Retrieval-Augmented Neural Response Generation Using Logical Reasoning and Relevance Scoring

Nicholas Thomas Walker
Norwegian Computing Center
Oslo, Norway
walker@nr.no

Stefan Ultes
University of Bamberg
Bamberg, Germany
stefan.ultes@uni-bamberg.de

Pierre Lison
Norwegian Computing Center
Oslo, Norway
plison@nr.no

Abstract

Constructing responses in task-oriented dialogue systems typically relies on information sources such the current dialogue state or external databases. This paper presents a novel approach to knowledge-grounded response generation that combines retrieval-augmented language models with logical reasoning. The approach revolves around a knowledge graph representing the current dialogue state and background information, and proceeds in three steps. The knowledge graph is first enriched with logically derived facts inferred using probabilistic logical programming. A neural model is then employed at each turn to score the conversational relevance of each node and edge of this extended graph. Finally, the elements with highest relevance scores are converted to a natural language form, and are integrated into the prompt for the neural conversational model employed to generate the system response.

We investigate the benefits of the proposed approach on two datasets (KVRET and GraphWOZ) along with a human evaluation. Experimental results show that the combination of (probabilistic) logical reasoning with conversational relevance scoring does increase both the factuality and fluency of the responses.

1 Introduction

Although Large Language Models (LLMs) are widely used for conversational response generation, they still suffer from a number of shortcomings, including their propensity to produce hallucinated content (Ji et al., 2023). Recent work has demonstrated how to exploit external information sources such as knowledge bases (KBs) to improve the output of LLMs in various downstream tasks (Yu et al., 2022a), including dialogue systems (Wang et al., 2021). A promising approach is Retrieval-Augmented Generation (RAG), which operates by first retrieving relevant information from external sources and then augmenting the input provided to the LLM with this retrieved content (Lewis et al., 2020). While RAG has been demonstrated to reduce hallucinations (Shuster et al., 2021), LLMs are nonetheless easily distracted by irrelevant information (Shi et al., 2023). For this reason, one should strike a balance between providing the model with potentially useful information and avoiding overloading it with too many spurious or irrelevant facts.

Moreover, while LLMs have recently shown some success at reasoning benchmarks (Bubeck et al., 2023), their ability to engage in multi-step reasoning remains poor. In particular, Dziri et al. (2023) provide a systematic investigation of the performance of LLMs on several compositional reasoning tasks, and find that those models largely rely on pattern matching shortcuts and fall short of exhibiting generic problem-solving skills.

This paper presents a novel approach to retrieval augmented generation in task-oriented dialogue systems that seeks to address those challenges. Following (Walker et al., 2022), we represent the background knowledge of the system as a *graph* of dynamically updated facts representing the dialogue state. This initial graph is first enriched at each turn with derived facts inferred through probabilistic logical programming using a limited number of rules, using ProbLog as a framework (Fierens et al., 2015). The conversational relevance of each fact is then scored using a neural model, based on various features expressing both the conversational saliency of each entity and semantic similarity between the fact and the recent dialogue history. The most relevant facts are then converted into sentences and incorporated into the input of the response generation model. Crucially, the relevance scoring model and the response generation model are optimised jointly based on dialogue examples. Figure 1 provides a general sketch of the approach.

The paper makes the following contributions:

1. The use of probabilistic logical programming

Proceedings of the 27th Workshop on the Semantics and Pragmatics of Dialogue, August, 16–17, 2023, Maribor.

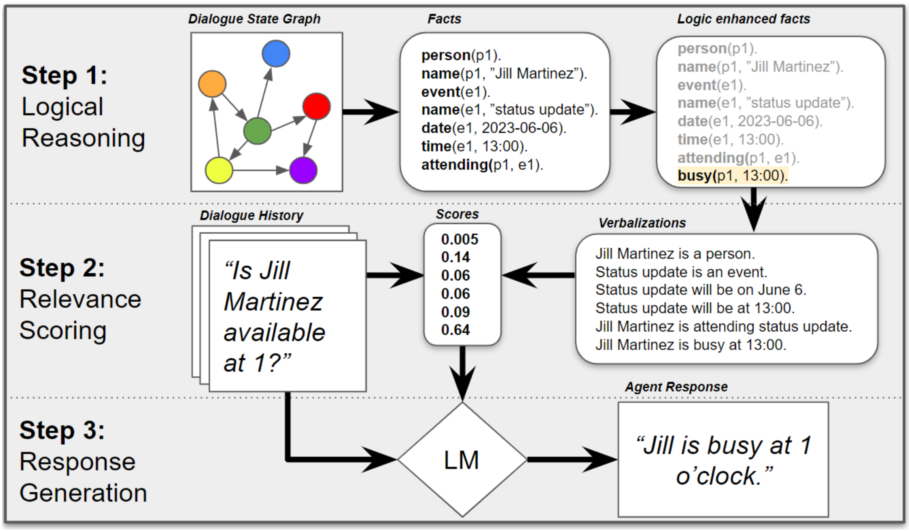
Step 1: Logical Reasoning
Dialogue State Graph
Facts
person(p1).
name(p1, "Jill Martinez").
event(e1).
name(e1, "status update").
date(e1, 2023-06-06).
time(e1, 13:00).
attending(p1, e1).
Logic enhanced facts
person(p1).
name(p1, "Jill Martinez").
event(e1).
name(e1, "status update").
date(e1, 2023-06-06).
time(e1, 13:00).
attending(p1, e1).
busy(p1, 13:00).
Step 2: Relevance Scoring
Dialogue History
"Is Jill Martinez available at 1?"
Scores
0.005
0.14
0.06
0.06
0.09
0.64
Verbalizations
Jill Martinez is a person.
Status update is an event.
Status update will be on June 6.
Status update will be at 13:00.
Jill Martinez is attending status update.
Jill Martinez is busy at 13:00.
Step 3: Response Generation
LM
Agent Response
"Jill is busy at 1 o'clock."

Figure 1: General sketch of the proposed approach. The starting point is a dialogue state represented as a knowledge graph that combines both background information and various features extracted from the dialogue turns (transcriptions, speakers, entity mentions). In Step 1, the facts representing the nodes and edges of the graph are first extended with derived facts using probabilistic logical programming. Those facts are then verbalized (converted into natural language sentences) in Step 2, and a neural model is employed to score their conversational relevance with regard to the current dialogue context. Finally, the k most relevant facts are included in Step 3 as part of the prompt for the neural language model responsible for producing the actual system response.

   to extend the knowledge graph representing the current dialogue state with derived facts.

2. A neural scoring model that relies on both dialogue-level features (such as recency) and semantic similarity to determine the most relevant nodes and edges of this augmented graph.

3. An empirical evaluation of the above approach using two dialogue datasets (KVRET and GraphWOZ) and a human evaluation.

2 Related Work

Several papers have investigated the use of neural models to retrieve relevant information from knowledge sources and integrate their results in response generation. Dinan et al. (2018) distinguish between the tasks of knowledge selection and response generation for knowledge-grounded dialogue agents. This information may be structured or unstructured (Young et al., 2018; Zhao et al., 2020), and generally consists of documents describing entities which may be relevant to the dialogue. For an open-domain dialogue model, this background information can be drawn from sources such as Wikipedia. In task-oriented dialogue, relevant information will depend more heavily on the domain of the dialogues. Previous work has also demonstrated the effectiveness of jointly learning a language model with a knowledge retrieval model (Zhang et al., 2021), simplifying the task of identifying relevant items without labelled data.

External information, often in the form of KBs, is crucial to many dialogue models (Ghazvininejad et al., 2018; Parthasarathi and Pineau, 2018; Zhang et al., 2018; Madotto et al., 2018). Multiple approaches exist for combining retrieval and generation to yield higher quality responses, such as by first generating a response and subsequently refining it (Weston et al., 2018). The model used by Peng et al. (2023) queries an LLM and evaluates the output for factuality, and re-queries the system with feedback to elicit a more factual response.

Thulke et al. (2021) propose an approach which samples a subset of the background knowledge rather than optimizing over the entirety of it, a process that we also integrate in our model training process. The Global-to-Local Knowledge Selection model is an alternative which pre-selects information across the whole of the background knowledge using topic transition vectors (Ren et al., 2020). Meanwhile, He et al. (2021) proposed a model which integrated information about system API calls to the retrieval model.

Numerous models make use of large, static, knowledge bases to augment language models. The KETOD model (Chen et al., 2022) used Wikipedia data to enhance a task-oriented dialogue system's responses with information about entities in the dialogue. Likewise, Kim et al. (2020) and Zhan et al. (2021) modelled knowledge selection with a latent variable model, which have also shown strong results for RAG in a zero-shot setting (Li et al., 2020). Moon et al. (2019) investigated a graph decoder model using random walks over a knowledge graph containing dialogue relevant information. Paranjape et al. (2021) made use of a "guide" retriever model to use posterior information from responses to help the retriever model learn from relevance in both the input and reference output responses. Cai et al. (2019) made of of a "skeleton-guided" response generator in a dialogue system.

RAG has also been used to improve common sense reasoning (Yu et al., 2022b), or incorporate graphs of commonsense knowledge to the model (Zhang et al., 2019a). Common sense knowledge in the SenticNet KB has also been used as a source of knowledge for a dialogue model (Young et al., 2018), albeit without logical reasoning over the graph. Liu et al. (2019b) explored a model which used multi-hop reasoning to identify a relevant vertex in a graph of "factoids" which are each associated with unstructured sentences. Other models have made use of linguistic rule-based components to combine semantic representations of the dialogue state with background knowledge to improve empathetic responses and dialogue flow in task-oriented dialogue (Smith et al., 2011).

A hierarchical approach to knowledge grounded task-oriented dialogue was presented by Lee and Jeong (2023), where the pipeline is composed of domain identification, entity extraction, and a pretrained language model to rank relevant documents. Other work proposed a novel factuality-specific sampling algorithm to improve LLM output (Lee et al., 2022), while Bonetta et al. (2021) used k-nearest neighbors to find relevant information.

3 Approach

As illustrated in Figure 1, the approach proceeds in three steps. Probabilistic logical programming is first employed to extend the initial knowledge graph with new facts based on a small set of rules. A neural scoring model then determines the relevance of those facts in the current dialogue context. The most relevant facts are then included as part of the input to the second neural model, which is responsible for the actual response generation. The next sections describe those steps.

3.1 Dialogue state representation

Following (Walker et al., 2022), we represent the current dialogue state (along with other background information that might be relevant for response generation) as a *knowledge graph* consisting of multiple entities connected by relations. The graph is always grounded in a specific dialogue and continuously evolves during the interaction, with new nodes and edges representing dialogue turns, speakers, or entity mentions. The dynamic and dialogue-specific nature of this knowledge graph stands in contrast with the static KBs (based on e.g. Wikipedia or similar sources) typically used in knowledge-grounded generation.

To account for uncertainties associated with noisy or partial observations (such as ASR transcriptions of user utterances or ambiguous referential links), both node attributes and labelled edges may be associated with probabilities.

3.2 Probabilistic logical programming

To explicitly reason over this graph, we rely on the probabilistic logical programming language ProbLog (Kimmig et al., 2011; Fierens et al., 2015). We assign each node to a unique identifier and represent the node attributes and edges as (ground) logical predicates, as illustrated in Figure 1. Node attributes and edges associated with a probability < 1 are expressed as probabilistic facts.

3.2.1 ProbLog

A ProbLog program consists of two parts: a set of ground probabilistic facts, and a logic program, expressed as a set of logical clauses. The clauses may be themselves associated with probabilities. ProbLog also allows for the definition of "annotated disjunctions" where mutually exclusive facts are coupled with a discrete probability distribution. Syntax-wise, ProbLog is a probabilistic extension of Prolog and supports both probabilistic and inductive reasoning. Given a set of logical rules and ground facts, ProbLog provides inference algorithms to efficiently query the probability of one or more predicates. This inference is done by converting the facts and logical program to a compact encoding such as Sentential Decision Diagrams (Vlasselaer et al., 2014) and then running weighted model counting (Chavira and Darwiche, 2008) on this compiled representation.

3.2.2 Entity linking rules

An important task in goal-oriented dialogues is to connect entity mentions to the actual entities present in the KB. For instance, if "Jill Martinez" is mentioned by the user, this mention must be linked to the actual node (p_1) for that person in the KB. Entity mentions may correspond to named entities, but may also take the form of pronouns (”she”) or generic noun phrases ("the meeting").

We first detect entity mentions in user utterances using a neural sequence labelling model fine-tuned on labelled, in-domain data from a pretrained ROBERTA model (Liu et al., 2019a). A small set of probabilistic ProbLog rules is then employed to determine the most likely reference among the entities in the knowledge graph. Those rules take advantage of both edit distance metrics and recency measures (Walker et al., 2022). Each rule is attached to a probability reflecting its strength. Those probabilities are estimated empirically from partial interpretations on the training data, following the approach described in Gutmann et al. (2011). After applying those entity linking rules, the outcome is then written back to the knowledge graph as probabilistic `refers_to` edges linking each observed mention to the entity it refers to.

3.2.3 Commonsense rules

Consider a scenario where a task-oriented dialogue system must answer a user question:

> *"What events do I have today?"*

Assuming the knowledge graph contains basic information about calendar events such as their date, time and attendees, answering this question rests upon multiple reasoning steps. As multi-step reasoning remains a challenging task for language models (Liu et al., 2023), we specify a small number of commonsense reasoning rules to automatically derive new facts from the current dialogue state. For the above example, the connection between dates and events in context can be made explicit with the following rule:

```
person(P), event(E),
attendee(E,P), date(E,D),
date(today,D)
⟹ attending_today(E,P)
```

A second example of a logical rule is as follows:

```
room(R), ¬(event(E),
location(E,R), date(E,D),
date(today,D), start_time(E,ST),
end_time(E,ET),
time_between(T,ST,ET,1))
⟹  room_available_today(R,T)
```

The above rule simply states that a room R is available today at a given time T if no event is scheduled at that time in that room.

The goal of those commonsense rules is to deduce facts that may provide useful information to the response generation model. Those logically derived facts will typically correspond to information that may be queried by the users, such as a person's agenda for today or the availability of a room at a given time. To avoid deriving too many spurious or irrelevant facts, we only query ProbLog for facts pertaining to entities recently mentioned in the dialogue history. For our experiments, we query entities mentioned in the current turn.

After applying both entity linking and commonsense rules, the facts are converted to a natural language *verbalization*. Each predicate is associated with a handcrafted template which creates a natural language form of the fact. For example, a person defined by the fact `person(p_123)` with a name `name(p_123, "Lisa Wilson")` can be verbalized as *Lisa Wilson is a person.*

3.3 Relevance Scoring

The second component of the proposed approach is a neural model that scores the relevance of the verbalized facts (including both the intial ones as well as the ones derived through logical reasoning). Given a dialogue history $x = [u_1, ...u_n]$ corresponding to a list of utterances and a set of verbalized facts Z, the relevance scoring model expresses the probability $P(z|x)$ that the fact $z \in Z$ is relevant for responding to x.

The model is expressed as a simple feedforward neural network based on the following inputs:

1. Semantic similarity measures between z and x, using the cosine similarity between the embedding of the verbalized fact z and the embedding of the most recent k utterances in the dialogue history x (concatenated if $k > 1$):

$$sim(z, x) = \frac{Enc(z) \cdot Enc(x_{[n-k:n]})}{\|Enc(z)\| \; \|Enc(x_{[n-k:n]})\|}$$

   The *Enc* embeddings are obtained with a sentence-BERT model (Reimers and Gurevych, 2019) optimized for semantic search and question answering[1]. The cosine similarity is computed for $k = 1$ and 2.

---

[1] https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1

2. BM25 information retrieval scores (Robertson et al., 2009) using the verbalized facts as database and the user utterance as a query.

3. Recency score expressing whether the fact z pertains to a recently mentioned entity. This score relies on the `refers_to` predicates derived from entity linking and captures the conversational saliency of entities and facts related to them. For instance, facts related to p_1 in Fig. 1 are salient since the person is mentioned in the last utterance.

The relevance model $P(z|x)$ is trained jointly with the response generation. Concretely, we define the probability of a response y given a dialogue history x as:

$$P(y|x) = \sum_{z \in Z} P(y|x, z)P(z|x) \tag{1}$$

where $P(y|x, z)$ is provided by the response generation model (see below), and express the probability of a response y given a prompt concatenating the dialogue history x and fact z, and $P(z|x)$ express the relevance of z for x. The relevance model $P(z|x)$ is then optimized by back-propagating the cross-entropy loss of Eq. (1) using a training set of dialogue examples. Intuitively, a fact will therefore be deemed as relevant if its inclusion in the prompt makes it relatively easier for the generation model to produce the correct response. To ensure the inference remains efficient, Eq. (1) is simplified by sampling the K most relevant facts instead of marginalizing over all possible facts.

3.4 Response Generation

The final step of our approach is to generate a response y based on both the current dialogue history x and a set of relevant facts $z_1 : z_K$, where K denotes the number of facts (sorted by relevance) to include in the input prompt. Any pretrained language model can be employed for this task. We rely for our experiments on both the GODEL model (Peng et al., 2022) which is specifically designed for goal-oriented dialog as well the generic GPT 3.5 model (Brown et al., 2020).

4 Evaluation

We evaluate the performance of the proposed approach on two existing dialogue datasets along with a human evaluation. We present below the experimental design, and discuss the results.

4.1 Datasets

GraphWOZ (Walker et al., 2022)

GraphWOZ contains task-oriented dialogue with dialogues discussing people and places in a fictional organization to schedule meetings and discover information. Each dialogue is paired with synthetically generated calendar events. The graphs contain fictive people, rooms, and events along with dialogue information such as utterances and mentions of entities in utterances.

Although the calendar information is synthetically generated, generation of new dialogue utterances with calendar information may not accurately reflect real system-human interaction. In consideration of this factor and to compensate for the small amount of training data, we augment the GraphWOZ training set with modified versions of the original dialogues where entities are replaced in both the knowledge base and dialogue history. We replace entity names with randomly sampled replacements, and the dates and times of the dialogues and events in the KBs are replaced such that relative terms such as "today", "tomorrow", "morning", and "afternoon" remain consistent in the modified dialogue.

The entity linking and commonsense rules for this dataset are provided in the Appendix.

KVRET (Eric et al., 2017)

This dataset contains task-oriented dialogue in three domains: weather, navigation, and calendar scheduling. Each type of dialogue contains associated KB information representing objects of interest which are discussed in the dialogue. The knowledge bases in KVRET were created by randomly sampling attribute values for defined slots according to the domain. We convert these KBs into a ProbLog program along with the user utterances and mentions of objects. For simplicity, we take a string equality match of a substring in the utterance to an object in the KB as a `refers_to` relation.

We rely on three simple ProbLog rules for this dataset. The weather domain has a rule which determines "today" along the weather for a particular day and location. In the calendar scheduling domain, we provide a rule to handle location names with multiple potential referents. Lastly, for the navigation domain we define a rule comparing the distance from the user to two points of interests and determining which one is closest. Each of

Model	Dev			Test		
	BLEU	METEOR	BERTScore	BLEU	METEOR	BERTScore
$\text{GODEL}_{\text{NoFacts}}$	0.17	0.37	0.89	0.11	0.36	0.88
$\text{GODEL}_{\text{AllFacts}}$	0.14	0.38	0.88	0.13	0.33	0.88
$\text{GODEL}_{\text{Relevance}}$	0.18	0.38	0.89	0.14	0.33	0.88
$\text{GODEL}_{\text{Relevance+Logic}}$	0.17	0.37	0.89	0.16	0.35	0.88
$\text{GPT}_{\text{NoFacts}}$	0.08	0.35	0.88	0.06	0.32	0.87
$\text{GPT}_{\text{AllFacts}}$	0.07	0.36	0.88	0.06	0.32	0.87
$\text{GPT}_{\text{Relevance}}$	0.07	0.35	0.88	0.06	0.35	0.87
$\text{GPT}_{\text{Relevance+Logic}}$	0.07	0.37	0.88	0.07	0.36	0.87

Table 1: Results with reference-based metrics on the development and test set of GraphWOZ.

Model	Dev			Test		
	BLEU	METEOR	BERTScore	BLEU	METEOR	BERTScore
$\text{GODEL}_{\text{NoFacts}}$	0.18	0.45	0.91	0.11	0.36	0.91
$\text{GODEL}_{\text{Relevance}}$	0.18	0.42	0.91	0.16	0.41	0.91
$\text{GODEL}_{\text{Relevance+Logic}}$	0.20	0.43	0.91	0.17	0.42	0.91

Table 2: Results with reference-based metrics on the development and test set of KVRET.

these rules therefore makes information explicitly available to the system which would be unavailable from context or otherwise require logical inference that an LLM is not optimized to perform.

4.2 Models

We experiment with the four following types of response generation models:

NoFacts Generation model that does not use the knowledge graph at all and produce a response based on the current dialogue history.

AllFacts+Logic Generation model using all verbalized facts (including logically derived ones), without relevance scoring. These facts are shuffled and truncated to fit into the context window of the generation model.

Relevance Generation model using the initial facts from the knowledge graph (but without logically derived ones) ranked using the relevance scoring model. The 10 most relevant facts are then prepended to the prompt.

Relevance+Logic Generation model using both the initial facts and the logically derived ones, along with the relevance scoring model to select the 10 most relevant facts.

We experiment with two generative models: the encoder-decoder GODEL (Peng et al., 2022), which is pre-trained on large volumes of multi-turn dialogues, and the recent GPT-3.5 model (Brown et al., 2020). We first test the response generation capabilities of GPT-3.5 with the three different approaches on GraphWOZ. For each turn, we provide the system with the dialogue history up to the current turn. When using all facts, background knowledge is added as a single document in the initial prompt, as repeating the entire document of the facts at each turn would result in truncation of the dialogue history without adding additional information.

4.3 Metrics

For both GraphWOZ and KVRET, we use standard evaluation metrics such BLEU, METEOR and the averaged BERTScore F1 (Zhang et al., 2019b). We also use the recently introduced UniEval (Zhong et al., 2022), a reference-free metric which has been shown to correlate well with human judgments.

We also evaluate the *factuality* of the responses by manually annotating them with two types of error. The first error type are *hallucinations*, which we define as either (a) a statement that contradicts the KB, including contradictions implied by the dialogue context ; (b) a statement referring to a nonexistent entity in the KB ; or (c) a statement describing a calendar action that would create a calendar conflict if enacted.

Model	**Coherence**	**Groundedness**	**Naturalness**	**Understandability**
$\text{GODEL}_{\text{NoFacts}}$	0.946	0.908	0.871	0.864
$\text{GODEL}_{\text{AllFacts}}$	0.975	0.943	0.903	0.896
$\text{GODEL}_{\text{Relevance}}$	0.916	0.878	0.862	0.855
$\text{GODEL}_{\text{Relevance+Logic}}$	0.979	0.951	0.868	0.861
$\text{GPT}_{\text{NoFacts}}$	0.951	0.880	0.943	0.938
$\text{GPT}_{\text{AllFacts}}$	0.952	0.878	0.931	0.925
$\text{GPT}_{\text{Relevance}}$	0.969	0.912	0.935	0.931
$\text{GPT}_{\text{Relevance+Logic}}$	0.949	0.883	0.928	0.922

Table 3: UniEval Score (Reference-free) on the test set of GraphWOZ.

Model	**Dev**		**Test**	
	Hallucinations	Retrieval Errors	Hallucinations	Retrieval Errors
$\text{GPT}_{\text{NoFacts}}$	34 (18%)	17 (9%)	32 (17%)	16 (8%)
$\text{GPT}_{\text{AllFacts}}$	23 (13%)	11 (6%)	23 (13%)	20 (11%)
$\text{GPT}_{\text{Relevance}}$	21 (12%)	13 (7%)	24 (13%)	16 (8%)
$\text{GPT}_{\text{Relevance+Logic}}$	15 (8%)	14 (8%)	25 (14%)	9 (5%)

Table 4: Turns containing hallucinations and retrieval errors (GraphWOZ, 181 turns in Dev, 180 in Test)

The second type of error occurs when the system fails to retrieve information it has access to or should have access to. We denote this type of error as a *retrieval error*, and we annotate a turn as containing a retrieval error if it contains:

- A statement denying having access to information which exists in the calendar, or does not answer a question while answering another ;
- A confirmation of inviting a person to a meeting without specifying whom ;
- A request to confirm a piece of information that was already confirmed.

4.4 Results

Tables 1 and 2 respectively provide the results of the reference-based metrics on GraphWOZ and KVRET. We observe modest differences between the models, particularly on the development set. This is expected as those metrics are not really suited to measure the extent to which a language model produces factual answers. We do nevertheless observe a slight increase in BLEU score when logic-derived facts are added. This result is consistent in both GraphWOZ and KVRET. The differences in UniEval, shown in Table 3 were also small, with the model using all facts performing comparably to the Relevance+Logic model.

However, for the manually annotated hallucinated and retrieval errors in Table 4 (only annotated for the GPT models), we do observe a substantial reduction in overall errors for the Relevance+Logic model compared to the baselines. Particularly in the development set, the model using all facts (without relevance scoring) frequently produce hallucinations, such as incorrect attributes of various people and events.

4.5 Error Analysis

Qualitatively, the errors produced by the models also differed. The GPT models using selected facts were more likely to provide all available information, whether requested or not. In some cases, the model produced statements connecting unrelated facts and implying an error, for example:

> *User*: Hi robot, my name is Curtis Williams, please let me know when the meeting called, deliverables team meeting. [*sic*]
> *System*: Hello Curtis! The deliverables team meeting will be held on Friday, April 08 at 11:30 AM in the Beta conference room. Annette Harding is organizing the meeting on behalf of the Mathematics group. You and Angela Jimenez are scheduled to attend the meeting. The meeting is scheduled to last for 30 minutes until the Niches team meeting starts at 12:00. Michael Glover is attending the Niches team meeting and Annette Harding is also attending the Deliverables team meeting with you. Is there anything else you need assistance with?

Although the general content is factual, the niches team meeting was on a different day despite the implication of the phrasing. The GPT model with all facts responded erroneously at the same turn as:

> Alright Curtis, I have checked your calendar and it looks like there is a "Deliverables Team Meeting" scheduled for tomorrow at 2:00 pm in room 619. Would you like me to add any notes or reminders for the meeting?"

While more concise, the system incorrectly states the time of the event (11:30) and date (today), as well as the location (Beta). This apparently arises from misattribution of background facts relating to other events scheduled across times and locations.

4.6 Human Evaluation

Experimental setup

To confirm the performance of the approach in actual interactions, we also conduct a human evaluation in the context of a receptionist scenario similar to GraphWOZ, where the participant interacts with the system to find information about entities and schedule events. We recruited 16 participants including students from the university and employees to interact with the dialogue systems through text. Users were instructed to interact with the system to accomplish a task, mark the conversation as finished when either the task appeared complete or the dialogue system unrecoverably failed. After each dialogue, the users were prompted to rate the dialogue on a scale of 1 to 5 for two statements, where 1 is "Never", 2 is "Mostly Not", 3 "Sometimes", 4 "Mostly", and 5 "Always":

- *The system responded to me in a conversationally relevant way.*
- *The system successfully completed my task and gave me the information I asked for.*

Users were instructed to repeat this process for 30 minutes, with priority given to conversation quality. For each dialogue, a model was randomly selected and a random dialogue state similar to the GraphWOZ dataset was generated for the dialogue. A task was then randomly generated from a set of task templates involving fictive entities. The collected dialogues were then manually annotated for both hallucinations and retrieval errors. Because the total number of turns varied from model to model, we evaluate the proportion of turns which contain hallucinations and retrieval errors.

Model	**Task**	**Appropriateness**
$\text{GODEL}_{\text{None}}$	3.35	3.07
$\text{GODEL}_{\text{All}}$	3.75	3.63
$\text{GODEL}_{\text{Logic}}$	4.08	3.75
GPT_{None}	4.18	4.59
GPT_{All}	4.09	4.32
$\text{GPT}_{\text{Logic}}$	4.37	4.11

Table 5: Average participant scores for the model task completion and appropriateness criteria.

Model	**Hallucinations**	**Retrieval errors**	**#**
$\text{GODEL}_{\text{None}}$	0.17	0.41	105
$\text{GODEL}_{\text{All}}$	0.24	0.32	84
$\text{GODEL}_{\text{Logic}}$	0.22	0.21	67
GPT_{None}	0.32	0.14	88
GPT_{All}	0.22	0.39	117
$\text{GPT}_{\text{Logic}}$	0.20	0.12	132

Table 6: Proportion of system responses containing either hallucinations or retrieval errors in the human evaluation experiments. The last column indicates the total number of system utterances from all dialogues with that model.

Results

As for the GraphWOZ results, the human interaction experiments indicate a reduction in the proportion of turns with hallucinations or retrieval errors, as shown in Table 6. This reduction is observed for both model types when the logic-enhanced relevance scoring model was used.

The participant scores in Table 5 ranked the models which used the relevance scored facts highest. While the GODEL model using the relevant facts scored higher in appropriateness, the opposite pattern is observable in the GPT models, although the scores remain relatively high. As not every participant interacted with every model, differences in scoring between individual participants cannot be discounted as a factor impacting these results, thus a larger study would be beneficial.

5 Conclusion

This paper presented a novel approach to retrieval-augmented response generation in task-oriented dialogue systems. The approach relies a dynamic knowledge graph representing the dialogue state, which is enriched at each turn with facts derived from a small set of rules specified in the ProbLog language. Those facts are then ranked by relevance using a dedicated scoring model which accounts for both the semantic similarity and conversational saliency of each fact. The most relevant facts are then incorporated to the background knowledge provided as input to the response generation model.

We provide experimental results showing that the combination of logical reasoning with a relevance scoring model leads to more factual responses. In particular, the logical rules seem to assist the generation model's ability to provide responses grounded in multi-step reasoning based on the available background knowledge.

The proportion of errors remains, however, relatively high, likely due to the very limited number of dialogues available for training in GraphWOZ and KVRET. Future work will focus on evaluating the potential of this approach in other (and potentially broader) dialogue domains.

References

Giovanni Bonetta, Rossella Cancelliere, Ding Liu, Paul Vozila, et al. 2021. Retrieval-augmented transformer-xl for close-domain dialog generation. In *The International FLAIRS Conference Proceedings*, volume 34, pages 1–4. Florida Online Journals.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019. Retrieval-guided dialogue response generation via a matching-to-generation framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1866–1875, Hong Kong, China. Association for Computational Linguistics.

Mark Chavira and Adnan Darwiche. 2008. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799.

Zhiyu Chen, Bing Liu, Seungwhan Moon, Chinnadhurai Sankar, Paul Crook, and William Yang Wang. 2022. KETOD: Knowledge-enriched task-oriented dialogue. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2581–2593, Seattle, United States. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. 2023. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. 2015. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401.

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 AAAI Conference on Artificial Intelligence*, volume 32.

Bernd Gutmann, Ingo Thon, and Luc De Raedt. 2011. Learning the parameters of probabilistic logic programs from interpretations. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11*, pages 581–596. Springer.

Huang He, Hua Lu, Siqi Bao, Fan Wang, Hua Wu, Zhengyu Niu, and Haifeng Wang. 2021. Learning to select external knowledge with multi-scale negative sampling. *arXiv preprint arXiv:2102.02096*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. In *International Conference on Learning Representations*.

Angelika Kimmig, Bart Demoen, Luc De Raedt, Vitor Santos Costa, and Ricardo Rocha. 2011. On the implementation of the probabilistic logic programming language problog. *Theory and Practice of Logic Programming*, 11(2-3):235–262.

Hayoung Lee and Okran Jeong. 2023. A knowledge-grounded task-oriented dialogue system with hierarchical structure for enhancing knowledge selection. *Sensors*, 23(2):685.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Linxiao Li, Can Xu, Wei Wu, Yufan Zhao, Xueliang Zhao, and Chongyang Tao. 2020. Zero-resource knowledge-grounded dialogue generation. *Advances in Neural Information Processing Systems*, 33:8475–8485.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.

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. *arXiv preprint arXiv: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 (EMNLP-IJCNLP)*, pages 1782–1792, Hong Kong, China. Association for Computational Linguistics.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

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 Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy. Association for Computational Linguistics.

Ashwin Paranjape, Omar Khattab, Christopher Potts, Matei Zaharia, and Christopher D Manning. 2021. Hindsight: Posterior-guided training of retrievers for improved open-ended generation. In *International Conference on Learning Representations*.

Prasanna Parthasarathi and Joelle Pineau. 2018. 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, Brussels, Belgium. Association for Computational Linguistics.

Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. Godel: Large-scale pre-training for goal-directed dialog. *arXiv preprint arXiv:2206.11309*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

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*. Association for Computational Linguistics.

Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. Thinking globally, acting locally: Distantly supervised global-to-local knowledge selection for background based conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8697–8704.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Cameron Smith, Nigel Crook, Daniel Charlton, Johan Boye, Raul Santos De La Camara, Markku Turunen, David Benyon, Björn Gambäck, Oli Mival, Nick Webb, et al. 2011. Interaction strategies for an affective conversational agent. *Presence*, 20(5):395–411.

David Thulke, Nico Daheim, Christian Dugast, and Hermann Ney. 2021. Efficient retrieval augmented generation from unstructured knowledge for task-oriented dialog. *arXiv preprint arXiv:2102.04643*.

Jonas Vlasselaer, Joris Renkens, Guy Van den Broeck, and Luc De Raedt. 2014. Compiling probabilistic logic programs into sentential decision diagrams. In *Proceedings Workshop on Probabilistic Logic Programming (PLP)*, pages 1–10.

Nicholas Thomas Walker, Stefan Ultes, and Pierre Lison. 2022. Graphwoz: Dialogue management with conversational knowledge graphs. *arXiv preprint arXiv:2211.12852*.

Hao Wang, Bin Guo, Wei Wu, Sicong Liu, and Zhiwen Yu. 2021. Towards information-rich, logical dialogue systems with knowledge-enhanced neural models. *Neurocomputing*, 465:248–264.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022a. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38.

Wenhao Yu, Chenguang Zhu, Zhihan Zhang, Shuohang Wang, Zhuosheng Zhang, Yuwei Fang, and Meng Jiang. 2022b. Retrieval augmentation for commonsense reasoning: A unified approach. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4364–4377, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haolan Zhan, Lei Shen, Hongshen Chen, and Hainan Zhang. 2021. CoLV: A collaborative latent variable model for knowledge-grounded dialogue generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2250–2261, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2019a. Grounded conversation generation as guided traverses in commonsense knowledge graphs. *arXiv preprint arXiv:1911.02707*.

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 (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2021. Joint retrieval and generation training for grounded text generation. *arXiv preprint arXiv:2105.06597*.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

A.1 Entity linking rules

The ProbLog rules employed for entity linking are given below. The probabilities attached to the rules are estimated empirically using the parameter estimation approach implement in ProbLog library, based on Learning from Interpretations (Gutmann et al., 2011).

```
0.60838635::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), jw_similarity(N,S,O), O>0.9.

0::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), jw_similarity(N,S,O), O>0.8.

0::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), jw_similarity(N,S,O), O>0.7.

0.72255423::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), lev_distance(N,S,O), O < 2.

0.30394455::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), lev_distance(N,S,O), O < 3.

0::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), lev_distance(N,S,O), O < 6.

0.0019686::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), lcs(N,S,O), O > 3.

0::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), string(M,S), lcs(N,S,O), O > 6.

0::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), nb_common_words(N,S,O), O > 0.

0::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), nb_common_words(N,S,O), O > 1.

0::refers_to(M,E) :- new(U), mention(U,M), string(M,S),
    is_processable_time(S,0), name(E,N), nb_common_words(N,S,O), O > 2.

0.27142172::refers_to(M,E) :- new(U), mention(U,M), respond_to(U,AR1),
    mention(AR1,PM1), refers_to(PM1, E).

0.12752306::refers_to(M,E) :- new(U), mention(U,M), respond_to(U,AR1),
    respond_to(AR1,PU1), mention(PU1,PM1), refers_to(PM1, E).

0.07429096::refers_to(M,E) :- new(U), mention(U,M), respond_to(U,AR1),
    respond_to(AR1,PU1), respond_to(PU1,AR2), mention(AR2,PM1), refers_to(PM1, E).

0.01403269::refers_to(M,E) :- new(U), mention(U,M), respond_to(U,AR1),
    respond_to(AR1,PU1), respond_to(PU1,AR2), respond_to(AR2,PU2),
mention(PU2,PM2), refers_to(PM1, E).
```

A.2 Commonsense rules

The rules employed for commonsense reasoning on the GraphWOZ dialogues are provided below.

```
event_today(E,T) :- event(E), start_time(E,T), date(at_today,D), date(E,D).

event_tomorrow(E,T) :- event(E), start_time(E,T), date(at_tomorrow,D), date(E,D).

person_group(P,G) :- people(P), group(P,G).

group_members(G,L) :- group(G), findall(P, person_group(P,G), L).

count_members(G,N) :- group(G), refers_to(M,G), group_members(G,L), list_length(L,N).
```

```
room_available_today(R,T) :- room(R), \+(location(E,R), date(E,D), date(at_today,D),
    start_time(E,ST), end_time(E,ET), time_between(T,ST,ET,1)).

room_available_tomorrow(R,T) :- room(R), \+(location(E,R), date(E,D),
    date(at_tomorrow,D), start_time(E,ST), end_time(E,ET), time_between(T,ST,ET,1)).

room_available_now(P) :- room(P), \+(room_busy_now(P)).

room_busy_now(P) :- room(P), time(at_now,T), attendee(E,P), date(E,D),
    date(at_today,D), start_time(E,ST), end_time(E,ET), time_between(T,ST,ET,1).

person_available_today(P,T) :- refers_to(M,P), people(P),
    string(_,T), is_time_expression(T,1), \+(person_busy_today(P,T)).

person_busy_today(P,T) :- refers_to(M,P), people(P), string(_,T),
    is_time_expression(T,1), attendee(E,P), date(E,D), date(at_today,D),
    start_time(E,ST), end_time(E,ET), time_between(T,ST,ET,1).

person_available_tomorrow(P,T) :- refers_to(M,P), people(P), string(_,T),
    is_time_expression(T,1), \+(person_busy_tomorrow(P,T).

person_busy_tomorrow(P,T) :- refers_to(M,P), people(P), string(_,T),
    is_time_expression(T,1), attendee(E,P), date(E,D), date(at_tomorrow,D),
    start_time(E,ST), end_time(E,ET), time_between(T,ST,ET,1).

person_available_now(P) :- refers_to(M,P), people(P), time(at_now,T),
    \+(person_busy_now(P)).

person_busy_now(P) :- refers_to(M,P), people(P), time(at_now,T), attendee(E,P),
    date(E,D), date(at_today,D), start_time(E,ST), end_time(E,ET),
    time_between(T,ST,ET,1).

attending_today(E,P) :- attendee(E,P), date(E,D), date(at_today,D).

person_events_today(P,L) :-  refers_to(M,P), people(P),
    findall(X, attending_today(X,P), L).

attending_tomorrow(E,P) :- attendee(E,P), date(E,D), date(at_tomorrow,D).

person_events_tomorrow(P,L) :-  refers_to(M,P), people(P),
    findall(X, attending_tomorrow(X,P), L).

available_rooms_now(L) :- findall(R, room_available_now(R), L).

available_rooms_today(L,T) :- string(_,M), morning_time(M,1), between(8,11,T),
    findall(R, room_available_today(R,T), L).

available_rooms_tomorrow(L,T) :- string(_,M), morning_time(M,1), between(8,11,T),
    findall(R, room_available_tomorrow(R,T), L).

available_rooms_today(L,T) :- string(_,M), afternoon_time(M,1), between(12,17,T),
    findall(R, room_available_today(R,T), L).

available_rooms_tomorrow(L,T) :- string(_,M), afternoon_time(M,1), between(12,17,T),
    findall(R, room_available_tomorrow(R,T), L).

time_place(E,D,T) :- refers_to(M,E), event(E), date(E,D), start_time(E,T).
```