Grounded Conversation Generation as Guided Traverses in Commonsense Knowledge Graphs

Houyu Zhang[1] *† **Zhenghao Liu**[2]* **Chenyan Xiong**[3] **Zhiyuan Liu**[2]
[1]Department of Computer Science, Brown University, Providence, USA
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
[3]Microsoft Research AI, Redmond, USA

Abstract

Human conversations naturally evolve around related concepts and scatter to multi-hop concepts. This paper presents a new conversation generation model, ConceptFlow, which leverages commonsense knowledge graphs to explicitly model conversation flows. By grounding conversations to the concept space, ConceptFlow represents the potential conversation flow as traverses in the concept space along commonsense relations. The traverse is guided by graph attentions in the concept graph, moving towards more meaningful directions in the concept space, in order to generate more semantic and informative responses. Experiments on Reddit conversations demonstrate ConceptFlow's effectiveness over previous knowledge-aware conversation models and GPT-2 based models while using 70% fewer parameters, confirming the advantage of explicit modeling conversation structures. All source codes of this work are available at https://github.com/thunlp/ConceptFlow.

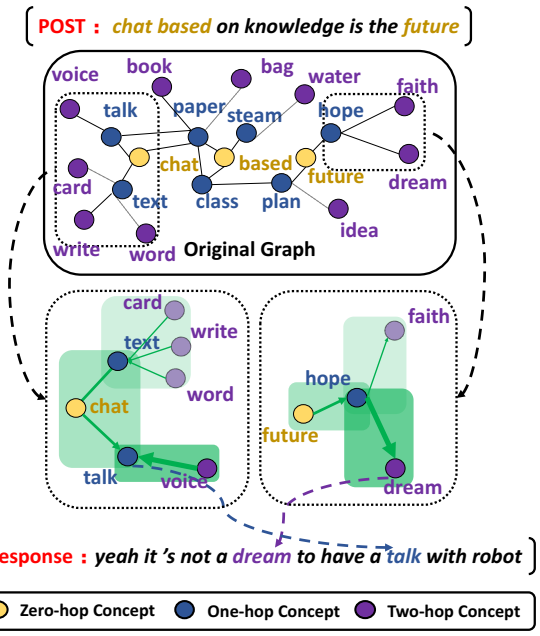


Figure 1: An Example of Concept Shift in a Conversation. Darker green indicates higher relevance and wider arrow indicates stronger concept shift (captured by ConceptFlow).

1 Introduction

The rapid advancements of language modeling and natural language generation (NLG) techniques have enabled fully data-driven conversation models, which directly generate natural language responses for conversations (Shang et al., 2015; Vinyals and Le, 2015; Li et al., 2016b). However, it is a common problem that the generation models may degenerate dull and repetitive contents (Holtzman et al., 2019; Welleck et al., 2019), which, in conversation assistants, leads to off-topic and useless responses. (Tang et al., 2019; Zhang et al., 2018; Gao et al., 2019).

Conversations often develop around Knowledge. A promising way to address the degeneration problem is to ground conversations with external knowledge (Xing et al., 2017), such as open-domain knowledge graph (Ghazvininejad et al., 2018), commonsense knowledge base (Zhou et al., 2018a), or background documents (Zhou et al., 2018b). Recent research leverages such external knowledge by using them to ground conversations, integrating them as additional representations, and then generating responses conditioned on both the texts and the grounded semantics (Ghazvininejad et al., 2018; Zhou et al., 2018a,b).

Integrating external knowledge as extra semantic representations and additional inputs to the conversation model effectively improves the quality of generated responses (Ghazvininejad et al., 2018; Logan et al., 2019; Zhou et al., 2018a). Never-

*Indicates equal contribution.
†Part of work is conducted at Tsinghua University.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2031–2043
July 5 - 10, 2020.

theless, some research on discourse development suggests that human conversations are not "still": People chat around a number of related concepts, and shift their focus from one concept to others. Grosz and Sidner (1986) models such concept shift by breaking discourse into several segments, and demonstrating different concepts, such as objects and properties, are needed to interpret different discourse segments. Attentional state is then introduced to represent the concept shift corresponding to each discourse segment. Fang et al. (2018) shows that people may switch dialog topics entirely in a conversation. Restricting the utilization of knowledge only to those directly appear in the conversation, effective as they are, does not reach the full potential of knowledge in modeling human conversations.

To model the concept shift in human conversations, this work presents ConceptFlow (**Con**versation generation with Con**cept Flow**), which leverages commonsense knowledge graphs to model the conversation flow in the explicit concept space. For example, as shown in Figure 1, the concepts of a conversation from Reddit evolves from "chat" and "future", to adjacent concept "talk", and also hops to distant concept "dream" along the commonsense relations—a typical involvement in natural conversations. To better capture this conversation structure, ConceptFlow explicitly models the conversations as traverses in commonsense knowledge graphs: it starts from the grounded concepts, e.g., "chat" and "future", and generates more meaningful conversations by hopping along the commonsense relations to related concepts, e.g., "talk" and "dream".

The traverses in the concept graph are guided by graph attention mechanisms, which derives from graph neural networks to attend on more appropriate concepts. ConceptFlow learns to model the conversation development along more meaningful relations in the commonsense knowledge graph. As a result, the model is able to "grow" the grounded concepts by hopping from the conversation utterances, along the commonsense relations, to distant but meaningful concepts; this guides the model to generate more informative and on-topic responses. Modeling commonsense knowledge as concept flows, is both a good practice on improving response diversity by scattering current conversation focuses to other concepts (Chen et al., 2017), and an implementation solution of the attentional state mentioned above (Grosz and Sidner, 1986).

Our experiments on a Reddit conversation dataset with a commonsense knowledge graph, ConceptNet (Speer et al., 2017), demonstrate the effectiveness of ConceptFlow. In both automatic and human evaluations, ConceptFlow significantly outperforms various seq2seq based generation models (Sutskever et al., 2014), as well as previous methods that also leverage commonsense knowledge graphs, but as static memories (Zhou et al., 2018a; Ghazvininejad et al., 2018; Zhu et al., 2017). Notably, ConceptFlow also outperforms two fine-tuned GPT-2 systems (Radford et al., 2019), while using *70% fewer* parameters. Explicitly modeling conversation structure provides better parameter efficiency.

We also provide extensive analyses and case studies to investigate the advantage of modeling conversation flow in the concept space. Our analyses show that many Reddit conversations are naturally aligned with the paths in the commonsense knowledge graph; incorporating distant concepts significantly improves the quality of generated responses with more on-topic semantic information added. Our analyses further confirm the effectiveness of our graph attention mechanism in selecting useful concepts, and ConceptFlow's ability in leveraging them to generate more relevant, informative, and less repetitive responses.

2 Related Work

Sequence-to-sequence models, e.g., Sutskever et al. (2014), have been widely used for natural language generation (NLG), and to build conversation systems (Shang et al., 2015; Vinyals and Le, 2015; Li et al., 2016b; Wu et al., 2019). Recently, pre-trained language models, such as ELMO (Devlin et al., 2019), UniLM (Dong et al., 2019) and GPT-2 (Radford et al., 2018), further boost the NLG performance with large scale pretraining. Nevertheless, the degenerating of irrelevant, off-topic, and non-useful responses is still one of the main challenges in conversational generation (Rosset et al., 2020; Tang et al., 2019; Zhang et al., 2018; Gao et al., 2019).

Recent work focuses on improving conversation generation with external knowledge, for example, incorporating additional texts (Ghazvininejad et al., 2018; Vougiouklis et al., 2016; Xu et al., 2017; Long et al., 2017), or knowledge graphs (Long et al., 2017; Ghazvininejad et al., 2018). They have

shown external knowledge effectively improves conversation response generation.

The structured knowledge graphs include rich semantics represented via entities and relations (Hayashi et al., 2019). Lots of previous studies focus on task-targeted dialog systems based on domain-specific knowledge bases (Xu et al., 2017; Zhu et al., 2017; Gu et al., 2016). To generate responses with a large-scale knowledge base, Zhou et al. (2018a) and Liu et al. (2018) utilize graph attention and knowledge diffusion to select knowledge semantics for utterance understanding and response generation. Moon et al. (2019) focuses on the task of entity selection, and takes advantage of positive entities that appear in the golden response. Different from previous research, ConceptFlow models the conversation flow explicitly with the commonsense knowledge graph and presents a novel attention mechanism on all concepts to guide the conversation flow in the latent concept space.

3 Methodology

This section presents our **Con**versation generation model with latent Con**cept Flow** (ConceptFlow). Our model grounds the conversation in the concept graph and traverses to distant concepts along commonsense relations to generate responses.

3.1 Preliminary

Given a user utterance $X = \{x_1, ..., x_m\}$ with m words, conversation generation models often use an encoder-decoder architecture to generate a response $Y = \{y_1, ..., y_n\}$.

The encoder represents the user utterance X as a representation set $H = \{\vec{h}_1, ..., \vec{h}_m\}$. This is often done by Gated Recurrent Units (GRU):

$$\vec{h}_i = \text{GRU}(\vec{h}_{i-1}, \vec{x}_i), \tag{1}$$

where the $\vec{x}_i$ is the embedding of word x_i.

The decoder generates t-th word in the response according to the previous $t-1$ generated words $y_{<t} = \{y_1, ..., y_{t-1}\}$ and the user utterance X:

$$P(Y|X) = \prod_{t=1}^{n} P(y_t|y_{<t}, X). \tag{2}$$

Then it minimizes the cross-entropy loss L and optimizes all parameters end-to-end:

$$L = \sum_{t=1}^{n} \text{CrossEntropy}(y_t^*, y_t), \tag{3}$$

where y_t^* is the token from the golden response.

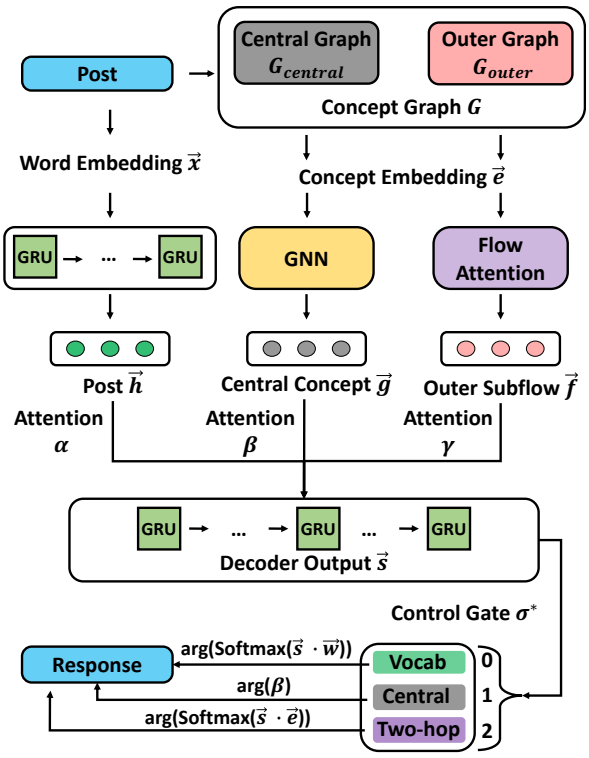


Figure 2: The Architecture of ConceptFlow.

The architecture of ConceptFlow is shown in Figure 2. ConceptFlow first constructs a concept graph G with central graph G_{central} and outer graph G_{outer} according to the distance (hops) from the grounded concepts (Sec. 3.2).

Then ConceptFlow encodes both central and outer concept flows in central graph G_{central} and outer graph G_{outer} , using graph neural networks and concept embedding (Sec. 3.3).

The decoder, presented in Section 3.4, leverages the encodings of concept flows and the utterance to generate words or concepts for responses.

3.2 Concept Graph Construction

ConceptFlow constructs a concept graph G as the knowledge for each conversation. It starts from the grounded concepts (zero-hop concepts V^0), which appear in the conversation utterance and annotated by entity linking systems.

Then, ConceptFlow grows zero-hop concepts V^0 with one-hop concepts V^1 and two-hop concepts V^2. Concepts from V^0 and V^1, as well as all relations between them, form the central concept graph G_{central}, which is closely related to the current conversation topic. Concepts in V^1 and V^2 and their connections form the outer graph G_{outer}.

3.3 Encoding Latent Concept Flow

The constructed concept graph provides explicit semantics on how concepts related to commonsense knowledge. ConceptFlow utilizes it to model the conversation and guide the response generation. It starts from the user utterance, traversing through central graph G_{central}, to outer graph G_{outer}. This is modeled by encoding the central and outer concept flows according to the user utterance.

Central Flow Encoding. The central concept graph G_{central} is encoded by a graph neural network that propagates information from user utterance H to the central concept graph. Specifically, it encodes concept $e_i \in G_{\text{central}}$ to representation $\vec{g}_{e_i}$:

$$\vec{g}_{e_i} = \text{GNN}(\vec{e}_i, G_{\text{central}}, H), \tag{4}$$

where $\vec{e}_i$ is the concept embedding of e_i. There is no restriction of which GNN model to use. We choose Sun et al. (2018)'s GNN (GraftNet), which shows strong effectiveness in encoding knowledge graphs. More details of GraftNet can be found in Appendix A.3.

Outer Flow Encoding. The outer flow f_{e_p}, hopping from $e_p \in V_1$ to its connected two-hop concept e_k, is encoded to $\vec{f}_{e_p}$ by an attention mechanism:

$$\vec{f}_{e_p} = \sum_{e_k} \theta^{e_k} \cdot [\vec{e}_p \circ \vec{e}_k], \tag{5}$$

where $\vec{e}_p$ and $\vec{e}_k$ are embeddings for e_p and e_k, and are concatenated ($\circ$). The attention θ^{e_k} aggregates concept triple (e_p, r, e_k) to get $\vec{f}_{e_p}$:

$$\theta^{e_k} = \text{softmax}((w_r \cdot \vec{r})^\top \cdot \tanh(w_h \cdot \vec{e}_p + w_t \cdot \vec{e}_k)), \tag{6}$$

where $\vec{r}$ is the relation embedding between the concept e_p and its neighbor concept e_k. w_r, w_h and w_t are trainable parameters. It provides an efficient attention specifically focusing on the relations for multi-hop concepts.

3.4 Generating Text with Concept Flow

To consider both user utterance and related information, the texts from the user utterance and the latent concept flows are incorporated by decoder using two components: 1) the context representation that combines their encodings (Sec. 3.4.1); 2) the conditioned generation of words and concepts from the context representations (Sec. 3.4.2).

3.4.1 Context Representation

To generate t-th time response token, we first calculate the output context representation $\vec{s}_t$ for t-th time decoding with the encodings of the utterance and the latent concept flow.

Specifically, $\vec{s}_t$ is calculated by updating the $(t-1)$-th step output representation $\vec{s}_{t-1}$ with the $(t-1)$-th step context representation $\vec{c}_{t-1}$:

$$\vec{s}_t = \text{GRU}(\vec{s}_{t-1}, [\vec{c}_{t-1} \circ \vec{y}_{t-1}]), \tag{7}$$

where $\vec{y}_{t-1}$ is the $(t-1)$-th step generated token y_{t-1}'s embedding, and the context representation $\vec{c}_{t-1}$ concatenates the text-based representation $\vec{c}_{t-1}^{\text{text}}$ and the concept-based representation $\vec{c}_{t-1}^{\text{concept}}$:

$$\vec{c}_{t-1} = \text{FFN}([\vec{c}_{t-1}^{\text{text}} \circ \vec{c}_{t-1}^{\text{cpt}}]). \tag{8}$$

The **text-based representation** $\vec{c}_{t-1}^{\text{text}}$ reads the user utterance encoding H with a standard attention mechanism (Bahdanau et al., 2015):

$$\vec{c}_{t-1}^{\text{text}} = \sum_{i=1}^{m} \alpha_{t-1}^{j} \cdot \vec{h}_j, \tag{9}$$

and attentions α_{t-1}^{j} on the utterance tokens:

$$\alpha_{t-1}^{j} = \text{softmax}(\vec{s}_{t-1} \cdot \vec{h}_j). \tag{10}$$

The **concept-based representation** $\vec{c}_{t-1}^{\text{concept}}$ is a combination of central and outer flow encodings:

$$\vec{c}_{t-1}^{\text{cpt}} = \left(\sum_{e_i \in G_{\text{central}}} \beta_{t-1}^{e_i} \cdot \vec{g}_{e_i} \right) \circ \left(\sum_{f_{e_p} \in G_{\text{outer}}} \gamma_{t-1}^{f} \cdot \vec{f}_{e_p} \right). \tag{11}$$

The attention $\beta_{t-1}^{e_i}$ weights over central concept representations:

$$\beta_{t-1}^{e_i} = \text{softmax}(\vec{s}_{t-1} \cdot \vec{g}_{e_i}), \tag{12}$$

and the attention γ_{t-1}^{f} weights over outer flow representations:

$$\gamma_{t-1}^{f} = \text{softmax}(\vec{s}_{t-1} \cdot \vec{f}_{e_p}). \tag{13}$$

3.4.2 Generating Tokens

The t-th time output representation $\vec{s}_t$ (Eq. 7) includes information from both the utterance text, the concepts with different hop steps, and the attentions upon them. The decoder leverages $\vec{s}_t$ to generate the t-th token to form more informative responses.

It first uses a gate σ^* to control the generation by choosing words ($\sigma^* = 0$), central concepts ($V^{0,1}$, $\sigma^* = 1$) and outer concept set (V^2, $\sigma^* = 2$):

$$\sigma^* = \text{argmax}_{\sigma \in \{0,1,2\}}(\text{FFN}_\sigma(\vec{s}_t)), \tag{14}$$

The generation probabilities of word w, central concept e_i, and outer concepts e_k are calculated

over the word vocabulary, central concept set $V^{0,1}$, and outer concept set V^2:

$$y_t \sim \begin{cases} \text{softmax}(\vec{s}_t \cdot \vec{w}), \sigma^* = 0 \\ \text{softmax}(\vec{s}_t \cdot \vec{g}_{e_i}), \sigma^* = 1 \\ \text{softmax}(\vec{s}_t \cdot \vec{e}_k), \sigma^* = 2, \end{cases} \quad (15)$$

where $\vec{w}$ is the word embedding for word w, $\vec{g}_{e_i}$ is the central concept representation for concept e_i and $\vec{e}_k$ is the two-hop concept e_k's embedding.

The training and prediction of ConceptFlow are conducted following standard conditional language models, i.e. using Eq. 15 in place of Eq. 2 and training it by the Cross-Entropy loss (Eq. 3). Only ground truth responses are used in training and no additional annotation is required.

4 Experiment Methodology

This section describes the dataset, evaluation metrics, baselines, and implementation details of our experiments.

Dataset. All experiments use the multi-hop extended conversation dataset based on a previous dataset which collects single-round dialogs from Reddit (Zhou et al., 2018a). Our dataset contains 3,384,185 training pairs and 10,000 test pairs. Preprocessed ConceptNet (Speer et al., 2017) is used as the knowledge graph, which contains 120,850 triples, 21,471 concepts and 44 relation types.

Evaluation Metrics. A wide range of evaluation metrics are used to evaluate the quality of generated responses: PPL (Serban et al., 2016), Bleu (Papineni et al., 2002), Nist (Doddington, 2002), ROUGE (Lin, 2004) and Meteor (Lavie and Agarwal, 2007) are used for relevance and repetitiveness; Dist-1, Dist-2 and Ent-4 are used for diversity, which is same with the previous work (Li et al., 2016a; Zhang et al., 2018). The metrics above are evaluated using the implementation from Galley et al. (2018). Zhou et al. (2018a)'s concept PPL mainly focuses on concept grounded models and this metric is reported in Appendix A.1.

The Precision, Recall, and F1 scores are used to evaluate the quality of learned latent concept flow in predicting the golden concepts which appear in ground truth responses.

Baselines. The six baselines compared come from three groups: standard Seq2Seq, knowledge-enhanced ones, and fine-tuned GPT-2 systems.

Seq2Seq (Sutskever et al., 2014) is the basic encoder-decoder for language generation.

Knowledge-enhanced baselines include MemNet (Ghazvininejad et al., 2018), CopyNet (Zhu et al., 2017) and CCM (Zhou et al., 2018a). MemNet maintains a memory to store and read concepts. CopyNet copies concepts for the response generation. CCM (Zhou et al., 2018a) leverages a graph attention mechanism to model the central concepts. These models mainly focus on the grounded concepts. They do not explicitly model the conversation structures using multi-hop concepts.

GPT-2 (Radford et al., 2019), the pre-trained model that achieves the state-of-the-art in lots of language generation tasks, is also compared in our experiments. We fine-tune the 124M GPT-2 in two ways: concatenate all conversations together and train it like a language model (GPT-2 *lang*); extend the GPT-2 model with encode-decoder architecture and supervise with response data (GPT-2 *conv*).

Implement Details. The zero-hop concepts are initialized by matching the keywords in the post to concepts in ConceptNet, the same with CCM (Zhou et al., 2018a). Then zero-hop concepts are extended to their neighbors to form the central concept graph. The outer concepts contain a large amount of two-hop concepts with lots of noises. To reduce the computational cost, we first train ConceptFlow (select) with 10% random training data, and use the learned graph attention to select top 100 two-hop concepts over the whole dataset. Then the standard train and test are conducted with the pruned graph. More details of this filtering step can be found in Appendix A.4.

TransE (Bordes et al., 2013) embedding and Glove (Pennington et al., 2014) embedding are used to initialize the representation of concepts and words, respectively. Adam optimizer with the learning rate of 0.0001 is used to train the model.

5 Evaluation

Five experiments are conducted to evaluate the generated responses from ConceptFlow and the effectiveness of the learned graph attention.

5.1 Response Quality

This experiment evaluates the generation quality of ConceptFlow automatically and manually.

Automatic Evaluation. The quality of generated responses is evaluated with different metrics from three aspects: relevance, diversity, and novelty. Table 1 and Table 2 show the results.

In Table 1, all evaluation metrics calculate the relevance between the generated response and the

Model	Bleu-4	Nist-4	Rouge-1	Rouge-2	Rouge-L	Meteor	PPL
Seq2Seq	0.0098	1.1069	0.1441	0.0189	0.1146	0.0611	48.79
MemNet	0.0112	1.1977	0.1523	0.0215	0.1213	0.0632	47.38
CopyNet	0.0106	1.0788	0.1472	0.0211	0.1153	0.0610	43.28
CCM	0.0084	0.9095	0.1538	0.0211	0.1245	0.0630	42.91
GPT-2 (lang)	0.0162	1.0844	0.1321	0.0117	0.1046	0.0637	29.08*
GPT-2 (conv)	0.0124	1.1763	0.1514	0.0222	0.1212	0.0629	24.55*
ConceptFlow	**0.0246**	**1.8329**	**0.2280**	**0.0469**	**0.1888**	**0.0942**	**29.90**

Table 1: Relevance Between Generated and Golden Responses. The PPL results* of GPT-2 is not directly comparable because of its different tokenization. More results can be found in Appendix A.1.

	Diversity(↑)			Novelty w.r.t. Input(↓)				
Model	Dist-1	Dist-2	Ent-4	Bleu-4	Nist-4	Rouge-2	Rouge-L	Meteor
Seq2Seq	0.0123	0.0525	7.665	0.0129	1.3339	0.0262	**0.1328**	**0.0702**
MemNet	0.0211	0.0931	8.418	0.0408	2.0348	0.0621	0.1785	0.0914
CopyNet	0.0223	0.0988	8.422	0.0341	1.8088	0.0548	0.1653	0.0873
CCM	0.0146	0.0643	7.847	0.0218	**1.3127**	0.0424	0.1581	0.0813
GPT-2 (lang)	**0.0325**	**0.2461**	**11.65**	0.0292	1.7461	0.0359	0.1436	0.0877
GPT-2 (conv)	0.0266	0.1218	8.546	0.0789	2.5493	0.0938	0.2093	0.1080
ConceptFlow	0.0223	0.1228	10.27	**0.0126**	1.4749	**0.0258**	0.1386	0.0761

Table 2: Diversity (higher better) and Novelty (lower better) of Generated Response. Diversity is calculated within generated responses; Novelty compares generated responses to the input post. More results are in Appendix A.1.

Model	Parameter	Average Score		Best@1 Ratio	
		App.	Inf.	App.	Inf.
CCM	35.6M	1.802	1.802	17.0%	15.6%
GPT-2 (conv)	124.0M	2.100	1.992	26.2%	23.6%
ConceptFlow	35.3M	2.690	2.192	30.4%	25.6%
Golden	Human	2.902	3.110	67.4%	81.8%

Table 3: Human Evaluation on Appropriate (App.) and Informativeness (Inf.). The Average Score takes the average from human judgments. Best@1 Ratio indicates the fraction of judges consider the case as the best. The number of parameters are also presented.

Model	App.	Inf.
ConceptFlow-CCM	0.3724	0.2641
ConceptFlow-GPT2	0.2468	0.2824

Table 4: Fleiss' Kappa of Human Agreement. Two testing scenarios Appropriate (App.) and Informativeness (Inf.) are used to evaluate the the quality of generated response. The Fleiss' Kappa evaluates agreement from various annotators and focuses on the comparison of two models with three categories: win, tie and loss.

golden response. ConceptFlow outperforms all baseline models by large margins. The responses generated by ConceptFlow are more on-topic and match better with the ground truth responses.

In Table 2, Dist-1, Dist-2, and Ent-4 measure the word diversity of generated responses and the rest of metrics measure the novelty by comparing the generated response with the user utterance. ConceptFlow has a good balance in generating novel and diverse responses. GPT-2's responses are more diverse, perhaps due to its sampling mechanism during decoding, but are less novel and on-topic compared to those from ConceptFlow.

Human Evaluation. The human evaluation focuses on two aspects: appropriateness and informativeness. Both are important for conversation systems (Zhou et al., 2018a). Appropriateness evaluates if the response is on-topic for the given utterance; informativeness evaluates systems' ability to provide new information instead of copying from the utterance (Zhou et al., 2018a). All responses of sampled 100 cases are selected from four methods with better performances: CCM, GPT-2 (conv), ConceptFlow, and Golden Response. The responses are scored from 1 to 4 by five judges (the higher the better).

Table 3 presents Average Score and Best@1 ratio from human judges. The first is the mean of five judges; the latter calculates the fraction of judges that consider the corresponding response the best among four systems. ConceptFlow outperforms all other models in all scenarios, while only using 30% of parameters compared to GPT-2. This demonstrates the advantage of explicitly modeling conversation flow with structured semantics.

The agreement of human evaluation is tested to demonstrate the authenticity of evaluation results. We first sample 100 cases randomly for our human evaluation. Then the responses from four better

conversation systems, CCM, GPT-2 (conv), ConceptFlow and Golden Responses, are provided with a random order. A group of annotators are asked to score each response ranged from 1 to 4 according to the quality on two testing scenarios, appropriateness and informativeness. All annotators have no clues about the source of generated responses.

The agreement of human evaluation for CCM, GPT-2 (conv) and ConceptFlow are presented in Table 4. For each case, the response from ConceptFlow is compared to the responses from two baseline models, CCM and GPT-2 (conv). The comparison result is divided into three categories: win, tie and loss. Then the human evaluation agreement is calculated with Fleiss' Kappa (κ). The κ value ranges from 0.21 to 0.40 indicating fair agreement, which confirms the quality of human evaluation.

Both automatic and human evaluations illustrate the effectiveness of ConceptFlow. The next experiment further studies the effectiveness of multi-hop concepts in ConceptFlow.

5.2 Effectiveness of Multi-hop Concepts

This part explores the role of multi-hop concepts in ConceptFlow. As shown in Figure 3, three experiments are conducted to evaluate the performances of concept selection and the quality of generated responses with different sets of concepts.

This experiment considers four variations of outer concept selections. *Base* ignores two-hop concepts and only considers the central concepts. *Rand*, *Distract*, and *Full* add two-hop concepts in three different ways: *Rand* selects concepts randomly, *Distract* selects all concepts that appear in the golden response with random negatives (distractors), and *Full* is our ConceptFlow (select) that selects concepts by learned graph attentions.

As shown in Figure 3(a), *Full* covers more golden concepts than *Base*. This aligns with our motivation that natural conversations do flow from central concepts to multi-hop ones. Compared to *Distract* setting where all ground truth two-hop concepts are added, ConceptFlow (select) has slightly less coverage but significantly reduces the number of two-hop concepts.

The second experiment studies the model's ability to generate ground truth concepts, by comparing the concepts in generated responses with those in ground truth responses. As shown in Figure 3(b), though *Full* filtered out some golden two-hop concepts, it outperforms other variations by large margins. This shows ConceptFlow's graph attention mechanisms effectively leverage the pruned concept graph and generate high-quality concepts when decoding.

Depth	Amount	Golden Coverage	
		Ratio	Number
Zero-hop	5.8	9.81%	0.579
+ One-hop	98.6	38.78%	2.292
+ Two-hop	880.8	61.37%	3.627
+ Three-hop	3769.1	81.58%	4.821
ConceptFlow	198.6	52.10%	3.075

Table 5: Statistics of Concept Graphs with different hops, including the total Amount of connected concepts, the Ratio and Number of covered golden concepts (those appear in ground truth responses). ConceptFlow indicates the filtered two-hop graph.

The high-quality latent concept flow leads to better modeling of conversations, as shown in Figure 3(c). *Full* outperforms *Distract* in their generated responses' token level perplexity, even though *Distract* includes all ground truth two-hop concepts. This shows that "negatives" selected by ConceptFlow, while not directly appear in the target response, are also on-topic and include meaningful information, as they are selected by graph attentions instead of random.

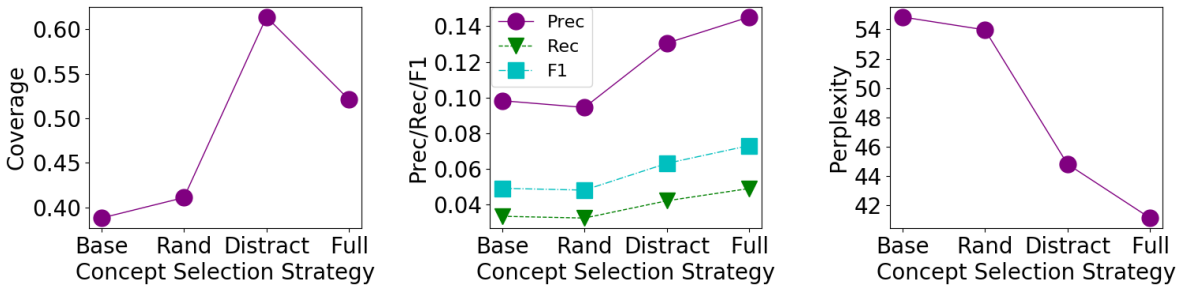
More studies of multi-hop concept selection strategies can be found in Appendix A.2.

5.3 Hop Steps in Concept Graph

This experiment studies the influence of hop steps in the concept graph.

As shown in Table 5, the Number of covered golden concepts increases with more hops. Compared to zero-hop concepts, multi-hop concepts cover more golden concepts, confirming that conversations naturally shift to multi-hop concepts: extending the concept graph from one-hop to two-hop improves the recall from 39% to 61%, and to three-hop further improves to 81%.

However, at the same time, the amounts of the concepts also increase dramatically with multiple hops. Three hops lead to 3,769 concepts on average, which are 10% of the entire graph we used. In this work, we choose two-hop, as a good balance of coverage and efficiency, and used ConceptFlow (select) to filter around 200 concepts to construct the pruned graph. How to efficiently and effectively leverage more distant concepts in the graph is reserved for future work.



(a) Golden Concept Coverage. (b) Response Concept Generation. (c) Response Token Generation.

Figure 3: Comparisons of Outer Concept Selection Methods. *Base* only considers the central concepts and ignores two-hop concepts. *Rand* randomly selects two-hop concepts. *Distract* incorporates golden concepts in the response with random negatives (distractors). *Full* chooses two-hop concepts with ConceptFlow's graph attention.

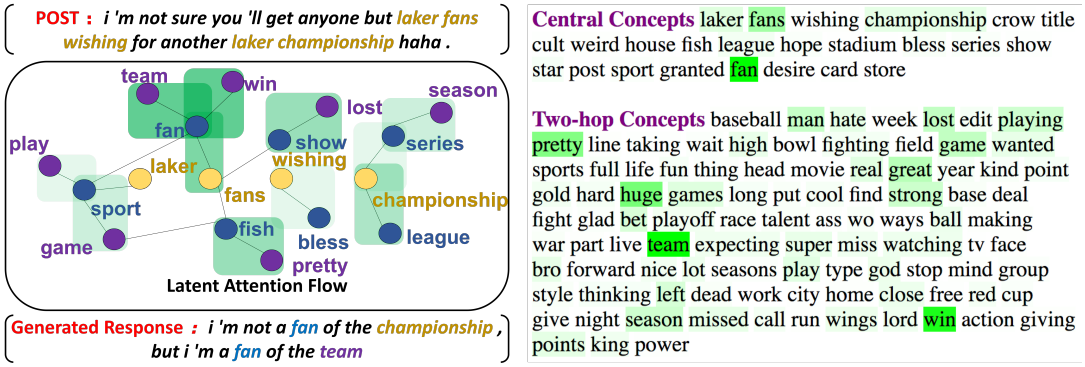
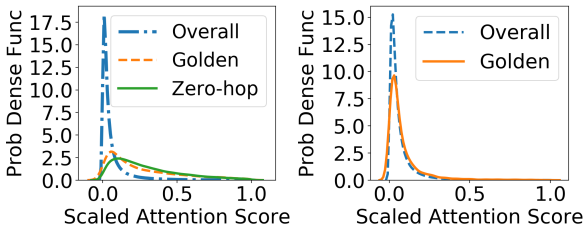


Figure 4: Case Study (Best viewed in color). **Left:** Attention flow in commonsense concept graph, where zero-hop concepts, one-hop concepts and two-hop concepts are highlighted. **Right:** Attention scores over all concepts. Darker green indicates higher attention scores.



(a) Central Concept. (b) Two-hop Concept.

Figure 5: Distribution of Attention Score. The distributions of Overall (all concepts of the certain part), Golden (concepts in the golden response) and Zero-hop (concepts appear in the post) are presented. The attention score is calculated by scaling the mean of attention scores of n step decoding.

5.4 Case Study

Some cases from three conversation models are listed in Table 6. Responses from CCM may repeat the same contents as it does not explicitly model the traverse in the concept space. For example, the responses from the first and third cases always repeat "I'm not sure". On the other hand, GPT-2 generates more fluent responses compared to CCM. Nevertheless, some cases from GPT-2 merely copy contents or concepts from the given post. For example, for the third case, GPT-2 (conv) mainly discusses the concept music.

In comparison, the generated responses from our ConceptFlow are more fluent and informative than those from both CCM and GPT-2. For example, in the third case, ConceptFlow brings associated concepts "sound" and "check" to the response generation, hopping from the grounded concepts "music" and "advice". Introducing these multi-hop concepts effectively improves the informativeness and diversity of generated responses.

Figure 4 presents a case study of ConceptFlow. The attention score β^{e_i} and γ^f are presented in the form of color intensity. The "championship" of zero-hop, "fan" of one-hop and "team" of two-hop receive more attention than others and are used to

#1 Post	**actually i stayed at the building right next to the lighthouse . i believe it was a boiler room . another group was already sleeping at the lighthouse .**
CCM	i 'm not sure if you 're joking , but i 'm not sure if you 're talking about the lighthouse .
GPT-2 (conv)	i 'm not sure if it was a blanket or a blanket .
ConceptFlow	i 'm pretty sure it was a room with a door . good luck , i 'm sure there are some good things in the house .
#2 Post	**at this point it may be more beneficial for him to keep playing as much as he can before the next season . trick his body into thinking he is still in the season .**
CCM	i think he 's just going to be a great player .
GPT-2 (conv)	i 'm sure he 'll be fine . he 's a great player , but he 's not a top 10 player .
ConceptFlow	he 's been playing the same game for the last two years . he 's been playing the same way .
#3 Post	**i 've known how to change the music , it does n't feel natural . thanks for the advice though .**
CCM	i 'm not sure if you 're joking , but i 'm not sure if you 're being sarcastic or not . i 'm not sure if you 're being sarcastic or not , but i 'm not sure if you 're being sarcastic or not . i 'm pretty sure it 's just a video .
GPT-2 (conv)	i 've been playing with a lot of different music and it 's always been the same .
ConceptFlow	i did n't even notice the natural sound . i 'll check it out !
#4 Post	**are milan and inter no longer top clubs given that they currently do n't have the finances to compete with the best ?**
CCM	milan is n't a top club , but they are n't top clubs . they are n't the only ones who are n't in the world .
GPT-2 (conv)	i think they are . they are a club that has been in the top 4 for a long time .
ConceptFlow	the best part of the team is the way they are , and we do n't want to play the best team in the country .
#5 Post	**no i 've never felt like i had friends whoops**
CCM	i 'm sorry , i feel you .
GPT-2 (conv)	i 've never felt like i had friends whoops . i 've never felt like i 've had friends whoops .
ConceptFlow	you 're gonna have to call your friend !

Table 6: Some Cases Generated by Different Models. Tokens from zero concepts, one-hop concepts and two-hop concepts are highlighted.

generate the response. The concept flow from "fans" to "fan" models the concept shift from user post to response. The concept flow from "fan" to "team" further describes the concept shift in response generation. In addition, some concepts, such as "win" and "pretty", share higher attention and may help to understand the one-hop concepts, and are filtered out when generating response by the gate σ^* according to the relevance with conversation topic.

5.5 Learned Attentions on Concepts

This experiment studies the learned attention of ConceptFlow on different groups of concepts. We consider the average attention score (β for central concepts and α (Appendix A.4) for two-hop concepts) from all decoding steps. The probability density of the attention is plotted in Figure 5.

Figure 5(a) shows the attention weights on central concepts. ConceptFlow effectively attends more on golden and zero-hop concepts, which include more useful information. The attention on two-hop concepts are shown in Figure 5(b). ConceptFlow attends slightly more on the Golden two-hop concepts than the rest two-hop ones, though the margin is smaller—the two-hop concepts are already filtered down to high-quality ones in the ConceptFlow (select) step.

6 Conclusion and Future Work

ConceptFlow models conversation structure explicitly as transitions in the latent concept space, in order to generate more informative and meaningful responses. Our experiments on Reddit conversations illustrate the advantages of ConceptFlow over previous conversational systems. Our studies confirm that ConceptFlow's advantages come from the high coverage latent concept flow, as well as its graph attention mechanism that effectively guides the flow to highly related concepts. Our human evaluation demonstrates that ConceptFlow generates more appropriate and informative responses while using much fewer parameters.

In future, we plan to explore how to combine knowledge with pre-trained language models, e.g. GPT-2, and how to effectively and efficiently introduce more concepts in generation models.

Acknowledgments

Houyu Zhang, Zhenghao Liu and Zhiyuan Liu is supported by the National Key Research and Development Program of China (No. 2018YFB1004503) and the National Natural Science Foundation of China (NSFC No. 61772302, 61532010). We thank Hongyan Wang, Shuo Wang, Kaitao Zhang, Si Sun, Huimin Chen, Xuancheng Huang, Zeyun Zhang, Zhenghao Liu and Houyu Zhang for human evaluations.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explorations*, pages 25–35.

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

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of NeurIPS*, pages 13042–13054.

Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A. Smith, and Mari Ostendorf. 2018. Sounding Board: A user-centric and content-driven social chatbot. In *Proceedings of NAACL*, pages 96–100.

Michel Galley, Chris Brockett, Xiang Gao, Bill Dolan, and Jianfeng Gao. 2018. End-to-end conversation modeling: Moving beyond chitchat.

Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. In *Proceedings of NAACL*, pages 1229–1238.

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 AAAI*, pages 5110–5117.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, pages 175–204.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of ACL*, pages 1631–1640.

Hiroaki Hayashi, Zecong Hu, Chenyan Xiong, and Graham Neubig. 2019. Latent relation language models. *arXiv preprint arXiv:1908.07690*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL*, pages 110–119.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of EMNLP*, pages 1192–1202.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *Proceedings of the ACL*, pages 1489–1498.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of ACL*, pages 5962–5971.

Yinong Long, Jianan Wang, Zhen Xu, Zongsheng Wang, Baoxun Wang, and Zhuoran Wang. 2017. A knowledge enhanced generative conversational service agent. In *DSTC6 Workshop*.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of ACL*, pages 845–854.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. In *Proceedings of Technical report, OpenAI*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Corbin Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, and Paul Bennett. 2020. Leading conversational search by suggesting useful questions. In *Proceedings of The Web Conference 2020*, pages 1160–1170.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of AAAI*, pages 3776–3784.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of ACL*, pages 1577–1586.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of AAAI*, pages 4444–4451.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of EMNLP*, pages 4231–4242.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. In *Proceedings of ACL*, pages 5624–5634.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Pavlos Vougiouklis, Jonathon Hare, and Elena Simperl. 2016. A neural network approach for knowledge-driven response generation. In *Proceedings of COLING*, pages 3370–3380.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of ACL*, pages 808–819.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of AAAI*, pages 3351–3357.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into conversation modeling via recall-gate LSTM. In *2017 International Joint Conference on Neural Networks, IJCNN*, pages 3506–3513.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of NeurIPS*, pages 1810–1820.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018a. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of IJCAI*, pages 4623–4629.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018b. A dataset for document grounded conversations. In *Proceedings of EMNLP*, pages 708–713.

Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.

A Appendices

Supplementary results of the overall performance and ablation study for multi-hop concepts are presented here. More details of Central Flow Encoding and Concept Selection are also shown.

A.1 Supplementary Results for Overall Experiments

This part presents more evaluation results of the overall performance of ConceptFlow from two aspects: relevance and novelty.

Table 7 shows supplementary results on Relevance between generated responses and golden responses. ConceptFlow outperforms other baselines with large margins among all evaluation metrics. Concept-PPL is the Perplexity that calculated by the code from previous work (Zhou et al., 2018a). Zhou et al. (2018a) calculates Perplexity by considering both words and entities. It is evident that more entities will lead to a better result in terms of Concept-PPL because the vocabulary size of entities is always smaller than word vocabulary size.

More results for model novelty evaluation are shown in Table 8. These supplementary results compare the generated response with the user post to measure the repeatability of the post and generated responses. A lower score indicates better performance because the repetitive and dull response will degenerate the model performance. ConceptFlow presents competitive performance with other baselines, which illustrate our model provides an informative response for users.

These supplementary results further confirm the effectiveness of ConceptFlow. Our model has the ability to generate the most relevant response and more informative response than other models.

A.2 Supplementary Results for Multi-hop Concepts

The quality of generated responses from four two-hop concept selection strategies is evaluated to further demonstrate the effectiveness of ConceptFlow.

We evaluate the relevance between generated responses and golden responses, as shown in Table 9. *Rand* outperforms *Base* on most evaluation metrics, which illustrates the quality of generated response can be improved with more concepts included. *Distract* outperforms *Rand* on all evaluation metrics, which indicates that concepts appearing in golden responses are meaningful and important for the conversation system to generate a more on-topic and informative response. On the other hand, *Full* outperforms *Distract* significantly, even though not all golden concepts are included. The better performance thrives from the underlying related concepts selected by our ConceptFlow (select). This experiment further demonstrates the effectiveness of our ConceptFlow to generate a better response.

A.3 Model Details of Central Flow Encoding

This part presents the details of our graph neural network to encode central concepts.

A multi-layer Graph Neural Network (GNN) (Sun et al., 2018) is used to encode concept $e_i \in G_{\text{central}}$ in central concept graph:

$$\vec{g}_{e_i} = \text{GNN}(\vec{e}_i, G_{\text{central}}, H), \tag{16}$$

where $\vec{e}_i$ is the concept embedding of e_i and H is the user utterance representation set.

The l-th layer representation $\vec{g}^l_{e_i}$ of concept e_i is calculated by a single-layer feed-forward network (FFN) over three states:

$$\vec{g}^l_{e_i} = \text{FFN}\left(\vec{g}^{l-1}_{e_i} \circ \vec{p}^{l-1} \circ \sum_r \sum_{e_j} f_r^{e_j \to e_i}\left(\vec{g}^{l-1}_{e_j}\right)\right), \tag{17}$$

where $\circ$ is concatenate operator. $\vec{g}^{l-1}_{e_j}$ is the concept e_j's representation of $(l-1)$-th layer. $\vec{p}^{l-1}$ is the user utterance representation of $(l-1)$-th layer.

The $(l-1)$-th layer user utterance representation is updated with the zero-hop concepts V^0:

$$\vec{p}^{l-1} = \text{FFN}(\sum_{e_i \in V^0} \vec{g}^{l-1}_{e_i}). \tag{18}$$

$f_r^{e_j \to e_i}(\vec{g}^{l-1}_{e_j})$ aggregates the concept semantics of relation r specific neighbor concept e_j. It uses attention $\alpha_r^{e_j}$ to control concept flow from e_i:

$$f_r^{e_j \to e_i}(\vec{e}^{l-1}_j) = \alpha_r^{e_j} \cdot \text{FFN}(\vec{r} \circ \vec{g}^{l-1}_{e_j}), \tag{19}$$

where $\circ$ is concatenate operator and $\vec{r}$ is the relation embedding of r. The attention weight $\alpha_r^{e_j}$ is computed over all concept e_i's neighbor concepts according to the relation weight score and the Page Rank score (Sun et al., 2018):

$$\alpha_r^{e_j} = \text{softmax}(\vec{r} \cdot \vec{p}^{l-1}) \cdot \text{PageRank}(e_j^{l-1}), \tag{20}$$

where $\text{PageRank}(e_j^{l-1})$ is the page rank score to control propagation of embeddings along paths starting from e_i (Sun et al., 2018) and $\vec{p}^{l-1}$ is the $(l-1)$-th layer user utterance representation.

The 0-th layer concept representation $\vec{e}_i^0$ for concept e_i is initialized with the pre-trained concept

Model	Bleu-1	Bleu-2	Bleu-3	Nist-1	Nist-2	Nist-3	Concept-PPL
Seq2Seq	0.1702	0.0579	0.0226	1.0230	1.0963	1.1056	-
MemNet	0.1741	0.0604	0.0246	1.0975	1.1847	1.1960	46.85
CopyNet	0.1589	0.0549	0.0226	0.9899	1.0664	1.0770	40.27
CCM	0.1413	0.0484	0.0192	0.8362	0.9000	0.9082	39.18
GPT-2 (lang)	0.1705	0.0486	0.0162	1.0231	1.0794	1.0840	-
GPT-2 (conv)	0.1765	0.0625	0.0262	1.0734	1.1623	1.1745	-
ConceptFlow	**0.2451**	**0.1047**	**0.0493**	**1.6137**	**1.7956**	**1.8265**	26.76

Table 7: More Metrics on Relevance of Generated Responses. The relevance is calculated between the generated response and the golden response. Concept-PPL is the method used for calculating Perplexity in CCM (Zhou et al., 2018a), which combines the distribution of both words and concepts together. The Concept-PPL is meaningless when utilizing different numbers of concepts (more concepts included, better Perplexity shows).

	Novelty w.r.t. Input(↓)						
Model	Bleu-1	Bleu-2	Bleu-3	Nist-1	Nist-2	Nist-3	Rouge-1
Seq2Seq	0.1855	0.0694	0.0292	1.2114	1.3169	1.3315	**0.1678**
MemNet	0.2240	0.1111	0.0648	1.6740	1.9594	2.0222	0.2216
CopyNet	0.2042	0.0991	0.056	1.5072	1.7482	1.7993	0.2104
CCM	**0.1667**	0.0741	0.0387	**1.1232**	**1.2782**	**1.3075**	0.1953
GPT-2 (lang)	0.2124	0.0908	0.0481	1.5105	1.7090	1.7410	0.1817
GPT-2 (conv)	0.2537	0.1498	0.1044	1.9562	2.4127	2.5277	0.2522
ConceptFlow	0.1850	**0.0685**	**0.0281**	1.3325	1.4600	1.4729	0.1777

Table 8: More Metrics on Novelty of Generated Responses. The novelty is calculated between the generated response and the user utterance, where lower means better.

Version	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Nist-1	Nist-2	Nist-3	Nist-4
Base	0.1705	0.0577	0.0223	0.0091	0.9962	1.0632	1.0714	1.0727
Rand	0.1722	0.0583	0.0226	0.0092	1.0046	1.0726	1.0810	1.0823
Distract	0.1734	0.0586	0.0230	0.0097	1.0304	1.0992	1.1081	1.1096
Full	**0.2265**	**0.0928**	**0.0417**	**0.0195**	**1.4550**	**1.6029**	**1.6266**	**1.6309**

Table 9: The Generation Quality of Different Outer Hop Concept Selectors. Both Bleu and Nist are used to calculate the relevance between generated responses and golden responses.

embedding $\vec{e}_i$ and the 0-th layer user utterance representation $\vec{p}^0$ is initialized with the m-th hidden state h_m from the user utterance representation set H. The GNN used in ConceptFlow establishes the central concept flow between concepts in the central concept graph using attentions.

A.4 Concept Selection

With the concept graph growing, the number of concepts is increased exponentially, which brings lots of noises. Thus, a selection strategy is needed to select high-relevance concepts from a large number of concepts. This part presents the details of our concept selection from ConceptFlow (select).

The concept selector aims to select top K related two-hop concepts based on the sum of attention scores for each time t over entire two-hop concepts:

$$\alpha_n = \sum_{t=1}^{n} \text{softmax}(\vec{s}_t \cdot \vec{e}_k), \tag{21}$$

where $\vec{s}_t$ is the t-th time decoder output representation and $\vec{e}_k$ denotes the concept e_k's embedding.

Then two-hop concepts are sorted according to the attention score α_n. In our settings, top 100 concepts are reserved to construct the two-hop concept graph V^2. Moreover, central concepts are all reserved because of the high correlation with the conversation topic and acceptable computation complexity. Both central concepts and selected two-hop concepts construct the concept graph G.