Enhancing Knowledge Selection via Multi-level Document Semantic Graph

Haoran Zhang, Yongmei Tan *
School of Artifical Intelligence, Beijing University of Posts and Telecommunications, China
{zhanghaoran, ymtan}@bupt.edu.cn

Abstract

Knowledge selection is a crucial sub-task of Document Grounded Dialogue System. Existing methods view knowledge selection as a sentence matching or classification. However, those methods can't capture the semantic relationships within complex document. We propose a flexible method that can construct multi-level document semantic graph from the grounding document automatically and store semantic relationships within the documents effectively. Besides, we also devise an auxiliary task to leverage the graph more efficiently and can help the optimization of knowledge selection task. We conduct extensive experiments on public datasets: WoW(Dinan et al., 2018) and Holl-E(Moghe et al., 2018). And we achieves state-of-the-art result on WoW. Our code has been released at https://github.com/ddf62/multi-level-semantic-document-graph.

Keywords: document grounded dialogue, knowledge selection, document semantic graph

1. Introduction

For a long time, developing a dialogue system that can communicate with humans naturally has received extensive attention from researchers. Natural language generation models (Brown et al., 2020; Ouyang et al., 2022) have the ability to generate fluent responses in open-domain dialogues without access to external knowledge. However, those models tend to generate generic, repetitive, or hallucinate content (Holtzman et al., 2019; Maynez et al., 2020), resulting in a boring response. To solve such a problem, knowledge grounded dialogue is proposed. Knowledge grounded dialogue refers to the process of generating informative and contextually relevant responses based on dialogue context and external knowledge (Dinan et al., 2018; Huang et al., 2020; Shuster et al., 2020). In this paper, we are interested in unstructured documents as external knowledge.

Document grounded dialogue (DGD) is a kind of dialogue system in which the content of chat is around the grounding document. As shown in Figure 1, the topic of the conversation is "forgetting" and the robot needs to utilize the grounding document's content to generate the response "...unable to call up the older memories...". DGD can be divided into two sub-tasks: knowledge selection and response generation (Ma et al., 2020). Knowledge selection aims to select the most related information from the background document based on dialogue contexts. It is a crucial sub-task for document grounded dialogue because it can determine the content of the generated response. Most existing research treats knowledge selection as sentence extraction (Wu et al., 2021; Daheim et al., 2021) or ranking task (Li et al., 2022a; Huang et al., 2021). They view documents as isolated sentences, whereas a document is not a bag of sentences. When facing complex documents, selecting true knowledge often requires a comprehensive understanding of the semantic relationships between different sentences. So it would be a challenge for those methods that break the origin structure of the document.

In this paper, we employ multi-level document semantic graphs to solve the aforementioned issue. Based on graph structure, by encoding sentences into graph nodes and connecting related nodes with edges, we can easily model the relationship between sentences. A line of knowledge graphs(KGs) (Vrandečić and Krötzsch, 2014; Speer et al., 2017) have been proposed. Many methods, e.g. Iyer et al. (2021), have also been proposed to utilize those KGs effectively. However, since the background documents may be product manuals or literary works, the knowledge contained in those documents would not be included in general knowledge bases. To fill the gap between background documents and general knowledge bases, we introduce a method to automatically construct graphs from documents that can capture the multi-level information of the grounding document. First, we construct multi-level document semantic graphs for each grounding document based on the results of coreference resolution and syntax analysis. Then we use a pretrained language model and a graph neural network to encode the graph. Finally, we pick up the most appropriate segment as the grounding knowledge based on the graphs. To make full use of the graph efficiently, we also devise an auxiliary task that can help with the optimization of the main task.

* Corresponding author

LREC-COLING 2024, pages 5996–6006

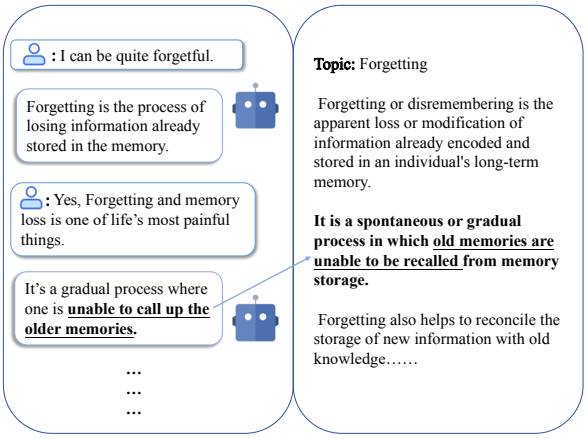

Figure 1: An example of the Document Grounded Dialogue. The left is the dialogue context and the right is the grounding document.

Our contributions in this paper are three-fold:

- We devise a multi-level graph that can be automatically constructed from grounding document.
- We devise an auxiliary task that can joined optimized with the knowledge selection task.
- Empirical results shows that our methods achieves state-of-the-art on WoW dataset for knowledge selection task.

2. Related Work

2.1. Knowledge Grounded Dialogue Systems

In recent years, how to introduce external knowledge into a dialogue system has received massive attention from researchers. For structured knowledge, like tables and knowledge triplets, Zhu et al. (2021); Pal et al. (2022) linearize the tables by adding some special tokens into the sequence, such as row id tokens and column id tokens, and use a large language model to encode the sequence to enhance the chat over the tables. Zeng et al. (2022) concatenate knowledge triplets with dialogue contexts as model input and use a sequence-to-sequence model to generate responses directly. For unstructured knowledge, Dinan et al. (2018) first retrieve several related knowledge sentences from the knowledge base and then train two versions of Generative Transformer Memory Network: the end-to-end version and the two-stage version. Li et al. (2019) incorporate grounding documents into the process of encoding conversation history and use a two-stage decoder to generate the response directly. Zhao et al. (2020) add a knowledge selection module into a pretrained language model and train the model in an unsupervised setting without human annotation. Gao et al. (2022) design different prompts to utilize knowledge stored in the large language model's parameters and train the pretrained large language model to generate responses based on grounding documents and dialogue content in an end-to-end manner. This method is suitable for scenarios where a few pieces of correct knowledge fragment data are annotated.

2.2. Knowledge Selection in Dialogue

Knowledge selection plays an important role in knowledge grounded dialogue systems. Most previous works define knowledge selection as a matching and ranking task. Kim et al. (2019); Zhao et al. (2020) use a latent vector to sequentially track the state of used knowledge and model the prior and posterior distribution of knowledge to make the knowledge selection results more accurate and diverse. Based on this, Xu et al. (2023) propose a probabilistic model with dual latent variables: one discrete latent variable for knowledge selection and one continuous latent variable for response generation. They jointly optimize knowledge selection and response generation in an end-to-end framework. Ren et al. (2020) introduce a global-to-local knowledge election mechanism to enhance knowledge selection without any extra annotations or information. Gao et al. (2022); Sun et al. (2023) design different prompts and use prompt learning to generate grounding knowledge in a sequence-to-sequence version. Similar to our method, Li et al. (2022b); Xu et al. (2022) construct grounding documents into graphs and do knowledge selection over the knowledge graph. In comparison, our approach captures the relationships between words within documents in a multi-level manner while preserving the information in the document as much as possible.

3. Approach

3.1. Problem Statement

Let $U = \{u_1, u_2, ..., u_{|U|}\}$ denote the dialogue context consisting of $|U|$ utterances. $D = \{d_1, d_2, ..., d_{|D|}\}$ denotes a set of grounding documents, where $|D|$ is the number of documents. Each documents d_i have a document title, we view it as the topic of the document. Each documents $d_i = \{s_1, s_2, ..., s_{|d_i|}\}$ contains $|d_i|$ knowledge segments. s_i can be spans, sentences or other reasonable divisions of the grounding document. In addition, every knowledge segment s_i is composed by $|s_i|$ words, i.e. $s_i = \{w_1, w_2, ..., w_{|s_i|}\}$. The task is to pick up appropriate knowledge segments from

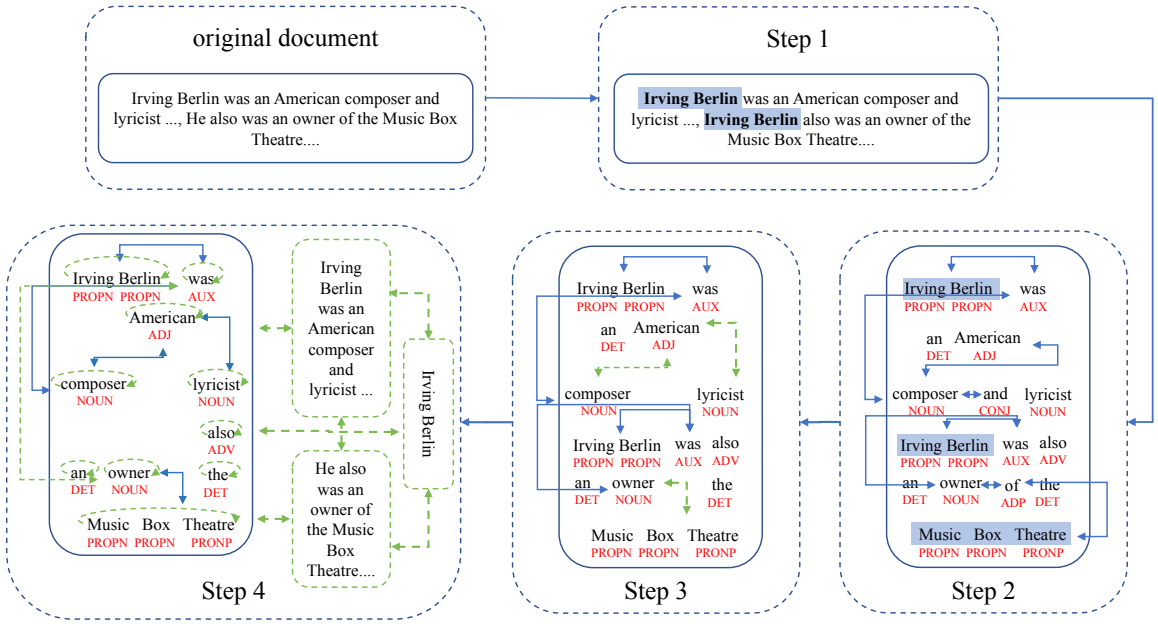

Figure 2: An example of the process of the graph construction. To make the description clearer, we delete some irrelevant edges and nodes.

the background documents D based on the dialogue context U.

3.2. Multi-level Document Semantic Graph Construction

In this part, we introduce a method to automatically convert grounding documents D into knowledge graphs G. An example of the general process is shown in the Figure 2 and the detailed algorithm is described below:

Step 1 We use **neuralcoref** [1] to get the coreference resolutions of each grounding document d_i and revert pronouns to their referents. For example, "**Irving Berlin** was an American composer and lyricist ..., **He** also was an owner of the Music Box Theatre on Broadway ....". In this document, "He" refers to "Irving Berlin", so we replace "He" with "Irving Berlin" to facilitate the fusion of word nodes. After this, we get a new set of documents $D' = \{d'_1, d'_2, ..., d'_{|D|}\}$.

Step 2 We input each document of d'_i into **spaCy** [2] to get the dependency parsing trees and Part-of-Speech(POS) labels. The dependency parsing tree can be viewed as the original semantic graph, so that there are $|D|$ original document sub-graphs $g_i = \{v_i, e_i\}$. v_i, e_i are nodes and edges respectively. At the same time, we remove all the punctuation in the documents. For every word, we merge adjacent words with the same part of speech into one semantic unit which can be viewed as the origin word nodes, such as "Irving Berlin" and "Music Box Theatre" in Figure 2. So that every knowledge segment is divided into several word nodes $s_i = \{w_1, w_2, ..., w_{|s_i|}\}$.

Step 3 There are many phrases made up of conjunctions or prepositions in the grounding documents, such as "Irving Berlin was an American **composer and lyricist** ..., Irving Berlin also was an **owner of the Music Box Theatre** on Broadway ....". For conjunctions, we treat the words joined by conjunctions with the same importance, such as "**composer**" and "**lyricist**", so we make the words connected by conjunctions share the same edges. For prepositions, we build a connection between word on both sides of preposition to reduce the graph complexity and make the information can transfer in the graph more efficient. Like the example above, we add an edge from "**onwer**" to "**Music Box Theatre**". At the end of this step, we also delete the conjunctions and prepositions word nodes (e.g., delete "and" and "of" nodes in the above examples).

Step 4 Finally, we add knowledge segment nodes and make them connected with all word nodes belonging to them. We also add a topic node for each documents and connect it to all knowledge segment nodes.

[1] https://github.com/huggingface/neuralcoref, MIT License

[2] https://spacy.io/

Topic node can be the topic or title of the document. In the original graph, different knowledge segment nodes are isolated. To make information can be spread between different knowledge segment nodes, we connect the knowledge segment nodes that are adjacent in the original document. Also we merge the word nodes with the same content in one document sub-graph to reduce the graph's redundancy. Besides those, to make the information can be maintained in it's own node, we add an edge for each nodes that pointing to itself. The changes of this step are shown in green part of Step 4 in Figure 2.

Finally, we can get the multi-level document semantic graph $G = \{g_1, g_2, ..., g_{|D|},\}$, $g_i = \{v_i, e_i\}$. Every v_i containing three levels nodes: topic nodes, knowledge segment nodes, word nodes.

$$v_i = \{tp_i, node_{s_1}, node_{s_2}, ..., node_{s_{|d_i|}}, node_{w_{1,1}}, node_{w_{1,2}}, ..., node_{w_{1,|s_1|}}, ..., node_{w_{|d_i|,|s_{|d_i|}|}}\}$$

tp_i is the topic node of i-th document sub-graph. $node_{s_i}$ is the i-th knowledge segment node in this sub-graph, $node_{w_{i,j}}$ is the j-th word node for i-th knowledge segment node.

3.3. Graph Node Initialization

Figure 3 is the general illustration of our knowledge selection model. Although we have the structure of the graph, to operate the calculation, we need to initialize the embedding of every node.

Word node, Topic node. We get node's embedding h_k through encoding the text of the node with the pretrained language model f_{LM}:

$$h_k = Pooling(f_{LM}(x_k)) \quad (1)$$

k can be the topic node tp or word node $node_w$, x_k is the text of word node. For pooling function, we use the mean pooling. $h_k \in \mathcal{R}^{dim}$, where dim is the dimension of the hidden states of pretrained language model.

Knowledge segment node. First, to make the embedding of the node can be dialogue context aware, we concatenate the previous p rounds of dialogue context with the content of each knowledge segment to contextualize the knowledge:

$$Text_{s_i} = [user], u_{n-p+1}, [agent], u_{n-p+2}, ..., [user], u_n, [know], s_i$$

$[user]$, $[agent]$, $[know]$ are special tokens representing the user utterances, agent responses and knowledge respectively. Then, we use $Text_{s_i}$ as the input of the pretrained language model f_{LM} to encode the node:

$$h_{s_i} = Pooling(f_{LM}(Text_{s_i})) \quad (2)$$

where $h_{s_i} \in \mathcal{R}^{dim}$. For pooling operation, we choose the first token $[CLS]$ as the output of the function.

So that we can get the embedding of the graph: $H = \{h_1, h_2, h_3, ..., h_{nn}\} \in \mathcal{R}^{nn \times dim}$, nn is the number of nodes.

3.4. Knowledge Selection

After the initialization of the graph, we use a Graph Neural Network (GNN) (Scarselli et al., 2008) to encode the whole graph.

$$H' = GNN(H) \quad (3)$$

where $H' \in \mathcal{R}^{nn \times gd}$, gd is the dimension of the GNN's hidden state.

To make the gradient better conduct to the pretrained language model layer, we add a residual structure (He et al., 2016) after the GNN:

$$\hat{H} = ReLU(W_1 H) + H' \quad (4)$$

$W_1 \in \mathcal{R}^{dim' \times dim}$ is learnable parameters. $\hat{H} \in \mathcal{R}^{nn \times dim'}$.

Because the aim of the task is to select the true knowledge segment, so we only use knowledge segment nodes to do final selection and mask other kinds of nodes. We use a fully connected layer to obtain the probability of whether the node is the appropriate knowledge segment:

$$p_i = \frac{exp(\hat{h_i} W_p + b_p)}{\sum_{j \in \hat{H'}} exp(\hat{h_j} W_p + b_p)} \quad (5)$$

$\hat{H'}$ is the set of knowledge segment nodes. $W_p \in \mathcal{R}^{dim'}$ and b_p are learnable parameters. Finally, we choose the knowledge segment with the highest probability as the true knowledge segment $\hat{s}$.

3.5. Auxiliary Task

There are a large amount of word nodes in the graph that don't be used in knowledge selection. However, after passing through the GNN, embedding of these nodes also contain certain semantic information related to the dialogue context and knowledge segments. Utilizing those nodes can help the model capture the relationship between dialogue context and candidate knowledge segments better. So we add an auxiliary binary classification task to utilize those word nodes: **word nodes selection**. The task is to make the model to predict the probability of whether the word node belongs

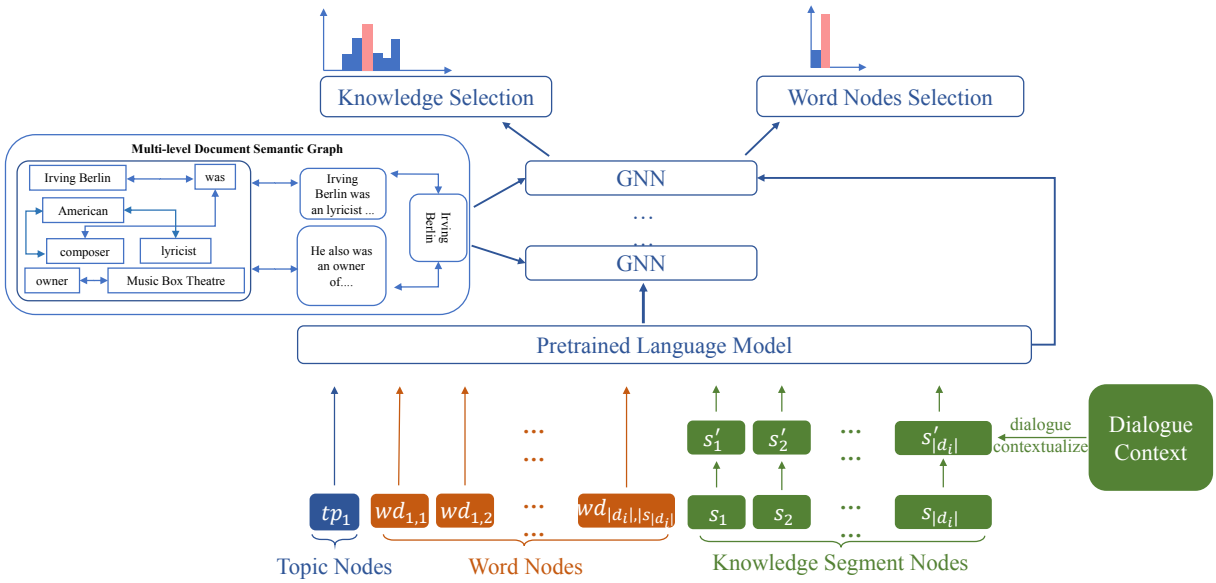

Figure 3: The architecture of the knowledge selection model. GAT refers to Graph Attention Network. Word Nodes Selection is the auxiliary task we add.

to the true knowledge segment node. We only consider the word nodes belong to the ground-truth document and the probability is shown in Eq.6:

$$p_{wd,i} = \frac{exp(\hat{h_{wd}} W_{a,i} + b_{a,i})}{\sum_{j\in\{0,1\}} exp(\hat{h_{wd}} W_{a,j} + b_{a,j})} \quad (6)$$
$$wd \in S'$$

S' is the set of word nodes that belong to the ground-truth document. $W_a \in \mathcal{R}^{dim' \times 2}$ and b_a are learnable parameters. $i \in \{0, 1\}$, 0 represents the node doesn't belong to the true knowledge segment node, 1 represents the node belongs to the true knowledge segment node. If there is an edge between the word node and true knowledge segment node, the golden label of the node is 1, otherwise 0. Moreover, we mask other word nodes that don't belong to the ground-truth document.

3.6. Training Objective

Eq.(7-8) is the loss function of knowledge selection and word nodes selection.

$$\mathcal{L}_{know} = -ylog(p) \quad (7)$$

$$\mathcal{L}_{word} = -\frac{1}{|S'|} \sum_{wd\in S'} log(\hat{p}_{wd,g}) \quad (8)$$

y, p are the ground-truth label and probabilities of the knowledge segment nodes respectively. $\hat{p}_{wd,g}$ is the probability corresponding to the golden label of wd-th word node.
Finally, the combined training objective is:

$$\mathcal{L} = \alpha * \mathcal{L}_{know} + (1 - \alpha) * \mathcal{L}_{word} \quad (9)$$

α is a hyper-parameters.

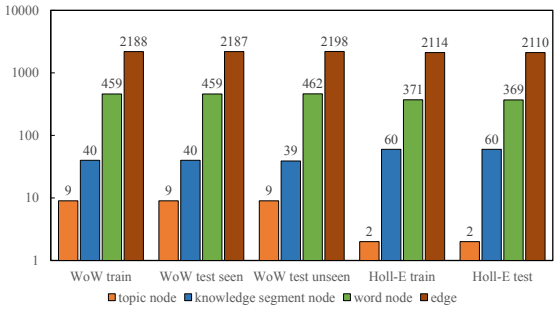

Figure 4: The average number of topic nodes, knowledge segment nodes, word nodes and edges in diifferent sets of WoW and Holl-E.

4. Experiments

4.1. Settings

4.1.1. Dataset

We validate our model on the public dataset: *Wizard of Wikipedia*(WoW) (Dinan et al., 2018) and Holl-E (Moghe et al., 2018).

WoW is an open domain dialogue dataset using Wikipedia documents as grounding knowledge documents. There are 18,430/1,948/1,933 dialogues in the training/validation/test set. Every dialogue contains an average of 9 turns. And the test set is split into two subset: **test seen set** and **test unseen set**. The topics in test seen set appear in the training set and the topics in test unseen set don't appear.

Holl-E is a movie domain dialog dataset using plot, reviews, comments and a fact table as ground-

Method	WoW(Test Seen)	WoW(Test Unseen)	Holl-E
Transformer MemNet(Dinan et al., 2018)	22.5	12.2	-
Transformer MemNet + Pretrain(Dinan et al., 2018)	24.5	23.7	-
SKT(Kim et al., 2019)	26.8	18.3	29.2
DIALKI(Wu et al., 2021)	32.9	35.5	-
Document Semantic Graphs(Li et al., 2022b)	29.4	30.8	37.7
CorefDiffs(Xu et al., 2022)	42.4	41.4	**40.9**
GenKS(Sun et al., 2023)	34.2	36.6	37.9
SPI(Xu et al., 2023)	36.5	34.8	38.3
Ours	**45.0**	**41.8**	36.0

Table 1: The knowledge selection experiment results on WoW and Holl-E. The results are reported in percentage(%).

Method	Test Seen	Test Unseen
Baseline	40.1	33.9
+ Graph	44.5	40.7
+ $\mathcal{L}_{word}$	**45.0**	**41.8**

Table 2: Ablation study on WoW. The results are reported in percentage(%). Baseline refers to the model that pretrained language model + MLP. Graph, $\mathcal{L}_{word}$ represent multi-level document semantic graph + residual and word node selection respectively.

Method	Test Seen	Test Unseen
Ours	45.0	41.8
w/o topic nodes	44.5	40.7
w/o word nodes	**45.0**	**41.8**

Table 3: Ablation study of the multi-level document semantic graph on WoW. The results are reported in percentage(%).

ing document. There are 7,729/930/913 dialogues in the training/validation/test set. Every dialogue contains 5 turns on average. Holl-E additionally provides multiple references for the test set and we only report performance for single reference.

4.1.2. Implementation Detail

We use Bert-base-uncased (Kenton and Toutanova, 2019) as the pretrained language model f_{LM} and concatenate 4 utterances to compose the $Text_{s_i}$, i.e. $p = 4$. We choose Graph Attention Network (GAT) (Veličković et al., 2018) as the GNN. GAT can be applied to inductive task which is exactly what we meet in our task. The size and structure of the graph constructed from different documents are always distinct. For WoW, We train the GAT with 128 hidden dimensions, 3 heads and stack 2 layers GAT. For Holl-E, We train one layer GAT with 128 hidden dimensions, 4 heads. The learning rate for Bert-base-uncased is set as $5e^{-5}$, $2e^{-3}$ for other parameters. α is set as $\frac{1}{2}$ and we train the model for 5 epochs. For WoW and Holl-E, the knowledge segment is at sentence level and we pick up the most appropriate segment as selected knowledge to match their settings. To deal with the utterances that don't utilize any knowledge, we add a special knowledge segment node **[no_passage_used]** into the graph. Also we preprocess Holl-E following Kim et al. (2019)'s script. The statistical details of multi-level document semantic graph can be found in Figure 4.

4.2. Baselines

We compare our method with the following models:

- **Transformer MemNet**(Dinan et al., 2018): It's the baseline released by author of WoW and uses a vanilla Transformer (Vaswani et al., 2017) to encode each sentence and dialogue context independently. The knowledge is selected based on the dot product attention between candidate sentences and dialogue context.
- **Transformer MemNet + Pretrain**(Dinan et al., 2018): It's another version of Transformer MemNet. It's pretrained on Reddit conversations.
- **Sequential Knowledge Transformer (SKT)**(Kim et al., 2019): It uses a sequential latent variable model to do knowledge selection in multi-turn dialogue.
- **DIALKI**(Wu et al., 2021): This model takes advantage of document structure to contextualize document passages together with the dialogue history. It selects the most relevant passage first and then locate the knowledge span.
- **Document Semantic Graphs**(Li et al., 2022b): This model constructs a document semantic

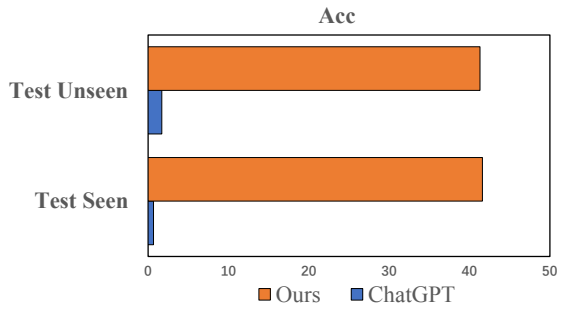

Figure 5: ChatGPT knowledge selection performance in WoW dataset, compared to our method. Acc is the accuracy of knowledge selection, reported in percentage(%).

graph from the document and uses the Edge-Aware Graph Attention Network to capture the semantics of the nodes and edges. Then they do knowledge selection over the graph.

- **CorefDiffs**(Xu et al., 2022): This model captures the inter- and intra-document knowledge relationship as a heterogeneous document graph and then integrates dialog flow for knowledge selection.
- **GenKS**(Sun et al., 2023): This model uses a sequence-to-sequence manner and captures the intra-knowledge and dialogue-knowledge interactions with the help of attention mechanism. They train the model with knowledge selection and generation together.
- **SPI**(Xu et al., 2023): SPI is a probabilistic model with dual latent variables, one discrete latent variable for knowledge selection and one continuous latent variable for response generation.

4.3. Experiment Results

To compare with above methods, we use accuracy of the knowledge selection results (**Accuracy**) as the main metric for evaluating.

$$Accuracy = \frac{T}{N} \tag{10}$$

T is the number of correctly selected samples for knowledge, whereas N denotes the total number of samples. The main results of the experiments are presented in Table 1.

4.3.1. Main Results

From Table 1, we can see that our method achieves competitive results and significantly outperforms all the baseline methods on WoW. Especially, compared to the previous best result reported by CorefDiffs, our method achieves 2.6% and 0.4% improvements in WoW test seen and WoW test unseen sets, respectively. Besides this, comparing the graph-based method CorefDiffs, our method with the generation method GenKS, and the extraction-based method DIALKI, we can find that graph structure has a great effect on improving the performance of knowledge selection. The performance drops in Holl-E, which we further analyze in Section 4.4.

4.3.2. Ablation Study

First, to study the impact of different modules of our model, we conduct many ablation experiments on WoW, and the results are shown in Table 2. For Baseline, we use Bert-base-uncased as the pre-trained language model to encode each knowledge segment independently and use a MLP to make predictions over all knowledge segment nodes. From Table 2, we can see that the performance of the model is significantly improved after adding the graph, proving that our multi-level semantic graph can effectively capture the semantics within documents. Furthermore, after adding $\mathcal{L}_{word}$, the accuracy has a 0.5% and 1.1% boost in test seen set and test unseen set, respectively. It shows that word node selection can not only help our model utilize the whole graph better but also help improve the model's performance on the main task.

Then, we further explore the importance of nodes at different granularities in multi-level document semantic graphs, and the results are shown in Table 3. Since knowledge segment nodes are the targets of knowledge selection, they cannot be removed, so we only conduct ablation experiments on title nodes and word nodes. "w/o word nodes" indicates that only the word nodes in the graph are removed. "w/o topic nodes" means that only the topic nodes in the graph are removed. When only the topic nodes are removed, the performance of the model declines more, indicating that topic nodes have a greater impact on the results than knowledge segment nodes. This might be because topic nodes are connected to all nodes, aggregating the information of the entire graph and facilitating the transfer of information within the graph.

4.3.3. Compare with Large Language Model

Large Language Models (LLMs), such as ChatGPT, have demonstrated huge potential in text classification task(Gilardi et al., 2023). However, whether LLMs can surpass our approach in knowledge selection task still needs to be explored. For ChatGPT, we design the prompt to instruct ChatGPT to make the selection:

Prompt: Now, we have a chat with the knowledge. I'm [user], you are [wizard]. Please tell me

Document length (Num of knowledge segments)	Test seen ≤ 40	≤ 45	≤ 50	≤ 55	> 55	Test unseen ≤ 40	≤ 45	≤ 50	≤ 55	> 55
DIALKI(Wu et al., 2021)	**65.0**	**50.6**	44.2	23.8	23.6	**66.8**	**55.2**	38.2	30.8	28.8
Ours	62.8	48.8	**46.8**	**41.6**	**35.4**	42.4	40.2	**39.6**	**38.2**	**42.4**

Table 4: Knowledge selection results under different document length on WoW. Document length is the number of candidate knowledge segments. The results are the average of 5 repeated experiments. The results are reported in percentage(%).

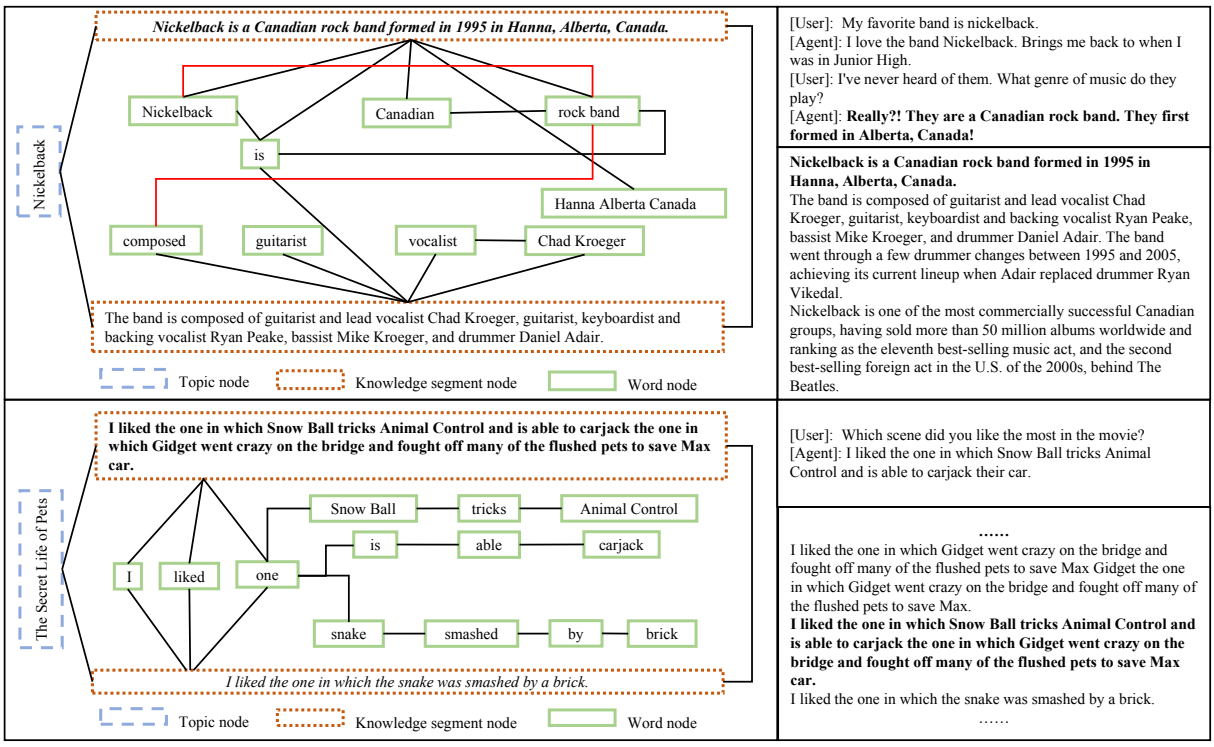

Figure 6: Samples of our knowledge selection result. The left side of the figure is the graph automatically generated and we delete the most of edges and nodes that are irrelevant. The bold node is the ground-truth knowledge segment node. The node with italic text is the node predicted by the model. The upper right is the dialogue and the lower right is the piece of grounding document.

which knowledge you would use to generate the next response. The dialogue is [dialogue]. The knowledge is [knowledge]. Please tell me the id of knowledge which is used to generate the next response: [id].

The details of the prompt can be found in Appendix A.To ensure that ChatGPT generates responses in the correct format, we provide one example randomly selected from the WoW training set for ChatGPT. We choose the gpt-3.5-turbo[3] to conduct experiments and randomly select 100 samples from both WoW test seen and WoW test unseen set to do evaluation. We repeat the experiments for three times and report the average results in Figure 5. We can find that the performance of ChatGPT is inferior to our method in a scenario with multiple candidate options. In our task, the average number of candidate options is 40, and Gilardi et al. (2023)'s setting is 14.

4.4. Analysis

Document length. Following Wu et al. (2021), we further study the knowledge selection performance under different document length sets. In each document length interval, we randomly select 100 samples from both WoW test seen and WoW test unseen set to do evaluation. And we repeat the experiments for 5 times and report the average results in Table 4. We can find that both in test seen and test unseen set, our method achieves better performance than DIALKI when document length is greater than 45. At the same time, the results show

[3]https://platform.openai.com/docs/models/gpt-3-5

that our method has fewer drops as the document length increases compared to DIALKI. In test seen set, our model can still maintain good performance when the document length is less than 55. For test unseen set, our method has better generalization ability. Even when the document length is greater than 55, the performance is the same as that of the document length less than 40.

Case study. Figure 6 includes two cases. The upper part of Figure 6 is picked from WoW test unseen set, and the model gives the right answer. We can see that the documents in WoW are complete documents with contextual semantic associations. The graph can distinguish that "The band" is referred to "rock band", i.e. "Nickelback" (linked with red edges). Also, with the help of pretrained language model, the model understands that"rock" can be one of the "genre of music" in the dialogue context. It reveals that our model can not only understand dialogue context well but also utilize the document through multi-level semantic graph effectively.

The bottom of Figure 6 is picked from Holl-E test set. We can see that our method successfully constructs the semantic structure of sentences. But documents in Holl-E contain comments from different users. Between these comments, there is no contextual semantic connection, and they have similar sentence structures, like "I liked the ..." in our case. During the construction of the graph, the same nodes from different sentences would be merged into one node. However, they don't contain any contextual semantic connection. So it can lead to our model becoming confused and making incorrect predictions. Those two cases reveal that our method is suitable for documents with a coherent semantic structure rather than loosely structured ones.

5. Conclusion

Knowledge selection is the crucial sub-task of Documents Grounded Dialogue System. In this paper, we devise an automatic method to construct a multi-level document semantic graph from the grounding document. To leverage the graph better, we also devise an auxiliary task to help with the learning of knowledge selection task. We conduct comprehensive experiments, and the results of the experiments verify the effectiveness of the method we propose, and we achieve state-of-the-art performance in WoW. Moreover, the results in long document situations show that our model has excellent stability to maintain good performance. Further analysis shows our method is better suited for documents with a complete semantic structure than loosely structured ones.

For future study, we would substitute the GAN with a more advanced attention-based neural network to improve our model's performance, e.g. HAN(Wang et al., 2019) and BA-GNN(Iyer et al., 2021). The current study still suffers from the limitation of pretrained language model's input length. Also we think it would be meaningful to consider how to combine our multi-level document semantic graph with the large language model to achieve better reasoning ability on long documents.

6. References

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.

Nico Daheim, David Thulke, Christian Dugast, and Hermann Ney. 2021. Cascaded span extraction and response generation for document-grounded dialog. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*, pages 57–62.

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*.

Chang Gao, Wenxuan Zhang, and Wai Lam. 2022. Unigdd: A unified generative framework for goal-oriented document-grounded dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Xinxian Huang, Huang He, Siqi Bao, Fan Wang, Hua Wu, and Haifeng Wang. 2021. Plato-kag: Unsupervised knowledge-grounded conversation via joint modeling. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 143–154.

Roshni G Iyer, Wei Wang, and Yizhou Sun. 2021. Bi-level attention graph neural networks. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1126–1131. IEEE.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2019. Sequential latent knowledge selection for knowledge-grounded dialogue. In *International Conference on Learning Representations*.

Kun Li, Tianhua Zhang, Liping Tang, Junan Li, Hongyuan Lu, Xixin Wu, and Helen Meng. 2022a. Grounded dialogue generation with cross-encoding re-ranker, grounding span prediction, and passage dropout. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 123–129.

Sha Li, Mahdi Namazifar, Di Jin, Mohit Bansal, Heng Ji, Yang Liu, and Dilek Hakkani-Tur. 2022b. Enhancing knowledge selection for grounded dialogues via document semantic graphs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.

Longxuan Ma, Wei-Nan Zhang, Mingda Li, and Ting Liu. 2020. A survey of document grounded dialogue systems (dgds). *arXiv preprint arXiv:2004.13818*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Vaishali Pal, Evangelos Kanoulas, and Maarten de Rijke. 2022. Parameter-efficient abstractive question answering over tables or text. *DialDoc 2022*, page 41.

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.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2020. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2453–2470.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Weiwei Sun, Pengjie Ren, and Zhaochun Ren. 2023. Generative knowledge selection for knowledge-grounded dialogues. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2032–2043.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks.

In *International Conference on Learning Representations*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032.

Zeqiu Wu, Bo-Ru Lu, Hannaneh Hajishirzi, and Mari Ostendorf. 2021. DIALKI: Knowledge identification in conversational systems through dialogue-document contextualization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1852–1863, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lin Xu, Qixian Zhou, Jinlan Fu, Min-Yen Kan, and See Kiong Ng. 2022. Corefdiffs: Co-referential and differential knowledge flow in document grounded conversations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 471–484.

Yan Xu, Deqian Kong, Dehong Xu, Ziwei Ji, Bo Pang, Pascale Fung, and Ying Nian Wu. 2023. Diverse and faithful knowledge-grounded dialogue generation via sequential posterior inference. In *International Conference on Machine Learning*, pages 38518–38534. PMLR.

Weihao Zeng, Keqing He, Zechen Wang, Dayuan Fu, Guanting Dong, Ruotong Geng, Pei Wang, Jingang Wang, Chaobo Sun, Wei Wu, et al. 2022. Semi-supervised knowledge-grounded pre-training for task-oriented dialog systems. In *Proceedings of the Towards Semi-Supervised and Reinforced Task-Oriented Dialog Systems (SereTOD)*, pages 39–47.

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.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287.

A. Prompt for ChatGPT

For ChatGPT, we set the prompt as:

Prompt: Now, we have a chat with the knowledge. I'm [user], you are [wizard]. Please tell me which knowledge you would use to generate the next response.The dialogue is [dialogue]. The knowledge is [knowledge]. Please tell me the id of knowledge which is used to generate the next response: [id].

[dialogue], [knowledge], [id] are special tokens, which will be replaced in specific examples. For example, the special token [dialogue] is replaced by the dialogue history like "[user]: My favorite band is nickelback. [wizard]: I love the band Nickelback. Brings me back to when I was in Junior High…". The special token [knowledge] is replaced by "[know_1] Nickelback is a Canadian rock band formed in 1995 in Hanna, Alberta, Canada. [know_2] The band is composed of guitarist and lead vocalist Chad…". [id] is the index of the grounding truth knowledge segment which will not be provided to ChatGPT during testing. ChatGPT needs to generate it. And it can be replaced like "[know_2]".