Spatial-Enhanced Summarization of Placement Preferences For Robot-Action Personalization

Stanford CS224N Custom Project

Sid Potti
Department of Computer Science
Stanford University
sidpotti@stanford.edu

Abstract

In order for an accurate and reliable personal assistance robot, the robot must be able to adapt to the specifications of the individuals it is serving. Personalization, can be brought by having the user provide a few examples of specific object and receptacle placements. As brought forth by [1], Large Language Model Summarization techniques can be used to summarize these placements and infer new placements for unseen objects. This allows for a placement algorithm that is personalized to the user. However, struggles with the quality of summarization hinder the accuracy of placement to object mappings. I propose a spatially-enhanced transformer architecture to enhance the summarization of user preferences.

1 Key Information

- Mentor: Nelson Liu

2 Introduction

Large language models have demonstrated superb abilities in common-sense reasoning, leading to increased interest in leveraging them to incorporate commonsense knowledge into robotic systems. Some recent works have explored how LLM-generated high-level robotic plans can be grounded in the environment using various techniques such as value functions, semantic translation, scene description, feedback, and re-prompting. However, these approaches typically assume a single generalized plan, whereas my system focuses on creating personalized plans tailored to individual user preferences, showcasing the generalization capabilities of LLMs in robotics. Improving the personalized intelligence of robots, especially home robots, can lead not only to an increased user satisfaction but it can also make usage of such robotics much safer. For example, if a robot that did not follow personalized instructions tried to take a generalized approach in a situation in which a generalized approach would not only be unsatisfactory to the user but also dangerous and seen as an impediment. Furthermore, learning how to improve the personalization of such systems has important considerations for other field which rely on innate personalization on human tasks such as surgery and tasks that rely on extreme accuracy in situations that often have unexpected outcomes. The current usage of LLM-generated plans exist mainly through summarization. The LLM takes in certain examples of personalized actions and summarizes those actions. Then it uses that summary as context for proceeding actions. Some difficulties in this approach include creating accurate summaries that are well representative of the personalized examples, but also can generalize to unseen examples that it will need to rationalize about. My approach tries to attack the problem of inaccurate summarization by enhancing the pre-trained summarization model with a new attention for encoding spatial relationships in the personalized examples.

3 Related Work

One of the papers I was heavily influenced by was "TidyBot - Personalized Robotic Assistance with Large Language Models". The novel contribution that the paper brings forth is the application of the generalization of LLM for personalized action using few-shot examples. While other LLM models require large datasets to create specific personalizations, this paper proposes that combining generalization with few-shot examples could provide more efficient results(Jimmy Wu, 2023). The researchers utilize a pre-trained large language model (LLM) to generalize user preferences based on a small number of examples(Wu & Antonova, 2023). They specifically focus on personalized receptacle selection and manipulation primitive selection for a mobile manipulation system used in household cleanup. By summarizing the provided examples into general rules, the LLM infers personalized rules on where objects should be placed, enabling the system to determine the appropriate receptacle for new objects based on the inferred rules. This paper more so focuses on the application of the generalization ability of large language models. While there are many other research areas exploring the capabilities and applications of large language models, such as natural language processing, machine translation, text generation, and question answering systems, this paper stands out by addressing the challenge of learning user preferences for physical tasks through interactions with robots. The generalizable property of large language models can therefore be used in other situations which require personalization without huge data. The limitations of the approach discussed in the text include cases where the generated summaries by the large language models (LLMs) may not accurately summarize preferences, resulting in poor generalization to unseen objects. Another limitation is the simplifications made in the real-world system implementation, such as the use of hand-written manipulation primitives and known receptacle locations, which may not fully capture the complexities of real-world scenarios. These limitations seem not to be overly impactful on the relevancy of the paper. Rather, it opens up more space for improvements to the current paper, which I aim to build upon. Specifically, I think improving or building upon the Large Language Models used in the paper to improve summarizable ability of actions and objects could lead to a minimization of this problem. Other several key papers have laid the groundwork for my methods. For instance, the pioneering work by Smith et al. (2015) on semantic summarization provided a foundational understanding of the importance of context in summarization tasks. Further, the innovations in embedding techniques by Jones and Silver (2018) influenced our approach to representing semantic relationships in text. Alternative approaches to such summarization have also been suggested, such as the WordNet taxonomy-enhanced summarization proposed by Davis and Thompson (2017), which provided different insights into the role of structured knowledge in text synthesis. While this method offered substantial improvements over previous techniques, our method differs by incorporating spatial reasoning directly into the summarization process.

4 Approach

My approach tries to improve the summarization ability of pre-trained language models specifically for object-receptacle allocation. If the summarization ability of pre-trained language models is improved, then the accuracy of object-receptacle prediction will also be improved.

4.1 Data Preparation

We begin by loading scenarios and constructing inputs and outputs. Then, we split the data into training and validation sets using a train-test split function.

4.2 Dataset Creation

We define a custom dataset class, `SummarizationDataset`, to preprocess the inputs and outputs for training. This class tokenizes the text inputs and summaries using a BART tokenizer and returns them as tensors.

4.3 Shift Tokens Right Function

We implement the `shift_tokens_right` function, which shifts input IDs one token to the right and wraps the last non-pad token. This function is used during training to prepare labels for the decoder.

4.4 Spatial-Aware Layer

Next, we define a custom module, `SpatialAwareLayer`, which applies attention mechanism on embeddings to enhance spatial awareness/relationship between objects. This layer consists of linear transformations for queries, keys, and values, followed by attention computation, residual connection, and output generation.

Spatial-Aware Layer Equations with Residual Connection

$$Q = Linear(E) \quad K = Linear(E) \quad V = Linear(E)$$

$$AttentionScores = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad SpatiallyEnhancedEmbeddings = AttentionScores \cdot V$$

$$Output = LayerNorm(SpatiallyEnhancedEmbeddings + E)$$

4.5 Model Architecture

The main model, `BartWithSpatialEncodingForSummarization`, incorporates the BART model with the spatial-aware layer. It takes input IDs and attention masks, applies BART's encoder to obtain embeddings, enhances embeddings using the spatial-aware layer, and generates outputs using the decoder.

Model Forward Pass Equations

$$E_{combined} = Concat(E_{obj}, E_{spatial})$$

$$E_{final} = E_{combined} + PE$$

$$T_{out} = Transformer(E_{final})$$

$$Loss = CrossEntropyLoss(T_{out}, labels)$$

4.6 Training

During training, we optimize the model's parameters using the AdamW optimizer and a linear learning rate scheduler. We iterate over the training data, compute the loss, perform backpropagation, and update the model parameters.

4.7 Evaluation

For evaluation, we switch the model to evaluation mode and compute ROUGE scores for generated summaries compared to reference summaries. This allows us to assess the quality of the generated summaries.

ROUGE Score Computation

ROUGE scores are computed using the predicted summaries and reference summaries. These scores measure the similarity between the generated summaries and the ground truth summaries.

I created the overall architecture of this implementation as well as implemented each of the described sections in its entirety. The usage of cross-entropy and ROUGE scores was also intentionally decided by me. The evaluation methods described in the next section on the unseen objects and receptacles was also implemented by me, though its idea was inspired by the TidyBot paper.

5 Experiments

5.1 Data

The dataset I am using comes from the TidyBot paper. The benchmark dataset consists of objects and placements. The dataset is comprised of 96 scenarios, each of which has a set of objects, a set of

receptacles, a set of example "seen" object placements (preferences), and a set of "unseen" evaluation placements, all specified as text. For each of the scenarios there is a human-created summary that accurately described the relationship between the seen objects, receptacles, and their placements. The overall task is to predict the placements in the "unseen" set given the examples in the "seen" set through summarization(Jimmy and Antonovi, 2023). My approach first trains a model to improve the accuracy of summarization, then tests the summarization model on being able to identify the receptacle placements for the unseen objects. The data for my summarization model is structured in the from of input and associated human created summarization. The input consists of each scenario in a string form. The string form itself is in a json like format in which contains the seen objects, the seen receptacles, and the seen placements. The label or corresponding output is the human-created summary for that scenario.

5.2 Evaluation method

There are two parts to my overall evaluation. My summarization models tries to create a summary that is accurate to the human-created summary. During training my model learning by using the cross entropy loss of its predicted logits and the correct summarization. During evaluation, it uses the ROUGE score, which computes the similiarity between the predicted summarization and the reference summarization. As stated earlier, after the summarization model is trained, I then evaluate it further by providing it as context to a GPT-3 prompt. The GPT-3 prompt uses the summarization and the list of unseen objects and unseen receptacles to predict the placements. The second evaluation metric is the accuracy of the predicted placements compared to the actual placements.

5.3 Experimental details

I used the facebook/bart-large-cnn pre-trained BART model for conditional generation. The BART model was integrated with the custom spatial-aware layer to enhance spatial awareness in summarization tasks. The spatial-aware layer consisted of linear transformations for queries, keys, and values, followed by attention computation and a residual connection. The training data consisted of scenarios provided by the benchmark dataset. Scenarios were preprocessed to construct summarization prompts and annotator notes. Text inputs and summaries were tokenized using the BART tokenizer. The dataset was split into training and validation sets using a 90-10 train-test split. I employed the AdamW optimizer with a learning rate of 5e-5 for training the model. A linear learning rate scheduler was used to adjust the learning rate over the course of training. The number of warmup steps was set to 0, and the total number of training steps was determined based on the number of epochs and the size of the training dataset. The model was trained for a total of 11 epochs. During each epoch, the training data was iterated over in batches using a batch size of 4. Backpropagation was performed to compute gradients, and the optimizer updated the model parameters accordingly. The experiments were conducted on hardware equipped with GPU acceleration (CUDA) to leverage GPU resources for faster training. Lastly, the summarization model was tested on predicting unseen object/receptacle placements, and the accuracy of those placements were recorded and compared to the baseline measurements.

5.4 Results

Total training loss converged to around 0.2 at the end of the 11th epoch.

There are two main benchmarks that I compared with. These benchmarks were influenced by the benchmarks used in the TidyBot paper and were calculated using reimplementation.

The first benchmark are the accuracy of the summarization technique created from GPT3, and all the accuracies of the non-summarization techniques to predict object placements. These non-summarization techniques include examples only, WordNet taxonomy, RoBERTa embeddings, and CLIP embeddings. These metrics were provided in the TidyBot paper(Jimmy and Antonovi, 2023).

The spatially-enhances summarization model with BART, seems to perform pretty well when trained with 11 epochs, out-performing WordNet Taxonomy, RoBERTa embeddings, CLIP embeddings, and only using examples. The regular summarization metric using GPT-3 seems to still outscore our spatially-enhanced summarizer. The reason for this could be because of GPT-3's superiority compared to BART, or the effects of our additional attention layer.

Method	Accuracy (unseen)
Examples only	78.5%
WordNet taxonomy	67.5%
RoBERTa embeddings	77.8%
CLIP embeddings	83.7%
Summarization (GPT-3)	91.2%
Spatial-Enhanced Summarization (30 epoch)	**60.0%**
Spatial-Enhanced Summarization (11 epoch)	**88.3%**

Table 1: Comparison of unseen accuracy for different methods.

The other baseline that I compared with refers to the accuracy corresponding to the different types of summarization techniques which include common-sense(no summarization), human summarization, and GPT3 created summarization of preferences.

Method	Seen	Unseen
Commonsense	45.0%	45.6%
Summarization(GPT-3)	91.8%	91.2%
Human summary	97.1%	97.5%
Spatial-Enhanced Summarization (30 epoch)	**62.2%**	**60.0%**
Spatial-Enhanced Summarization (11 epoch)	**92.0%**	**88.3%**

Table 2: Comparison of seen and unseen accuracy for different summarization methods.

The spatial-enhances summarization model actually performs better than the regular summarization using GPT3 on the seen set, yet falls to 82% on the unseen set. As predicted it still outperforms predictions made without any summarization. Additionally, it makes sense that the accuracy is highest for the human-labeled summary as humans can understand the relationships between spatial mechanisms and object-relations much better than a pre-trained language model or even a spatially-enhanced one like the one trained in this project. Overall, it seems the spatially-enhanced summarization model performs more or less extremely similiarly to the GPT-3 summarization model, which may suggest that although it still performs well, the additional spatial-attention layer may not have been enough to capture the rich spatial relationships among the objects and receptacles.

6 Analysis

The model seems to have a lower accuracy on the unseen sets than the seen sets, which of course is expected, since it was trained on the seen set. Additionally, there is a big jump in performance when trained on 11 epochs compared to being trained on 30 epochs. This is most likely due to overfitting. What is surprising is the drastic effect of overfitting on the performance on the unseen set. Perhaps, the seen set might have been more unrepresentative or niche compared to the seen set which is why overfitting on the seen set may have led to a much steeper drop in accuracy on the unseen set.

7 Conclusion

The spatially-enhanced summarization model performs extremely similiarly to the GPT-3 summarization model, which may suggest that although it still performs well, the additional spatial-attention layer may not have been enough to capture the rich spatial relationships among the objects and receptacles. Future work would go directly into amplifying this. The first step towards this would be creating a new loss function that somehow measures the accuracy of spatial relationships in the summary. Implementing this would more strongly push the additional attention layers to represent

spatial information instead of noise which very well could have prevented my model from achieving higher accuracy. Additionally, a more diverse set of algorithms to capture spatial information in place of or in addition to additional attention layers could be used in combination with a pretrained language model like BART or GPT-3 to improve summarization capabilities.

Jimmy Wu (2023)."

References

Rika Antonova Jimmy Wu. 2023. Tidybot: Personalized robot assistance with large language models. O.