

# Retrieving Relevant Knowledge Subgraphs for Task-Oriented Dialogue

Nicholas Thomas Walker<sup>1</sup>, Pierre Lison<sup>2</sup>, Laetitia Hilgendorf<sup>1</sup>,  
Nicolas Wagner<sup>1</sup>, Stefan Ultes<sup>1</sup>

<sup>1</sup>Otto-Friedrich University of Bamberg, <sup>2</sup>Norwegian Computing Center  
{nicholas.walker,nicolas.wagner,stefan.ultes}@uni-bamberg.de  
plison@nr.no  
laetitia-mina.hilgendorf@stud.uni-bamberg.de

## Abstract

Knowledge graphs are a rich source of useful background information for RAG-based dialogue systems, but the accidental inclusion of extraneous information harms model performance. Thus, retrieval models must carefully balance quality and quantity of retrieved information. To address this problem, we investigate a method of retrieving relevant subgraphs of maximum relevance and minimum size by framing this trade-off as a Prize-collecting Steiner Tree problem. We evaluate this approach with an analysis of subgraph retrieval with existing data along with a user study and evaluation. The results of our user study and analysis indicate promising efficacy of a simple subgraph retrieval approach compared with a top- $k$  retrieval model.

## 1 Introduction

Contemporary approaches in NLP using large language models (LLMs) often employ retrieval augmented generation (RAG) as a means of improving LLM output without the need for costly fine-tuning (Lewis et al., 2020). However, studies have shown that the inclusion of irrelevant information harms system performance (Shi et al., 2023; Ding et al., 2024). LLMs in dialogue systems may also make use of RAG to include additional context to assist the model to accurately answer queries or otherwise provide more interesting conversational content (Dinan et al., 2018; Chaudhuri et al., 2021). This information is often drawn from *knowledge graphs* (KGs), which are a flexible structure for representing background information seeing application in question-answering and dialogue (Li et al., 2022; Huang et al., 2019; Peng et al., 2023). Particularly in task-oriented dialogue, RAG can reduce hallucinations (Shuster et al., 2021) and retrieve task-specific information. Systems relying on KGs often require filtering methods to identify useful information (Thulke et al., 2021). Thus, RAG re-

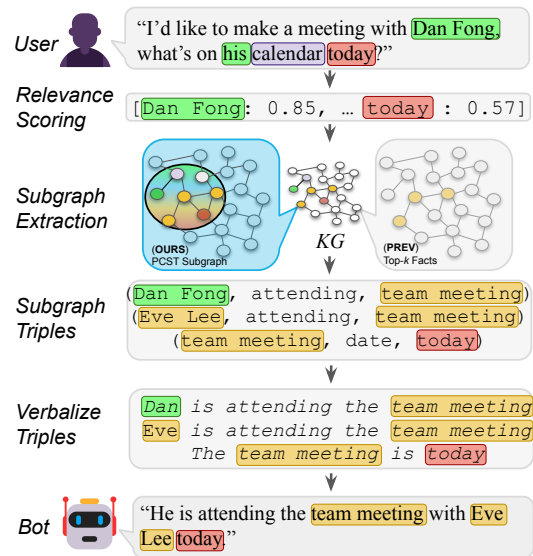


Figure 1: Subgraph retrieval methods in a RAG pipeline for task-oriented dialogue. Given input dialogue context, the graph elements are scored for relevance to context. Our model using the Prize-Collecting Steiner Tree (PCST) algorithm then retrieves a subgraph maximizing the reward (relevance) of nodes while minimizing a cost assigned to edges yielding the smallest subgraph with the greatest reward, in contrast to top- $k$  retrieval.

quires methods that flexibly retrieve maximal relevant information with minimal extraneous items.

In light of this challenge, we investigate a graph-based approach to maximizing relevant information retrieval from knowledge graphs while constraining the total amount of retrieved information. Recent work in question answering suggests that retrieval methods based on constructing *subgraphs* within a knowledge graph are effective for mitigating model hallucinations (He et al., 2024). Inspired by such retrieval mechanisms, we present a study of an approach to subgraph retrieval framed as a *Prize-Collecting Steiner Tree* (PCST) problem.

While the information of certain nodes and edges of the graph have relevance for generating an ap-

appropriate system response at a given turn, it is also possible for larger structures such as paths to represent complex interactions in the background knowledge. A sequence of nodes and edges may only be useful in combination with each other, and the exclusion of intervening elements between those that are retrieved may omit important information. Although the simplicity of a classic top- $k$  information retrieval is attractive for many applications, when it is applied to graphs, structural information is disregarded in the information retrieval and lost in translation to the LLM prompt. For this reason, we investigate whether subgraph-based retrieval provides more relevant information compared to a system retrieving and ranking individual edges without the specific graph structure.

In this work, we present a study comparing PCST subgraph retrieval against top- $k$  retrieval of KG information. Our work presents the three following contributions:

- An implementation of a simple but effective RAG pipeline for task-oriented dialogue using PCST-based subgraph retrieval.
- An experimental analysis of the PCST-based retriever using a task-oriented dialogue dataset with respect to both retrieval performance on annotated data and overall relevance of retrieved subgraphs.
- A user study comparing task-oriented dialogue systems using the PCST-based retriever against a top- $k$  baseline.

The experimental evaluation focuses on the relative utility of subgraph retrieval using PCST compared against commonly used top- $k$  retrieval which does not make use of graph structure, demonstrating the utility of the proposed method in a task-oriented dialogue setting.

The remainder of this paper is structured as follows: In Section 2, we review related approaches to retrieval in knowledge graphs within question answering and dialogue systems. Next, in Section 3 we outline the proposed approach to retrieval formulated as a Prize-Collecting Steiner Tree, and how this is implemented. Section 4 then contains the description of our experiments using the GraphWOZ dataset and our user experiment, along with our analysis of the results. Lastly, we outline the conclusions of our work in Section 5.

## 2 Related Work

Dialogue systems using RAG encompass both the use of large, static KGs of general information such as ones derived from dumps of Wikipedia (Dinan et al., 2018) as well as smaller dynamic KGs representing dialogue-specific information such as mental states or logical inferences (Qiu et al., 2022). An example of task-oriented dialogue using dialogue-specific KGs is the Key-Value Retrieval Networks of Eric et al. (2017). Meanwhile, Ghazvininejad et al. (2018) used a sequence-to-sequence model conditioned on dialogue history and background facts to generate more appropriate responses. Other models have used directly copied background information in response generation (He et al., 2023). To use KG information in an LLM, a conversion to natural language through the use of templates can enhance the quality of the model’s responses (Wu et al., 2023).

Subgraph retrieval specifically is a topic of increasing interest in both question answering (QA) and dialogue settings. Within QA, the G-Retriever model of He et al. (2024) made use of PCST subgraphs as a component that specifically informed our approach. Other work includes that of Zhang et al. (2022), who presented a separate trained subgraph retriever to provide results for a QA model. Additionally, Zhu et al. (2025) presented a subgraph retrieval model for QA targeted in part at reducing ambiguities and redundancies within the KG during retrieval.

As an example of subgraph retrieval in dialogue systems, Kang et al. (2022) presented the SURGE model, which utilized a mechanism to enforce consistency of embeddings between dialogue context and retrieved subgraphs. Likewise, Park et al. (2024) evaluated a method of subgraph retrieval on the OpenDialKG dataset. Some work has also investigated subgraph retrieval with respect to graphs incorporating commonsense reasoning (Wu et al., 2024). In contrast to models using trained components, our subgraph-retrieval eschews training neural models to directly use PCST subgraph construction along with out-of-the-box methods for scoring relevance in the graph.

## 3 Subgraph Retrieval

We propose a model of retrieval that constructs a subgraph using an algorithm to solve a Prize-collecting Steiner Tree problem (specifically the Goemans-Williamson algorithm, described in Sec-

tion 3.2). Let us assume an initial knowledge graph  $G = (V, E)$  where  $V$  denotes vertices (nodes) and  $E$  denotes edges across those vertices. The task of subgraph retrieval is to identify a subgraph  $G^* = (V^*, E^*)$  which is most relevant to the user query  $q$ . In the subgraph  $G^*$ ,  $V^* \subset V, E^* \subset E$ , and all edges in  $E^*$  connect nodes in  $V^*$ .

In a Prize-collecting Steiner Tree (PCST) problem, each node  $v$  is associated with a positive reward  $\pi(v) > 0$  (also called *prize*), while each edge  $e$  has a cost  $c(e) > 0$ . Solving a given PCST problem then involves finding the subgraph  $G^* = (V^*, E^*)$  that maximizes the total prize minus the costs of the edges of this subgraph:

$$G^* = \underset{G'=(V' \subset V, E' \subset E)}{\operatorname{argmax}} \sum_{v' \in V'} \pi(v') - \sum_{e' \in E'} c(e') \quad (1)$$

The subgraph extraction can be used as the retriever in a RAG pipeline, as shown in Figure 1, serving to retrieve relevant information from the knowledge graph that can then be inserted into an LLM prompt along with dialogue context. In our experiments, we assign a prize to graph elements based on a *verbalization* of their contents compared against the dialogue context to assess relevance. To assign these relevance scores, we experiment with both a neural model and a heuristic model. As a baseline, we evaluate a method that retrieves graph nodes and edges independently based solely on their predicted relevance, without the requirements of the PCST problem. This approach is conceptually similar to other approaches to retrieval in task-oriented using graph verbalizations (Walker et al., 2023). For our baseline, we use the same hyperparameter of  $k = 5$  top facts retrieved as evaluated in that work.

We evaluate these retrieval models using the GraphWOZ dataset (Walker et al., 2022), which contains dialogues associated with dialogue specific, dynamic KGs containing commonsense reasoning relations for the dialogue turns. This dataset contains a set of dialogues paired with task-specific background information stored in knowledge graphs along with annotations of information such as user intents. The size of the dataset along with the existing annotations of e.g. user-intent allow us to annotate likely-relevant nodes and edges at each turn in order to evaluate our approach. Following the analysis with this dataset, we conducted an experiment using similarly structured tasks in a live setting to collect user impressions of the sys-

tem’s performance.

### 3.1 Relevance & Verbalization

At each turn of the dialogue, only a subset of the KG information is likely to usefully contribute to the next response in the dialogue. From the system’s perspective, we wish to retrieve the maximum number of relevant elements while minimizing non-relevant elements. Simple attributes such as names may easily be retrieved as individual elements without information from their neighborhood. For instance, the question “Who is organizing the team meeting?” could be answered with the information in a single edge (“*John Smith* | organizing | “*team meeting*”).

However, more complex interactions might require multiple edges describing several entities and their relations. For example, “What times are there events scheduled in the Omega conference room today?” requires information from a larger subgraph connecting the conference room, events, times and the appropriate date. These elements are useful to the system in combination, and retrieving nodes and edges in isolation may lose information about useful long distance relations in the KG. An additional challenge is the absence of relevance labels.

For modelling relevance in our approach, each node and edge is assigned an individual relevance score with respect to the dialogue context. To calculate such scores, we convert the nodes and edges to text using verbalization templates, which result in a text form that can be inserted into the system prompt. This conversion of graph information to text allows for measures like cosine similarity or information retrieval heuristics to be used as the basis for an initial estimate of relevance.

To convert nodes and edges to text, we use templates based on the node and edge types. Node verbalization is based on the name (label) describing the entity, e.g. “John Smith” or “team meeting” inserted into a pre-defined template associated with the node type. Thus, the node for “John Smith” is verbalized as *John Smith is a person*. For edges, the verbalization is a pre-defined function which constructs a simple sentence using the semantic label of the edge between the start and end node. An example would be an edge (“*John Smith* | attending | “*team meeting*”) verbalized as *John is attending team meeting*. Each triple in the knowledge graph is thereby converted to text using a template associated with its semantic label.

This setup for assigning relevance can be viewed

as a standard document retrieval problem. We test two models to assign scores: The Okapi BM25 algorithm (Robertson et al., 2009) and a sentence-BERT model optimized for QA (Reimers and Gurevych, 2019). Using the scoring model, we calculate a score based on the the dialogue context and the verbalization of each node and edge of the graph. While the user utterance of the current turn is likely to be the most salient for selecting relevant entities, previously mentioned entities and information may also be useful. For this reason, we calculate a similarity score using several different lengths of dialogue history and average them. The similarity score of a graph element  $s$  (node or edge) to the dialogue context at time  $t$ , denoted  $\mathbf{c}_t$ , is thus calculated as:

$$\text{sim}(s, \mathbf{c}_t) = \frac{\sum_{i=0}^L \text{sim}(s, \mathbf{c}_{t-i:t})}{L} \quad (2)$$

where  $\mathbf{c}_{t-i:t}$  denotes the dialogue context with the most recent  $i$  turns up to time  $t$ ,  $\text{sim}$  is a similarity measure, and  $L$  is the maximum number of turns from the dialogue history to include when computing the similarity score (set to 3 in our experiments).

### 3.2 Subgraph Construction

Given a graph, the Goemans-Williamson algorithm for the PCST problem seeks to construct a connected subgraph  $G^* = (V^*, E^*)$  that maximizes the “prize” (or reward)  $\pi(v)$  of the nodes in the subgraph while minimizing the cost  $c(e)$  of its edges, requiring that the resulting subgraph be connected (Hegde et al., 2014). This algorithm is designed for the *unrooted* PCST problem where no specific node is required to be a part of the solution. The details of this algorithm are described in detail by Hegde et al. (2014). Similarly to the procedure used by He et al. (2024), we assign prize values to the nodes and edges of the knowledge graph as follows. We first calculate the relevance of the graph verbalizations described in Section 3.1. With a parameter  $k$  for the number of nodes and edges, we assign a prize of  $k - i$  where  $i$  is the rank of the node or edge ordered by cosine similarity. We set this parameter as  $k = 3$  nodes and  $k = 5$  edges.

In the original PCST problem, only nodes are assigned to reward values, while edges are given a cost  $C > 0$ . However, the semantic relationships described by the edge labels are crucial to understanding the information contained in the KG. Following a similar approach to He et al. (2024),

we convert each edge to a “virtual” node bearing label of the original edge and verbalize it as described in Section 3.1. This results in two new edges connecting the virtual node to the original start and end node of the triple. Therefore, each of these new edges has half the cost assigned by an original edge cost parameter  $C$ . It should also be noted that for the PCST problem, edge *direction* is not considered (all edges therefore treated as undirected, in contrast with our verbalization procedure). In our experiments, we assign the initial edge cost  $C = 1$  (thus each edge connecting the virtual node receives a cost  $C = 0.5$ ). The resulting virtual nodes can then be assigned a relevance score. To implement the subgraph retrieval in our experiments, we used the *pcst\_fast* library<sup>1</sup>.

## 4 Experiments

We assess the performance of proposed approach in two distinct sets of experiments. We first evaluate the PCST-based approach to subgraph retrieval by comparing it against a baseline top- $k$  retrieval on gold-standard annotations of existing data. Additionally, we analyze the models with reference to the overall relevance scores of retrieved graphs. For the second part of our work, we deploy the subgraph retrieval method with live users in a task-oriented setting and obtain their evaluations of the system relative to a baseline.

### 4.1 Retrieving gold-standard subgraphs

For the first experiments, we annotate the GraphWOZ dataset for relevant triples to the user question. While this annotation presents a challenge in defining which nodes are relevant (and indeed, whether relevance should be a binary value), the annotation gives an estimate of which nodes and edges should be retrieved by the graph given the dialogue context. As the dataset size is relatively modest, this annotation extension to GraphWOZ was performed manually by the first author.

#### Annotation procedure

We annotate the graph for each turn as follows. Entities mentioned by the user with a relation to a second entity which is expressed as an edge in the graph have this edge along with their start and end node labelled as relevant. Otherwise, an entity mentioned individually without specifying any relations to other entities or attributes has all edges connected to it with their start or end node labelled as

<sup>1</sup>[https://github.com/fraenkel-lab/pcst\\_fast](https://github.com/fraenkel-lab/pcst_fast)

Model	Dev			Test		
	Precision	Recall	F1	Precision	Recall	F1
Triples <sub>BM25</sub>	0.352	<b>0.342</b>	0.347	0.412	<b>0.385</b>	0.398
PCST <sub>BM25</sub>	<b>0.444</b>	0.320	<b>0.372</b>	<b>0.496</b>	0.372	<b>0.425</b>
Triples <sub>SBERT</sub>	0.317	0.314	0.316	0.320	0.313	0.316
PCST <sub>SBERT</sub>	0.383	0.281	0.325	0.351	0.278	0.310

Table 1: Precision, Recall, and F1 on turns of the development and test sets of GraphWOZ containing relevant nodes and/or edges. These scores are averaged over  $n = 100$  runs due to the possibility of nodes receiving equal relevance scores, thus resulting in the last node or edge in the top- $k$  differing in some dialogues. The development set contained 88 of 181 dialogue turns with at least one relevant node in the KG, while the test set contained 82 of 180.

relevant. A user utterance from the dataset “*Right, do you have Marie’s phone number?*” has the edge (“*Marie Shaffer*” | phone | “*4719170374*”) labelled as relevant. Lastly, utterances requesting a *type* of entity have all objects of that type and their immediate neighborhoods annotated as relevant.

### Retrieval methods

Using the relevant subgraph annotations in the GraphWOZ dataset, we evaluate the performance of PCST-based retriever and top- $k$  retriever with the use two different relevance scoring methods. Thus, we evaluate 4 variant approaches to KG retrieval for dialogue:

- A top- $k$  triples retriever, ranking individual triples from the KG with scores from a sparse retriever (BM25). These triples are then verbalized and used as context to the generator LLM to generate the system response.
- A top- $k$  triples retriever, ranking individual triples from the KG with cosine similarity scores from a dense retriever (Sentence-BERT). The triples are used for generation as with the sparse retriever.
- A PCST-based retriever, constructing a subgraph using the scores from a sparse retriever (BM25) as the prizes assigned to nodes and edges as described in Section 3.2. The triples of the retrieved subgraph are then verbalized and used as context to the generator LLM to generate the system response.
- A PCST-based retriever, constructing a subgraph using the cosine similarity scores from a dense retriever (Sentence-BERT). The subgraph is used for generation as with the sparse retriever.

To evaluate the retrieval results of these approaches on the re-annotated GraphWOZ dataset, we use the standard IR metrics of Precision and Recall. Nodes and edges annotated as relevant are the target class. Because the assigned relevance scores may be equal or approximately equal, the top- $k$  nodes and edges may differ slightly in evaluation. This condition applies both in the triples-retriever and the PCST-based retriever. For this reason, we sample the results of each retrieval-scoring model pair 100 times and average the results.

### Results

In our experiments, we observed that the retrieval models using BM25 relevance scores performed better than those using the sentence-BERT model, as shown in Table 1. While the triples retriever and PCST-based retriever performed similarly in F1-score when using sentence-BERT, we observed a slightly higher precision score in the PCST-based retriever than the triples-retriever, albeit a lower recall. Qualitatively, the improved precision in the PCST-based retriever using BM25 appears to result from turns where the model returns relatively small subgraphs centered around a core entity under discussion. In tandem with this effect, the lowered recall may arise from relatively distant relevant nodes failing to be included in the subgraph centered around the core entity. We additionally evaluate the results for statistical significance using the Mann-Whitney  $U$  test and report a  $p \approx 0$ .

Comparatively, both retrieval models using relevance derived from sentence-BERT showed similar performance to each other. Both of these models underperformed the models using BM25 relevance. This effect again appeared to relate to the advantage of selecting one or two core relevant entities. As the verbalizations are comparatively short (in

Model	Dev		Test	
	Sum Rel.	Per Node Rel.	Sum Rel.	Per Node Rel.
Triple <sub>BM25</sub>	24.69	1.65	23.21	1.55
PCST <sub>BM25</sub>	20.91	1.40	20.74	1.35
Triple <sub>SBERT</sub>	3.54	0.24	3.52	0.23
PCST <sub>SBERT</sub>	3.41	0.21	3.15	0.22

Table 2: Retrieved relevance on the development and test set of GraphWOZ calculated with respect to the gold-standard turn responses. The Sum Relevance (Sum Rel.) is the total of the raw relevance scores in the retrieved nodes and edges, while the Relevance Per Node (Per Node Rel.) divides the sum relevance by how many nodes are returned.

contrast to an entire document that might be retrieved in other applications), the advantage of the neural model over heuristics may be lessened.

## 4.2 Relevance to Response

To further understand the relation of the retrieved graph elements to the dialogue system output, we can also derive the total prize and prize per node of the retrieved subgraphs with respect to the gold-standard response from the dataset. This analysis yields a second estimate of the retrieval quality without reference to the annotated labels. At the outset, we can expect that relevant nodes and edges will have a higher cosine similarity score to the system response if they are used in generating that response and are most likely directly mentioned. For example, the system response to a user’s question such as “What meetings will Mark Williams attend?” will be highly similar if not identical to the verbalized fact “Mark Williams is attending status update.” This likelihood of graph information being verbalized directly in the output provides an indirect signal as to what nodes and edges may help in generating it. However, this relevance is only scored with respect to *individual* graph elements (nodes and edges), but cannot fully capture paths or larger structures. Consequently, subgraph retrieval should reliably include the most relevant elements, but also include some number of otherwise lower scored nodes. To investigate the impact of the inclusion of nodes on the total relevance, we analyze the relevance of the subgraphs with respect to the system response.

The results of this analysis are shown in Table 2. For the purposes of this analysis, the relevance scores are relative to the method of relevance scoring used, thus they should be com-

pared between the method of retrieval itself rather than across scoring methods (that is, the units of raw BM25 relevance scores are generally higher than those output by Sentence-BERT). This analysis showed similar numbers between the baseline and PCST-based retriever, with the baseline returning more noticeably higher relevance when using BM25 relevance scores. Both retrieval models using Sentence-BERT relevance scores returned more similar amounts of relevance, with the triples model returning slightly more. The BM25 function matches terms as a bag-of-words, thus the scoring function will miss relevant entities in turns with indirect references (e.g. pronouns). This limitation is somewhat mitigated in the PCST-based retriever using BM25, as a subgraph can be constructed that contains nodes assigned no relevance by the scoring model.

Our analysis here indicates that, while the PCST-based retriever returns a lower total relevance to response as indicated by the scoring model, the majority of the relevant graph elements (measured by the relevance scores) are also returned by this model. The relevant intuition of this decrease in relevance lies in the fact that the raw relevance scoring mechanism will not capture the relevance of larger structures such as paths, circuits, or other subgraph components above the level of triples. Thus, a retrieval based in subgraph structures will return nodes and edges which may nominally be less relevant according to the initial scoring function, but nevertheless assist in model outputs as part of a larger context. Notwithstanding this intuition, we should expect the most clearly relevant elements (more easily identified by a simple relevance scoring model) to be returned regardless.

When comparing the baseline to the PCST-based retriever, the difference in returned relevance relative to the scoring method was greater in the models using BM25. The greater disparity between the two models when using BM25 indicates less overlap in which elements are returned, suggesting that the difference in top- $k$  triples retrieval and PCST-based retrieval may be most apparent when using BM25 relevance scoring as a basis of comparison.

## 4.3 User Evaluation

To further compare the models in a practical setting following our analysis of the subgraph retrieval with the GraphWOZ dataset, we conducted a user study with simulated tasks. We recruited a small group of crowdsourced participants to evaluate the

## Virtual Agent

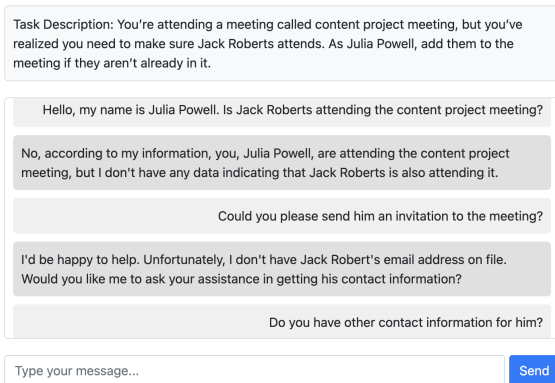


Figure 2: View of user interface, including task instructions presented to the user and the initial dialogue turns.

comparative performance of a task-oriented dialogue system using subgraph-retrieval versus a top- $k$  retrieval procedure and asked them to evaluate system performance according to three statements. In this section, we describe the details and results of this study.

### Setup

For the user study, we recruited an initial 39 native English-speaking participants from a crowdsourcing platform<sup>2</sup> to interact with the system, of which 31 sets of interactions<sup>3</sup> were retained for our analysis. The interactions of 8 participants were ultimately excluded for low quality of interaction as assessed by a simple attention check during the experiment to filter for spam. Each participant was tasked with interacting with the system for four separate dialogues with a subsequent evaluation of the system consisting of three statements for the user to score the system on. At the beginning of each interaction (shown in Figure 2), the user was assigned a task to be completed through dialogue with the system. For each dialogue, the users were required to complete at least 5 turns of dialogue with the system. The three statements presented to the users after each dialogue were:

1. “The system responded to me in a conversationally relevant way”
2. “The system gave me the information I asked for”

<sup>2</sup><https://clickworker.com>

<sup>3</sup>Due to a sampling error, one participant interacted with the non-verbalized baseline twice without interacting with the verbalized baseline, resulting in 32 total dialogues for the former and 30 for the latter; both PCST models have 31.

3. “The system successfully completed my task”

The first two statements asked the user to rate the system on a 1-5 point Likert scale, while the task completion statement was a binary answer of “yes” or “no.” The 1-5 point Likert scale values had the respective labels: 1) Never, 2) Mostly Not, 3) Sometimes, 4) Mostly, and 5) Always.

Each task given to the user was drawn from a pre-generated set of tasks, created by first generating a graph of data describing the calendar and people of a fictive organization. The number of people nodes to be generated was randomly drawn as between 25 and 35 and between 55 and 65 for event nodes. Every event was given a date between the day of and one week from the day of the study, and a time from and time between 8:00 and 5:00. The higher number of events compared to people in the graph avoided calendar sparsity and thereby increased the likelihood of users needing to engage in more complex dialogue turns such as negotiating availability in tasks involving scheduling new events. Along with the generated people and events, details such as locations (rooms drawn from a fixed set of names), emails, phone numbers were generated, and each person and group assigned with a group label (e.g. “Administration”). Each generated graph was then paired with a task description randomly selected from a set of templates containing delexicalized scenario descriptions and objectives (see Appendix A for the full list)

In this study, we evaluated 4 models of retrieval. The first two are baselines of top- $k$  retrieval with BM25 scoring (performing best according to the results of our analysis), one making use of our verbalization procedure and the other omitting it by instead simply concatenating labels of triples, as in other work cf. (Lanchantin et al., 2023). This comparison allows some estimation of the relative impact of this component of the pipeline. The remaining two models made use of PCST-based subgraph retrieval based respectively on BM25 and Sentence-BERT (Dense) relevance scoring as used in our previous analysis (described in Section 3.1). For all retrieval models, the output of the dialogue system is generated by an LLM using the retrieval output as part of a prompt detailing the agent’s role in helping users find information and schedule events and instructing the LLM to use the provided information to do so. For our LLM, we used the LLaMA 3.1 70B model (Grattafiori et al., 2024)

Model	Rel.	Info.	Completion
Baseline <sub>NV</sub>	4.0	3.38	0.56
Baseline <sub>V</sub>	4.0	3.13	0.6
PCST <sub>BM25</sub>	4.19	<b>3.61</b>	<b>0.65</b>
PCST <sub>SBERT</sub>	<b>4.26</b>	3.58	0.58

Table 3: Average participant scores assigned for the three statements for each model, corresponding to the statements assessing Relevance (Rel.), Informativeness (Info.) and Task Completion (Completion). The first two models are the baseline top- $k$  retrieval with non-verbalized (NV) and verbalized (V) facts, respectively, and the latter two are PCST subgraph retrieval using BM25 and sentence-BERT (SBERT) relevance scores. Relevance and Informativeness are scored on a **1-5** Likert scale, while Task Completion is a binary successful / unsuccessful dialogue.

deployed with Ollama<sup>4</sup>.

## Results

As shown in Table 3, the study results indicated that the dialogue model using PCST-based subgraph retrieval was slightly preferred by the users with regards to the three statements presented. The scores for informativeness and task completion were overall highest with the BM25 retrieval, and highest for relevance with the PCST retrieval using the Sentence-BERT retriever scores. The task completion rates were relatively similar amongst the models (slightly higher in the baseline using verbalization than the  $PCST_{SBERT}$  model), however somewhat the highest when using the  $PCST_{BM25}$  retriever. We again evaluated the results of each statement for statistical significance using the Mann-Whitney  $U$  test between the baseline and PCST-based retrievers, however no comparison of the baseline models to the PCST-retrieval models returned a value  $p$  falling below a threshold of  $p = 0.05$ .

Qualitatively, the retrieval methods returned relatively similar core items relevant to the exchange (typically the highest ranked element). However, the advantage of the PCST-based retrieval was particularly visible when retrieving multiple highly relevant entities somewhat removed from each other in the graph. In a circumstance such as reconciling availability between multiple people’s calendars, constraining the KG retrieval to return a connected subgraph containing the intermediate elements be-

tween the highest ranked nodes and edges encourages retrieval of useful contextual information. An example of this scenario is shown in Dialogue 1 in Appendix B. The retrieved subgraph centered on the Engineering group, which was the common point of intersection between the two mentioned people entities and their calendars (both of which are relevant). Throughout the dialogue, the user relied on information drawn from both their persona’s calendar and another member of Engineering.

## 5 Conclusion

In this paper, we have presented a method of KG retrieval for task-oriented dialogue. Approaching the retrieval component of RAG as a Prize-Collecting Steiner Tree problem, our model retrieves subgraphs that maximize dialogue turn relevance while minimizing retrieved subgraph size. The proposed subgraph retrieval method has the advantage of returning a variable number of nodes and edges that also have a structural relation in the graph. Subgraph-retrieval with the proposed method can vary according to dialogue context. Additionally, the subgraph can contain intermediate elements that receive no initial relevance score but are still relevant for the system response. Our experiments show that despite exhibiting slightly reduced relevance per node with respect to the gold-standard response, the subgraph retrieval method achieves a higher F1-score than a baseline top- $k$  retrieval model when using a heuristic relevance scoring model. Following this result, our evaluation of the system with user feedback also indicates that a task-oriented dialogue system using PCST-based subgraph retrieval may lead to increased perceptions of relevance and informativeness.

## 6 Limitations

This paper studies the performance of a subgraph-retrieval method in a task-oriented dialogue dataset. Our annotations assume that the relevance of nodes and edges in the graph is a binary value of  $\{relevant, irrelevant\}$ . However, in practice it may be advantageous to view relevance as a continuous value. The dataset we use is comparatively small, however task-oriented nature of the dialogue makes the annotation task somewhat simpler than for open-domain dialogue where potentially large classes of objects might be relevant. Given sufficient annotation, subgraph retrieval in other datasets with potentially larger sets of dialogues in different do-

<sup>4</sup><https://ollama.com/>



mains would aid in understanding the upsides and downsides of this approach. Additionally, our user study consists of a relatively small sampling of participants, which could be expanded upon in further work and with further questions, conditions, or LLMs. As noted, we did not find a statistically significant result when comparing the user scores with the Mann-Whitney  $U$  Test, thus further study with additional users would be warranted. Lastly, with respect to the verbalization procedure, the use of templates is naturally difficult to scale, therefore approaches using data-to-text techniques might assist in generalizing our approach to retrieval in knowledge graphs with different structures

## References

- Debanjan Chaudhuri, Md Rashad Al Hasan Rony, and Jens Lehmann. 2021. Grounding dialogue systems via knowledge graph aware decoding with pre-trained transformers. In *European Semantic Web Conference (ESWC)*, pages 323–339.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *arXiv preprint arXiv:2402.10612*.
- 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.
- 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.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Huang He, Hua Lu, Siqi Bao, Fan Wang, Hua Wu, Zheng-Yu Niu, and Haifeng Wang. 2023. [Learning to select external knowledge with multi-scale negative sampling](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1–7.
- Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*.
- Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. 2014. A fast, adaptive variant of the goemans-williamson scheme for the prize-collecting steiner tree problem. In *Workshop of the 11th DIMACS Implementation Challenge*, volume 2. Workshop of the 11th DIMACS Implementation Challenge.
- Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 105–113.
- Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2022. Knowledge-consistent dialogue generation with knowledge graphs. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.
- Jack Lanchantin, Sainbayar Sukhbaatar, Gabriel Synnaeve, Yuxuan Sun, Kavya Srinet, and Arthur Szlam. 2023. [A data source for reasoning embodied agents](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8438–8446.
- 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.
- Yu Li, Baolin Peng, Yelong Shen, Yi Mao, Lars Liden, Zhou Yu, and Jianfeng Gao. 2022. Knowledge-grounded dialogue generation with a unified knowledge representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 206–218.
- Jinyoung Park, Minseok Joo, Joo-Kyung Kim, and Hyunwoo Kim. 2024. Generative subgraph retrieval for knowledge graph-grounded dialog generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21167–21182.
- Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102.
- Liang Qiu, Yizhou Zhao, Yuan Liang, Pan Lu, Weiyan Shi, Zhou Yu, and Song-chun Zhu. 2022. Towards socially intelligent agents with mental state transition and human value. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 146–158.
- 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.

- 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 H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.
- 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.
- David Thulke, Nico Daheim, Christian Dugast, and Hermann Ney. 2021. [Efficient retrieval augmented generation from unstructured knowledge for task-oriented dialog](#). *arXiv:2102.04643*.
- Nicholas Thomas Walker, Stefan Ultes, and Pierre Lison. 2022. [Graphwoz: Dialogue management with conversational knowledge graphs](#). In *Proceedings of the 13th International Workshop on Spoken Dialogue Systems Technology*.
- Nicholas Thomas Walker, Stefan Ultes, and Pierre Lison. 2023. A retrieval-augmented neural response generation using logical reasoning and relevance scoring. In *Proceedings of the 27th Workshop on the Semantics and Pragmatics of Dialogue, Full Papers*.
- Sixing Wu, Jiong Yu, Jiahao Chen, and Wei Zhou. 2024. Generative commonsense knowledge subgraph retrieval for open-domain dialogue response generation. *Neural Networks*, 180:106666.
- Yike Wu, Nan Hu, Guilin Qi, Sheng Bi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-rewrite-answer: A KG-to-text enhanced LLMs framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*.
- Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784.
- Rui Zhu, Bo Liu, Qiuyu Tian, Ruwen Zhang, Shengxiang Zhang, Yanna Hu, and Jiuxin Cao. 2025. Knowledge graph based question-answering model with subgraph retrieval optimization. *Computers & Operations Research*, page 106995.

## A Task Templates

You're a visitor at a new office named `_NewPerson_`. You're supposed to meet with someone named `_Person_`. Find out if they're currently available and set up a meeting with them as soon as they are available.

You're a new employee named `_NewPerson_`. You're supposed to meet with someone named `_Person_`. Find out if they are available at `_Time_` tomorrow and set up a meeting with them at or as near to this time as possible.

You're visiting a research institute and want to find `_Person_` today. As `_NewPerson_`, find out if they are in a meeting, and if they are available set up a time to meet with them.

You've been invited as `_NewPerson_` to talk to members of a local research group in `_Group_`. Find out what events this group has tomorrow and who the organizers are. Set up a meeting with each of them that day.

You're attending a meeting called `_CalendarEvent_`, but you've realized you need to make sure `_Person_` attends. As `_NewPerson_`, add them to the meeting if they aren't already in it.

You're attending a meeting called `_CalendarEvent_`. As `_NewPerson_`, find out when the event starts, how long it lasts, where it will be, and who is organizing it.

You're attending a meeting called `_CalendarEvent_`. As `_NewPerson_`, find out who is organizing the event and what other events are on their calendar.

You're attending a meeting called `_CalendarEvent_`. As `_NewPerson_`, find out where this meeting will be and what other events will be in that location that day.

Your name is `_NewPerson_`, you want to meet with a member of the `_Group_` group. Find someone from this group who is currently available and set up a meeting with them as soon as possible in an available meeting room.

Your name is `_NewPerson_`, you want to meet with a member of the `_Group_` group. Find someone from this group who is currently available and set up a meeting with them as soon as possible in an available meeting room.

Your name is `_NewPerson_`, you're supposed to attend a meeting called `_CalendarEvent_`. Find out when it is and where it will be held. Then, find out if you have any meetings after that meeting.

You are `_NewPerson_` and have been invited to a meeting called `_CalendarEvent_`. Find out when and where this meeting will take place.

You are `_NewPerson_`, and you want to arrange an hour long meeting called `_NewEvent_`. Make an hour long meeting in an available conference room, and invite `_Person_` and `_Person_`.

Your name is `_NewPerson_`, and you'd like to talk to `_Person_`. Find out where their office is and what floor of the building it's on.

You are `_NewPerson_`. Make a meeting with `_Person_` in an available meeting room at `_Time_` tomorrow, or at their nearest availability.

You are `_NewPerson_`. Make a meeting with `_Person_` in their office at `_Time_` tomorrow, or at their nearest availability.

Your name is `_NewPerson_`, you want to get in contact with a member of the `_Group_` group. Find a member of this group and get their phone number and email address.

Your name is `_NewPerson_`, you want to get in contact with the organizer of `_Event_`. Find out their phone number, email address and what group they work in.

You are `_NewPerson_`. You're looking for the contact information of `_Person_`.

Find out what group they work in and get their email address.

You are `_NewPerson_`. Make a meeting at `_Time_` tomorrow with two members of the `_Group_` group, or the nearest available time.

You are `_NewPerson_`, and you want to arrange a meeting called `_NewEvent_`. Make a half hour long meeting in an available conference room, and invite `_Person_` and `_Person_`.

You are `_NewPerson_`, and you want to arrange a meeting called `_NewEvent_`. Make a half hour long meeting in an available conference room, and invite `_Person_` and at least one member of the `_Group_` group.

You are `_NewPerson_` and need to find the time and date of a meeting called `_Event_`. Find out when what day and time it is and what group is holding it.

You are `_NewPerson_`. Find out when the meeting called `_CalendarEvent_` is, and make a half hour long meeting with the same attendees immediately after it.

You are `_NewPerson_`. Find out who is organizing `_Event_`, and find out the room number for their office and what group they work in.

You are `_NewPerson_`. For tomorrow, arrange a meeting with a member of the `_Group_` group and a member of the `_Group_` group.

Your name is `_NewPerson_`, and you need to join a meeting called `_Event_`. Ask the robot to add you to the meeting, then make a new meeting afterwards with the organizer of that event and one other attendee.

Your name is `_NewPerson_`, and you need to make a meeting called `_NewEvent_` after your meeting `_CalendarEvent_`. Create the new meeting the same attendees of that event, excluding the organizer.

As `_NewPerson_`, find out what events you are organizing, if any. If you are organizing any events today or tomorrow, cancel them. Then, set up a meeting called `_NewEvent_` with one member from the `_Group_` group.

As `_NewPerson_`, find out what events you are attending, if any. Set up a meeting called `_NewEvent_` with the organizer of one of these events at `_Time_`, or with anyone else if you have no events.

Your name is `_NewPerson_`, and you need to make sure that `_Person_` attends a meeting of yours. Find out what events are on your calendar, and add them to one of them. If there are no events, create a meeting with the other person called `_NewEvent_`.

You are `_NewPerson_`, and you need to postpone your meetings from today to tomorrow. Find out what events you have today, and move them to tomorrow at the same time slots, or the nearest possible time slots.

As `_NewPerson_`, cancel any morning meetings on your calendar today, and cancel any afternoon meetings you have tomorrow.

As `_NewPerson_`, find a meeting on your calendar and move it to one day later at the same time, then add `_Person_` if they are available at that time.

You are a new employee named `_NewPerson_`. Find out what your new office is, then set up an introductory meeting with `_Person_` in your office tomorrow morning.

You are `_NewPerson_`, and you want to invite `_Person_` and two members of the `_Group_` group to a meeting. Arrange a meeting called `_Event_` tomorrow morning with these three people.

You are `_NewPerson_`, and you need to set up a two hour long meeting called `_NewEvent_` tomorrow morning. Invite three available members from the `_Group_` and `_Group_` groups. If you have any conflicting events on your calendar, cancel them.

## B Dialogue Examples

Here, we provide some example dialogues taken from the user experiment, both samples taken from the same user interacting with the system. The first of these was generated by the LLM model using  $PCST_{BM25}$  retrieval, the second by the Baseline retriever using verbalized facts: Line-breaks in the verbalized subgraphs are converted to spaces for these transcripts.

### Dialogue 1: PCST-retriever (BM25 Scoring)

**1 USER:** hello I am cameron harvey, I am here to meet Naomi Burton, can you help me set up a meeting with them?  
**2 SYSTEM:** I'd be happy to assist you, Cameron. As Naomi Burton is also a member of the Engineering group, I can check their availability. Would you like to schedule a meeting at your earliest convenience, perhaps after your content status update or communities seminar?  
**SUBGRAPH:** (*Cameron Harvey is attending content status update. Cameron Harvey is attending communities seminar. Cameron Harvey is a member of the Engineering group. Cameron Harvey's email address is cameron.harvey@company.email. Naomi Burton is a member of the Engineering group.*)  
**3 USER:** yes schedule a meeting with their earliest availability.  
**4 SYSTEM:** I've checked Naomi Burton's calendar and their earliest availability is actually right now, as they are the organizer of meeting 191e401bb8 which has just started. Shall I let them know you're here to meet with them?  
**SUBGRAPH:** (*Cameron Harvey is attending communities project meeting. Naomi Burton is the organizer of 191e401bb8. Cameron Harvey is a member of the Engineering group. Naomi Burton is a member of the Engineering group.*)  
**5 USER:** yes, where is there location so I can start going there now  
**6 SYSTEM:** Content status update, which Naomi Burton is currently attending, will be held in Meeting Room Beta. You may head to Meeting Room Beta to meet with them.  
**SUBGRAPH:** (*Naomi Burton is attending content status update. content status update will be held in Meeting Room Beta. Naomi Burton is the organizer of 191e401bb8. Naomi Burton is the organizer of a562641218.*)  
**7 USER:** can you give me a map to meeting room beta or tell me where to find a map?  
**8 SYSTEM:** I'm afraid I don't have the ability to provide physical maps. However, there are digital display screens located at each floor's elevator lobby that show the building layout and meeting room locations, including Meeting Room Beta. You can refer to those for directions.  
**SUBGRAPH:** (*applications workshop will be held in Meeting Room Beta. technologies status update will be held in Meeting Room Beta. mindshare conference will be held in Meeting Room Delta. Mindshare conference is on 2025-04-12. schemas status update will be held in Meeting Room Beta. Schemas status update is on 2025-04-12. mindshare conference will be held in Meeting Room Beta. )*  
**9 USER:** you did a great job, thank you so much for your help  
**10 SYSTEM:** You're welcome! If you need any more assistance or have any other questions, feel free to ask. Have a great day and I hope you find Meeting Room Beta easily.  
**SUBGRAPH:** (*Jessica Fisher is attending info-mediaries team meeting. info-mediaries team meeting will be held in Meeting Room Zeta. Jessica Fisher is attending convergence seminar. convergence seminar will be held in Meeting Room Beta. applications team meeting will be held in Meeting Room Beta. schemas status update will be held in Meeting Room Beta. interfaces project meeting will be held in Meeting Room Beta.*)

**Scores:** Relevance: **5**, Informativeness: **5**, Task Completion: **1**

## Dialogue 2: Triples-retriever (verbalized facts)

**1 USER:** I am William Jackson. I need to attend a meeting called bandwidth status update. Can you tell me where this meeting will be and what other events will be in that location that day?

**2 SYSTEM:** Hello William Jackson, the Bandwidth Status Update meeting (event ID: 7464667393) does not have a specified location. However, I can inform you that the Communities Seminar and Initiatives Follow Up meetings will be held in Meeting Room Delta on the same day as your scheduled event.

*SUBGRAPH: (58ab600be7 is an event named bandwidth project meeting 7464667393 is an event named bandwidth status update communities seminar will be held in Meeting Room Delta. initiatives follow up will be held in Meeting Room Delta. info-mediaries workshop will be held in Meeting Room Omega.)*

**3 USER:** What other meetings are that day in any location?

**4 SYSTEM:** In addition to the Communities Seminar and Initiatives Follow Up meetings, there will also be a Deliverables Discussion meeting on the same day. However, I do not have information about the location of this additional meeting.

*SUBGRAPH: (initiatives follow up will be held in Meeting Room Delta. deliverables discussion will be held in Meeting Room Delta. communities seminar will be held in Meeting Room Delta. bandwidth status update will be held in Meeting Room Delta. initiatives team meeting will be held in Meeting Room Delta.)*

**5 USER:** who is leading the bandwidth status update meeting?"

**6 SYSTEM:** I don't have information on who is leading the Bandwidth Status Update meeting (event ID: 7464667393).

*SUBGRAPH: (58ab600be7 is an event named bandwidth project meeting 7464667393 is an event named bandwidth status update 602168ab8d is an event named communities project meeting 09b5688676 is an event named initiatives team meeting bc0f0af667 is an event named communities seminar)*

**7 USER:** do you have any other information about this meeting?

**8 SYSTEM:** No, I don't have any additional information about the Bandwidth Status Update meeting beyond its event ID (7464667393) and name.

*SUBGRAPH: (58ab600be7 is an event named bandwidth project meeting 7464667393 is an event named bandwidth status update 09b5688676 is an event named initiatives team meeting 0a9ac3eebf is an event named networks team meeting a19b168ee2 is an event named synergies conference)*

**9 USER:** what do you suggest I do to find out where the meeting is?

**10 SYSTEM:** I don't have information on the location of the Bandwidth Status Update meeting. However, I can tell you that the bandwidth project meeting (event ID: 58ab600be7) does not seem to be related to the Bandwidth Status Update meeting. You may want to try contacting the meeting organizer or checking other sources for more information on the location of the Bandwidth Status Update meeting.

*SUBGRAPH: (58ab600be7 is an event named bandwidth project meeting models project meeting will be held in Meeting Room Beta. initiatives follow up will be held in Meeting Room Delta. info-mediaries workshop will be held in Meeting Room Omega. vortals seminar will be held in Meeting Room Omega.)*

**Scores:** Relevance: 4, Informativeness: 2, Task Completion: 0