relationships that connect them. By recognizing the subject entity of a conversation, a list of possible candidates to build a response upon could trivially be the neighbours of said entity.

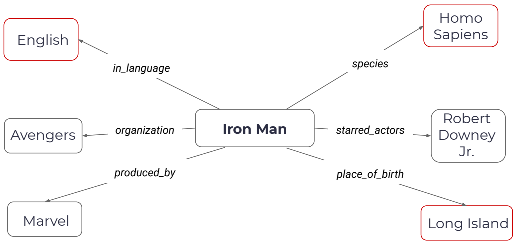

Figure 1: An example of an extract from a Knowledge Graph.

While a KG incorporates data about an entity, the entities related to it and their relationships, it is not sufficient to determine the relevancy of the related entities to the conversational context. Coming back to our example and the KG in Figure , we would not shift the conversation towards *English*, the language the Iron Man movie, since most likely it has low contextual relevancy. Thus, a mechanism must be set in place that determines whether a candidate entity would make for a relevant subject for the generated response.

There exist conversation datasets annotated with knowledge graph for natural grounded response generation. Moon et al. (2019) introduce a parallel open-ended dialog - KG corpus where each entity in the dialog is linked with its corresponding KG path. These path are generated using an attention-based graph decoder using BiLSTMs and a re-ranking model. Chaudhuri et al. (2019) also propose a method based on a KG in the domain of soccer, but to extract relevant entities they employ a special gating mechanism. Both approaches make use of techniques that are behind the current state-of-the-art. In this paper we would like to explore whether we can build upon these results and improve them by using modern advanced architectures such as a transformer model.

Our contribution can be summarized as an end-to-end process. Given an input dialog context and a set of candidate entities, a classifier decides on the relevancy of each candidate. A pretrained dialogue response generation model takes the dialog context and generates a response for each of the top relevant entities. These responses are evaluated and re-ranked using a pretrained dialog ranking transformer that in the end will output the best response.

With this in mind, we first build a BERT binary classifier that predicts whether an entity is relevant given a single dialog context and its relation to the subject. The intuition behind this is that the classifier should be able to learn what features make an entity appropriate to include as the response, with the relationship between this entity and the one in the dialogue context playing an important role. Nonetheless, the model should not rely solely on the relationship to make predictions as this might not be a reliable parameter. For example, even though the language of *Iron Man* is usually not suitable, when talking about the *The Intouchables* movie one might change the subject naturally to *French*.

To train the model, we make use of the OpenDialKG dataset to extract relevant entities and merge each dialog turn with the relation and object of its corresponding KG path, resulting in approximately 8,000 positive samples. In order to have a balanced dataset, negative examples must also be sampled. In the end, the classifier can be used together with the response generation model to produce a natural articulated reply. We explore several methods to include the predicted relevant entities in the response generated by DialoGPT.

2 RELATED WORK

Conversational dialogues between humans and systems have a long history in the development of Artificial Intelligence. A great amount of research has been conducted for improving the performance of generally passive response behaviours and leaving the area of proactive response behaviours less covered.

Our related work covers one major research topic, namely *Knowledge Grounded Conversation* for engaging and meaningful discussions. [1] caters the problem of generating responses for non-goal oriented dialogue systems. They present a new dataset in the domain of soccer containing 2,990 non-goal oriented conversations concerning various clubs and national teams. To build this conversational dataset, a user starts a conversation based on a given topic, such as a national or a club team. The other agent replies and it is prompted to use Wikipedia to answer questions. Detached from the conversational dataset, the authors also propose a soccer knowledge graph consisting of $(subject, relation, object)$ triples extracted from Wikipedia. As a baseline approach on the data, the authors propose the *KG Copy* model which is a sequence-to-sequence encoder-decoder based neural network model able to copy facts from the KG. The model can handle factoid questions, chit-chats and opinions by generating responses. One of the main drawbacks of the proposed model is that it can only respond to simple factoid questions based on embedding similarities between the context and the KG entities. There is room for improvement in linking the entities and their relationships to the query context. The dataset also includes poorly articulated responses as the authors do not filter out substandard conversations.

[6] introduces the *DuConv* dataset containing 30,000 chat conversations with 270,000 turns. The conversations are generated using a specific goal: one agent is prompted with a $start \rightarrow topic_a \rightarrow topic_b$ task which means that he is supposed to lead the conversation from any starting point to $topic_a$ and then to $topic_b$. Each topic represents an entity from a KG built based on comments and synopsis from movies. The authors propose two models to enable dialogue systems to converse: a retrieval-based model and a generation-based model. A retrieval-based model takes an input dialogue context and tries to find the best response by retrieving candidate entities from the DuConv database and then detecting the best one. This model consists of four modules: two encoders (context-response representation and knowledge representation), a knowledge reasoning module and a matching module. For the generation-based model, a vanilla seq2seq model is enhanced with an extra knowledge selection paradigm. The model takes as input the given dialogue context, the dialogue goal and related knowledge and encodes them using bi-directional GRUs. Then it is left to decide which knowledge would be appropriate next.

[3] propose a data-driven reasoning model that maps dialog transitions with KG paths. The base assumption is that given a seed entity, there exists a small subset of walkable paths within the KG involving some ideal entities to introduce as a response given a dialog context. The authors propose an attention-based graph decoder that walks an optimal path within a KG to prune unlikely candidate entities. The input is encoded using Bi-LSTMs with self-attention modules and an auto-regressive graph decoder takes the encoder output and generates walk paths. A zeroshot re-ranking model uses the candidate embeddings prediction results to rank entities based on their relevance and path scores. Another important contribution of this paper is the *OpenDialKG* corpus where each entity from the context dialog is manually appended with its corresponding KG path. The dataset consists of 15,000 chat conversations (91,000 turns) for two tasks: recommendation and chit-chat. The recommendations task uses entities related to movies (titles, actors, directors) and books (titles, authors), while the chit-chat uses entities related to sports (athletes, teams) and music (singers). To build the dataset and the parallel KG paths, a user starts a topic on a given entity, then the other agent is prompted with a set of candidate entities which are close in the knowledge graph to the seed entity. Our work is based on this *OpenDialKG* dataset which we find highly valuable due to its KG path annotations alongside the dialog context. The intuition is that a classifier paired with the power of a transformer language model can utilize this data to identify how it relates to the relevancy of a candidate entity.

3 RESEARCH GAP AND QUESTIONS

As seen in the previous section, an area which has not been extensively explored within the task of conversational dialogues is the inclusion of newer, state-of-the-art deep learning models for encoding and decoding the input. We focus on transformer models, more specifically BERT. The aim of our research is to answer the following main question:

How do transformers affect entity prediction performance in conversational Q&A knowledge grounded dialogue?

In order to answer this broad question, we investigate the following sub-questions:

(1) How many of the top entities selected by the classifier can be considered relevant? i.e. Is there a threshold on the classification score that can be used for entity selection?
(2) How can the entity classifier and generator be combined?
(3) Do transformers generate more relevant responses than the baseline?

(4) How can we rank the best responses given that the generator produces a response for each entity selected by the classifier? Do we need to train another model for response selection?

4 EXPERIMENTAL DESIGN

In this section we describe our knowledge graph grounded conversation generation pipeline. Figure 2 illustrates the overall architecture.

4.1 Candidate Entities

Given an input dialog context $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, the first step in the pipeline is to detect the seed entity $\mathbf{s} \in \mathbf{x}$, the subject of discussion. For simplicity, assume that there is only one seed entity per dialog context. There are around 100,000 unique entities in OpenDialKG therefore it is not computationally expensive to bruteforce the process and simply iterate through the entities from the network and check whether they are present in the input string. Given this seed entity $\mathbf{s}$, we can generate candidate entities that could naturally be part of the response following $\mathbf{x}$. To do this, we look in the 1-hop neighbourhood of $\mathbf{s}$, more precisely we look at its direct connections and randomly extract k candidates together with their respective relations $\{(r_1, o_1), ..., (r_k, o_k)\}$. Together with the dialog context, each of this candidate entity-relationship pairs is fed to the pretrained BERT classifier in order to predict their relevancy as a response to the input.

4.2 BERT Binary Classifier

In this section, we describe our transformer model. It focuses on learning appropriate relations and objects that represent good candidates for a conversational dialogue. Inspiration for this choice came from the results in *Voskarides et al.*[5], where a similar BERT encoder was used within a query resolution model, and from successful applications of BERT in understanding relations between entities such as in [7, 8].

4.2.1 Input sampling. We use the OpenDialKG dataset for generation of both positive and negative samples. Each conversation in the dataset is evaluated on a scale of 1 to 5 by both agents, measuring the coherence of the dialog. In order to increase the accuracy of our model with less data, we prune the OpenDialKG to only contain conversation where the user and assistant ratings are maximum, i.e. 5, leaving us with 4,628 conversations.

As stated before, OpenDialKG dialogues contain a sequence of utterances along with annotated knowledge paths that introduce new entities. Let us define positive samples as a dialog utterance from the pruned dataset together with the triple (s, r, o) corresponding to the KG path in the response. For a message $\mathbf{x}$, we only extract paths (s, r, o) such that $s \in \mathbf{x}$, more precisely such that the subject is directly referred in the previous message, leaving us with 7,827 such instances. The intuition behind this is that we want to constrain the model to reason as close as possible to the last utterance and focus only on one subject. Any triple in the KG network that is not linked to $\mathbf{x}$ in the pruned dataset is considered a possible negative sample for this message.

Having this data, we can build an input entry for each triplet that comes with a message: [$\mathbf{x}$ <SEP> r <SEP> o]. <SEP> is a special separator token used in BERT encoding which marks the end of one sentence. Table 2 shows an example of one positive sample and one negative sample. This type of input format forces the model to decide on relevant, appropriate matches between context and knowledge graph. The generation of negative samples is less straight-forward and in the following sections we present effects on the model output given different negative sampling strategies.

Input	Label
Do you like Iron Man <SEP> starred_actors <SEP> Robert Downey Jr.	1
Do you like Iron Man <SEP> country_of_origin <SEP> United States of America	0

Table 2: Examples of input sequences. The first entry can be considered a positive sample since it proposes a relevant entity. The second entry is a negative sample since the proposed entity would be irrelevant to the conversation.

4.2.2 BERT encoder. Since we are performing a classification task, we must pre-pend the BERT specific <CLS> token to each input. The <PAD> token is used for padding, as tokenization of input must result in a same length output. As BERT works with fixed-length sequences, we can choose the maximum length by investigating the distribution of token length of each input entry. Figure 3 shows that most of the entries have less than 60 tokens, thus we can safely set the maximum length of the input to 65. Next, we build the classification model on top of the basic BertModel.[1] More specifically, we add a dropout layer, a fully-connected layer and a softmax activation function for our two class output. A dropout probability of 0.2 is used to regularize the parameters.

4.2.3 Training data. We explore several approaches for constructing the train and test datasets. More precisely, we experiment with the ratio of positive to negative samples, as well as with the negative sampling method. By adjusting the sampling ratio and having more negative examples than positives, we expect the model to learn powerful features from the relevant entities such that it can improve its performance metric. Two negative sampling methods are explored: **random sampling** and **distribution-based sampling**.

The test dataset can contain both positive and negative instances, and it might hold that the classifier will be exposed to unseen (s, r, o) triplets which do not exist in the KG. Since the candidate entities are extracted from the KG network, this scenario will not happen in practice, but it allows us to discover better insights into the limitations of the model. We expect the results for the exclusively positive test dataset to be higher, as the false positives are not bound to appear.

Unbalanced data. Random negative samples.
For every positive example [$\mathbf{x}$, s_1, r_1, o_1], we sample by default 5 negative examples [$\mathbf{x}$, s_2, r_2, o_2] such that $o_1 \neq o_2$. The explanation behind this is that if we loosened the constraint and $o_1 = o_2$, the model would simply use the relationship types as discriminators for the relevancy of the candidate entities and this is undesirable. The candidate entities are randomly extracted from the 1-hop neighbourhood of the seed entity in the OpenDialKG network.

[1] https://huggingface.co/transformers/model_doc/bert.html#bertmodel

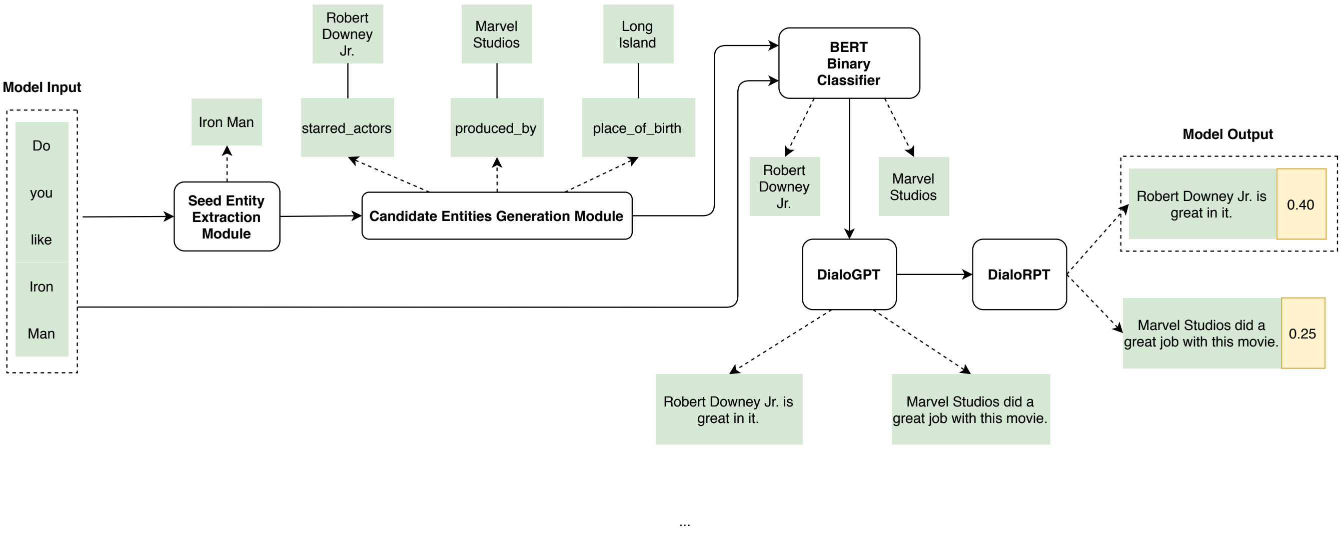
Model Input
Do
you
like
Iron
Man
Seed Entity Extraction Module
Iron Man
Candidate Entities Generation Module
starred_actors
Robert Downey Jr.
produced_by
Marvel Studios
place_of_birth
Long Island
BERT Binary Classifier
Robert Downey Jr.
Marvel Studios
DialoGPT
DialoRPT
Robert Downey Jr. is great in it.
Marvel Studios did a great job with this movie.
Model Output
Robert Downey Jr. is great in it.
0.40
Marvel Studios did a great job with this movie.
0.25
...

Figure 2: Overall architecture. An input sequence is fed to the system which consists of five modules. The output should be a coherent and relevant response to the input dialog.

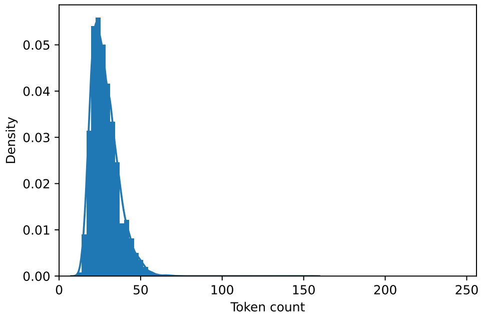

Figure 3: Input token distribution

Balanced data. Distribution-based negative samples. In this next approach, we aim to generate negative samples such that the dataset is balanced, having a 1:1 positive-negative ratio. Therefore, for each message **x** among the positive samples, we inspect the seed entity s and append the relationship predicate r together with an irrelevant entity o according to the following strategy:

- r - predicates have a global distribution as well as a local distribution given the entity they are connected to. We want to append a globally frequent, but locally infrequent predicate to each entity in order to emphasize the difference between appropriate and inappropriate entity-relation combinations.
- o - append the most frequent entities which are connected to the chosen subject. The object was initially selected based on the distribution according to the selected predicate. However, this leads to learning true entries instead of relevant entries. This happens because objects that are related to predicates may not be at all related to the subject in the knowledge graph. We want the model to make relevance distinction between two objects which are close to the subject in the knowledge graph. This is not guaranteed when selecting objects only based on the predicate frequency. Thus, both the subject and the predicate should be incorporated in choosing the object. The problem is that distributions based on subject-predicate are scarce for most of the pairs, therefore we chose the object based on the subject. This ensures that the picked entity is close to the subject in the knowledge graph.

The training, validation and testing set are split according to a 80-10-10 ratio. A prediction threshold needs to be determined, establishing the point at which the model becomes unreliable. For this we inspect the amount of mistakes that the model makes at several prediction scores and hence, we can ensure that the entries that are fed to the generator are relevant enough.

4.3 DialoGPT

Once a relevant entity is determined, a response needs to be generated such that:

i It is well-articulated;
ii It is relevant to the previous dialog utterance;
iii It contains the relevant entity.

DialoGPT has the potential to generate such outputs. It is a large-scale dialog response generation model pretrained on 147M multi-turn conversations from Reddit discussion threads. Human evaluation indicates that the quality of the text is comparable to human response quality considering only one-turn conversations. Just as seen before in Table 1, if we were to look at the bot's replies individually, they are coherent and in line with the previous utterance, but they are not engaging.

In this section, we are going to cover methods to impute a relevant entity in the response of DialoGPT using the implementation from Huggingface Transformers. [2]

4.3.1 Entity Imputation at Training Time. Given an input dialog context $\mathbf{x}$ and a relevant entity o, the goal is to generate a response $\mathbf{y} = \{y_1, ..., y_m\}$ such that $o \in \mathbf{y}$.
Intuitively, we can follow the same reasoning as for the classifier and train DialoGPT using the input context and the object. The tokenizer from DialoGPT does not implement a separator token such as `[SEP]` and thus we are required to artificially separate the entity from the text as in Table 3. More precisely, the model is re-trained using input sequences $\{\mathbf{x} <o>\}$ extracted from the pruned OpenDialKG dataset.

Context	Do you like Iron Man <Robert Downey Jr.>
Response	Yes, Robert Downey Jr. is great in it.

Table 3: Examples of input sequences for entity imputation at training time. The candidate entity is separate in the input from the dialog context by <>.

4.3.2 Entity Imputation during Inference. The entity can be injected forcefully in the beginning of the response at inference time. This implies that given the same goal as defined before, DialoGPT generates a response $\mathbf{y} = \{o, y_1, ..., y_m\}$. We trivially inject the entity after the `[EOS]` end-of-sentence token of the transformer, forcing the generator to continue the response considering this entity.

We also consider executing a round of fine-tuning on some conversations from OpenDialKG to expose the generator to more engaging dialog utterances.

4.4 DialoRPT

Recently, DialoGPT has been improved by integrating it with a set of large-scale dialog ranking models. DialoRPT[2] has been trained on +100 millions of human feedback data and it is used to re-rank the generated response candidates in accordance with some task. Given a context and responses generated by DialoGPT, it can predict whichever gets more upvotes, more direct replies, or a longer follow-up thread. These are performance measures for Reddit data, but we can adapt them to our task by making assumptions such as: a post with a long follow-up thread means that it has started a debate and it can be thus considered engaging. There are two more tasks which given one single response, DialoRPT compares it to a random human response or a machine generated response. These can be used to distinguish between the traditional DialoGPT and the fine-tuned one.

Context	Response	Score
Do you like Iron Man	Robert Downey Jr. is great in it	0.40
Do you like Iron Man	It is my favourite movie	0.14

Table 4: The re-ranking of two responses. The model predicts that the first response, introducing a new entity, will start a longer discussion thread.

We will experiment with the three ranker types: depth (long discussion threads), width (number of direct replies), and up-down (number of likes). After evaluation, we can decide which one is most suitable to use in order to compliment the pipeline.

4.5 Overall system

To test the performance entire system, we choose as baseline the KG-Copy Network[1] and the OpenDialKG walker[3] and we are focusing on the entity prediction task.

Data. The measurements for the baseline models are taken as given in the respective papers. To build the dataset for these experiments, we pick three seed entities who have between 40-50 neighbours in the Knowledge Graphs and for each of them, we build an artificial dialog context:

- *Kanye West* is a great songwriter.
- *Game of Thrones* is my favourite series.
- *Shrek Forever After* is such a fun movie.

We manually annotate each dialogue context's candidate entities as being relevant and irrelevant and we compare these ground truth labels to the predictions of the system.

Entity-F1. Chaudhuri et al. (2019) do not provide a description of this Entity-F1 metric. We compute the usual F1 score of the entities predicted as relevant:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}.$$

Recall@k. Moon et al. (2019) do not provide further detail on how $recall@k$ was applied. We assume a constant number of relevant entities for each k:

$$recall@k = \frac{\textit{number of relevant entities in the top-k positions}}{\textit{total number of relevant entities}}.$$

5 RESULTS AND ANALYSIS

Source code, datasets and experiments described below are made publicly available.[3] Pre-trained BERT classification model[4] and fine-tuned DialoGPT[5] are stored separately due to size constraints.

5.1 Classifier Performance

Unbalanced data. Random negative-samples.
Training the model on the unbalanced dataset results in over-fitting. For the test set, the model does not perform above the guessing chance of 80%, always predicting the candidate entities to be irrelevant to the input dialog context. The first line of Table 5 showcases the accuracy of the model for this case.

Balanced data. Random negative samples.
The second entry of Table 5 shows that the model goes above the guessing chance of 50%, but it is still under-performing and there is room for improvement for this method as well. We notice that the test set has a higher accuracy than the train set. This can be caused

[2] https://huggingface.co/microsoft/DialoGPT-large

[3] https://github.com/epureanudiana/InformationRetrieval2

[4] https://drive.google.com/file/d/1u16QFuds54LQ6fruVbo0MYwxOzd2ZYFG/view

[5] https://drive.google.com/drive/folders/1-Gpy6PEkW9xslpeEaBgl9a3lVTFaZh65

by the dropout layer, as during the training 20% of the features are set to 0 while during testing all features are used for prediction. The issue with the approach of sampling negative instances randomly is that not all relations and candidate entities which are not present in the training set are equally unsuitable candidates for a conversational response.

Balanced data. Distribution-based negative samples.
This approach returns the best results as the model is able to learn what makes for a good candidate relation and object. Again, we see the same situation as above regarding the higher test accuracy. Now, for the test dataset containing exclusively positive samples, we obtain an accuracy of 95,9%, which confirms our expectations.

Training with positive-negative pairs.
For the previous experimental setups, the dataset is split into training, validation and test by shuffling all the instances. Therefore, there is no guarantee that during training we feed both a positive and a negative instance given a subject entity. It becomes problematic when a subject appears only with relevant predicates and objects, as it does not provide information regarding what can be irrelevant. The same thing applies for subjects that exclusively appear as negative instances. The model may be biased with respect to an entity. We thus investigate the effect of positive-negative pairs in the accuracy. The training dataset is adjusted such that for each positive example $[\mathbf{x}, s_1, r_1, o_1]$, its corresponding negative example $[\mathbf{x}, s_1, r_2, o_2]$ is also present. For evaluating the model, only positive examples are used.
As it can be seen in the last two rows of Table 5, the accuracy slightly decreases to 94,5% when we also used the paired examples for training. However, the training accuracy tells us that there is no significant difference between the content of training data generated from random shuffling of instances and training data generated from paired instances. On the test set, the model is less confident in predicting relevant entries. Even if the accuracy is unchanged for the training data, it may be that the model better learns the difference between relevant and irrelevant entries. This, in turn, results in a better data generalization and hence, in a lower test score.

	Train	**Test**
Unbalanced data. Random neg. samples.	97%	77%
Balanced data. Random neg. samples.	70%	72%
Balanced data. Distr-based neg. samples.	89%	94%
Balanced data. Distr-based neg. samples. Only positive test data.	**89%**	**95.9%**
Balanced data. Distr-based neg. samples. Only positive test data & paired train data.	89%	94.5%

Table 5: BERT binary classifier results

Comparison to related work
We can compare the classifier with the techniques used in the aforementioned related work. [1] uses cosine similarity in order to find the best match between a query and an entity. An average embedding of each noun and verb phrase embedding from the query is used for computing the embedding of the query. Then the embedding of an entity is computed by averaging the embeddings of the local KG's subject entity and relation labels for each triple. Thus, similarly to our BERT classifier, a ranking of the best entities is defined, but using a different relevance function.
[3] ranks predicted entities based on a zeroshot relevance score. The subject entities are extracted from the current utterance and all potential knowledge graph paths starting from the entities are generated by an auto-regressive graph decoder. In contrast, the BERT classifier uses these annotated walk paths as positive inputs.

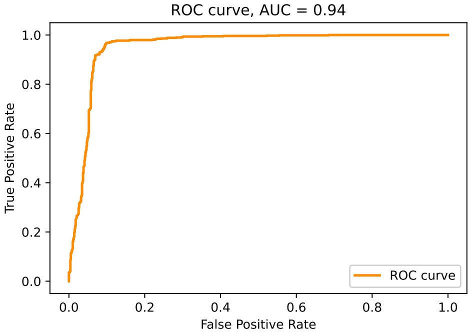

Figure 4: BERT binary classifier ROC curve

Limitations of BERT classifier
Consider the model using balanced data and the distribution-based negative sampling. We can now inspect the cases where the model under-performs and understand the reason behind this. Firstly, we measure the class separability of the model by plotting the ROC curve. As Figure 4 shows, the AUC is close to 1, therefore we have a model with a high class separation capacity.

Looking into the false positive and false negative cases, we can correct for the remaining percentage and ensure an appropriate and filtered input for the generator model. Analyzing the confusion matrix in Figure 5, we can see that more than 90% of the misclassifications are false positives, which is undesirable as it implies that the model predicts irelevant entities as being suitable to include in the response. These cases are analyzed below.

The false positives appear equally often with the following relations: *has_genre*, *directed_by*, *written_by*. This is expected, as these are the most frequent predicates, but we can inspect these cases further.

- *has_genre*. The model mostly fails in cases where the person is asking questions such as
  - Do you know anything about X?
  - What do you know about X?
  - Do you have any information about X?
  - Can you tell me more about X?

  where X is a person. This can be because the training data is skewed towards having *has_genre* next to these kind of questions but related to books and movies, not actual people. The introduced objects vary too much to be the cause of misclassification and thus the model predict the relevancy of the candidates solely based on the relationship.

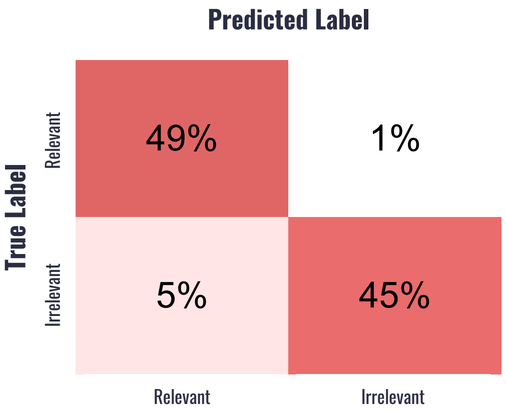

Figure 5: Confusion matrix to visualize the performance of the classifier, showcasing from left to right the distribution of true positives, false negatives, false positives and true negatives.

- *directed_by*. In this case errors appear for utterances about movies.
  - *What are your recommendations for movies similar to Divergent <SEP> directed_by <SEP> Young-adult fiction.*
  - *My favorite is World War Z. <SEP> directed_by <SEP> Zombie (Media genre).*

  When talking about movies, the conversation could naturally go towards the director of the movie or even its genre. The association between this relationship and objects different than a director, like in our example the genre, is highly improbable and thus the model should predict the entities as irrelevant. This does not happen as instead it uses either the relationship or the object as discriminator, but not their association which does not make sense.
- *written_by*. In this case, we deal with questions about movie recommendations, where the model tries to introduce this predicate followed by an object describing a person. The connection between relation and object makes sense. Below are some examples.
  - *The Imitation Game is a war movie. <SEP> written_by <SEP> Benedict Cumberbatch*
  - *I love Anthony Hopkins. Can you recommend some more films with him? <SEP> written_by <SEP> Peter and Paul*
  - *Could you recommend a movie similar to Kingdom of Heaven? <SEP> written_by <SEP> Ridley Scott*
  - *Could you recommend movies like Kill Bill: Vol. 1? <SEP> written_by <SEP> Quentin Tarantino*

  The first two examples showcase a behavior similar to the previous case, where the model would predict a candidate entity as being relevant even though the association between the relation and the object is not correct. The last two examples are correctly labeled as irrelevant since the dialog context concerns a recommendation for a movie, but factually speaking, these KG paths could have led to a correct relevant entity. This demonstrates an additional weakness of the classifier, namely that it is limited in scope by using solely 1-hop neighbours. Usually two movie entities will not be directly connected by a similarity relation, therefore to solve the recommendation tasks it is necessary to look in the distant neighbourhood.

To better generalize the type of errors present in the predictions, we classify them in three categories:

- **object-based**. The input *I think Enrique Iglesias is one of the most important singers from Spain. <SEP> directed_by <SEP> Takin' Back My Love* is such an example. The model is introducing a relevant object given the entity, however, irrelevant given the relation. The relation is also irrelevant to the subject. Thus, starting from a relevant object, the model fails at finding the relevant relation.
- **relation-based**. As an example we have *Do you know about the movie Les Misérables? <SEP> directed_by <SEP> Play.* In this case, the relation does make sense given the subject, but it does not given the object. Therefore, starting from a relevant relation, the model fails at finding the relevant object.
- **relation/object-based** Such an example is *Do you know The Dictator? <SEP> has_genre <SEP> Sacha Baron Cohen.* Here we can see relevance between the subject and the relation, between the relation and the object, as well as between the subject and the object, however, the triple does not make sense altogether.

Having inspected the weaknesses of the classifier, we can add further avoid them when feeding the generator. However, another issue is given by the fact that the classifier is overconfident in the prediction scores. If an instance is predicted to be relevant, its relevance score is higher than 0.9. This makes it hard to set a threshold value on the prediction score. Given this situation, we use a separate ranking model for a better stratification of the relevant entities and generated responses.

5.2 DialoGPT

The entity imputation at training time did not prove to be successful. The model was not able to learn to include in the response the entity given in the input alongside the dialog context. A reason for this might be the sparse training set extracted from OpenDialKG. For this reason, our results focus solely on the model using entity imputation at inference time.

5.3 DialoRPT

In order to choose a suitable ranker to complete the pipeline, different ranking settings (*updown*, *width*, *depth*) and models (GPT-2 with and without finetuning on the OpenDialKG dataset) were experimented with. Results of these experiments are shown in Table 7 (no finetuning on OpenDialKG conversations) and Table (with finetuning of DialoGPT), where percenteges of times DialoRPT ranks DialoGPT's answers with entity injection at inference higher than the corresponding candidate model answers.

Ranker setting	**DialoGPT**	**KG-Copy**
updown	80%	75%
depth	61%	100%
width	60%	75%

Table 6: Base - no finetune. Percentage of answers given by DialoGPT + entity injection in answers ranked higher than other models by DialoRPT.

Ranker setting	**DialoGPT**	**KG-Copy**
updown	94%	68%
depth	88%	68%
width	88%	68%

Table 7: Finetune. Percentage of answers given by DialoGPT + entity injection in answers ranked higher than other models by finetuned DialoRPT.

	Entity-F1
KG-Copy Network	23.58
Base model	58.75
Finetuned model	63.27

Table 8: Model performance against the baseline using the F1 metric. Our proposed models are compared against the original KG-Copy Network results.

	R@1	**R@3**	**R@5**	**R@10**	**R@25**
OpenDialKG	11.3	23.3	31.0	44.0	60.5
Base model	5.6	8.6	12.6	20.1	58.0
Finetuned model	3.1	14.3	21.6	35.0	63.9

Table 9: Model performance against the baseline using the recall@k metric. Our proposed models are compared against the original OpenDialKG results.

It is clear that DialoGPT prefers answers generated by DialoGPT with entity injection at inference. In order to decide which ranker setting to choose, we conduct qualitative analysis and we select the one that yields the most engaging reply.

One thing that can be noticed is that the ranker that predicts which answer would get more replies (width), and the one that predicts which would lead to a longer discussion (depth), both have similar preferences among candidate answers, and differ from the updown ranker, which predicts how many upvotes a comment would receive:

Having context *The Fog sounds cool, what else can you tell me about it?* and candidate answers *Todd Garner wrote it, right?, Tom Welling is in it right?, Supernatural (TV Genre)*, the rankers perform as follows:

- updown:
  - *Todd Garner wrote it, right?* score: 0.28
  - *Tom Welling is in it right?* score: 0.29
  - *Supernatural (TV Genre)* score: 0.46
- depth:
  - *Todd Garner wrote it, right?* score: 0.38
  - *Tom Welling is in it right?* score: 0.67
  - *Supernatural (TV Genre)* score: 0.27
- width:
  - *Todd Garner wrote it, right?* score: 0.41
  - *Tom Welling is in it right?* score: 0.64
  - *Supernatural (TV Genre)* score: 0.31

Based on the ranker's behavior on the OpenDialKG dataset there is no clear indicator of ranker setting is more suitable for the proposed pipeline. We therefore select a ranker setting that makes more theoretical sens, being it either the *width* or *depth*, with seemingly little impact of the exact choice based on our results. This choice is made as answers that are most engaging are preferred (which is an indirect target of both settings mentioned).

5.4 Quantitative Analysis of the System

Table 8 and Table 9 showcase the performance of our proposed model versus the baseline. Since the results for KG-Copy Network and OpenDialKG are not re-computed on the same dataset, we can not draw a fair comparison between these settings. Between our base model and the one using a finetuned version of DialoGPT, it seems that the latter performs better. The finetuned version return a higher F1 score and a higher recall for almost all *k*s, except Recall@1, where the difference is small.

6 CONCLUSIONS

We study how transformers affect entity prediction performance in conversational Q&A knowledge grounded dialogues. In order to do this, we propose a classification model based on BERT embeddings which learns the relationships between the conversation context and entities. We have seen that the classifier is overconfident in its predictions. This causes problems in setting a threshold for selecting the best subset of entities given their score. These entities are then fed to a state-of-the-art transformer model using entity imputation during inference. We show that other methods of combining the classifier and generator do not obtain satisfactory results. The lack of common evaluation metrics for both baseline models makes the analysis of our model slightly difficult. However, individual comparison between our model with each baseline score function shows that we can do better than the baselines or outperform them if starting from a specific threshold. We have also seen that indeed we need a ranking model for selecting the best response among all the generated responses. However, this part of the architecture can potentially be removed if we manage to define an accurate threshold for the classification model prediction scores. Thus, we answered all sub-questions around our point of interest and showed the potential of transformers in knowledge graph grounded conversation generation.

REFERENCES

[1] Debanjan Chaudhuri et al. "Using a KG-Copy Network for Non-goal Oriented Dialogues". In: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I.* 2019, pp. 93–109. DOI: 10.1007/978-3-030-30793-6_6. URL: https://doi.org/10.1007/978-3-030-30793-6%5C_6.

[2] Xiang Gao et al. "Dialogue Response RankingTraining with Large-Scale Human Feedback Data". In: *EMNLP.* 2020.

[3] Seungwhan Moon et al. “OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs”. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. 2019, pp. 845–854. doi: 10.18653/v1/p19-1081. url: https://doi.org/10.18653/v1/p19-1081.

[4] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: (2019).

[5] Nikos Voskarides et al. “Query Resolution for Conversational Search with Limited Supervision”. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. 2020, pp. 921–930. doi: 10.1145/3397271.3401130. url: https://doi.org/10.1145/3397271.3401130.

[6] Wenquan Wu et al. “Proactive Human-Machine Conversation with Explicit Conversation Goals”. In: *CoRR* abs/1906.05572 (2019). arXiv: 1906.05572. url: http://arxiv.org/abs/1906.05572.

[7] Kui Xue et al. “Fine-tuning BERT for Joint Entity and Relation Extraction in Chinese Medical Text”. In: *2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019, San Diego, CA, USA, November 18-21, 2019*. 2019, pp. 892–897. doi: 10.1109/BIBM47256.2019.8983370. url: https://doi.org/10.1109/BIBM47256.2019.8983370.

[8] Liang Yao, Chengsheng Mao, and Yuan Luo. “KG-BERT: BERT for Knowledge Graph Completion”. In: *CoRR* abs/1909.03193 (2019). arXiv: 1909.03193. url: http://arxiv.org/abs/1909.03193.

[9] Yizhe Zhang et al. “DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation”. In: *ACL, system demonstration*. 2020.