DVKCM: Knowledge-guided Conversation Generation with Dynamic Vocabulary

Xu Wang[†], Shuai Zhao[†]* , Bo Cheng[†], Jiale Han[†], Xiangsheng Wei[†], Yi Liang[†], Hao Yang[§]
[†]Beijing University of Posts and Telecommunications, Beijing, China
[§]2012 Labs, Huawei Technologies CO., LTD, Beijing, China
[†]{wxx,zhaoshuaiby,chengbo,hanjl,wxs,liangyi}@bupt.edu.cn, [§]yanghao30@huawei.com

***Abstract*—Knowledge-guided conversation models, whose inputs are current input sentence with its background knowledge, make the generation of responses more informative and meaningful. Existing methods assume that words in responses come from the vocabulary of the whole corpus. However, for specific input and knowledge, only a small vocabulary is useful in prediction and other words lead to uncorrelated noise. In this paper, we propose a Dynamic Vocabulary based Knowledge-guided Conversation Model (DVKCM). Inspired by dynamic vocabulary mechanism, DVKCM adopts the vocabulary construction module to allocate the sentence-level vocabulary which relates to the input sentence and background knowledge, and then only uses the small vocabulary to execute the decoding part. Through the sentence-level vocabulary mechanism, we reduce the generation of noise effectively. Experiments on both automatic and human evaluation verify the performance of our model compared with previous models. Moreover, we find that dynamic vocabulary can be applied to other conversation models to improve their performance.**

I. Introduction

Conversation task aims to generate corresponding responses for each input sentence. End-to-end generative networks [1], [2], as methods to solve the conversation task, have attracted more and more attention. A disadvantage of these models is that they are easy to generate responses that lack information, such as "I don't know" and "Yes". To solve the problem of lacking information, researchers [3]–[5] consider adding additional knowledge to each input sentence of the conversation model, so the conversation model can generate responses with background knowledge and enhance the informativeness of the responses. However, to ensure the coverage of target words in the response generation, these models predict the response words on a very large vocabulary, which guarantees the coverage of target words but increases the probability of generating incorrect words.

For example, in Tab. I, although the "ancestral home" for "Dingdang" is "Shandong Yantai", the response's "ancestral home" is "Canada Toronto", which is another people's "ancestral home". Such a mistake can be avoided if "Canada Toronto" is removed from the target vocabulary when the model generates the response.

Previously, some natural language generation tasks [6], [7] have contributed to reducing the size of the vocabulary by removing some words which are likely to be wrong before predicting. Each prediction is no longer based on the large vocabulary, which can reduce the probability of predicting wrong words. This method is called dynamic vocabulary. However, there is no exploration of how to apply dynamic vocabulary to knowledge-guided conversation generation.

TABLE I
Case demonstration of errors predicted by the model based on the whole vocabulary.

Background Knowledge	Response
"Dingdang", "Ancestral Home", "Shandong Yantai"	The film's star is Dingdang, I like him very much. His ancestral home is **Canada Toronto.**
"Kate.Balaude", "Ancestral Home", **"Canada Toronto"**	Yes, Kate·Balaude come from Canada Toronto.

To solve the above issues, we propose a dynamic vocabulary based knowledge-guided conversation model (DVKCM). For each input sentence, we adopt the vocabulary construction module to generate the corresponding sentence-level vocabulary, which is very small compared with the full target vocabulary but related to the input sentence and the background knowledge. For each sentence-level vocabulary, to get a fluent response, we extract the top N words in the frequency of the whole corpus. To get a knowledge-related response, we extract all the words of the current knowledge. To make the response relevant to the input sentence, we extract all the words of the input sentence. In the training, the vocabulary of the gold response is used. By utilizing the above words to construct a sentence-level vocabulary, for each input sentence, the target vocabulary size is reduced by removing unnecessary words. We adopt sentence-level vocabulary in the decoding module to generate the response.

Experiments are performed on LIC [8] and PERSONACHAT [9] datasets. We use automatic evaluation and human evaluation metrics and find that our method achieves significant and consistent improvement as compared to other baselines. Ablation experiments are used to verify the validity of dynamic vocabulary. We also find that our approach can be applied to other knowledge-guided conversation models to improve their performance. Our contributions are listed as follows:

- We propose the DVKCM model and apply the dynamic vocabulary to the knowledge-guided conversation. We

*Corresponding author

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

construct sentence-level dynamic vocabulary for each input sentence and reduce the probability of irrelevant words generation.
- Our model and baseline models are compared on LIC and PERSONACHAT datasets. We find that our model has improved by 1.4% and 5.4% on F1 metric, 1.6% and 4.5% on BLEU-1, respectively. We also find that applying the dynamic vocabulary to some existing models can improve their performance.

II. Related Work

The conversation model is based on the wide application of the Seq2Seq (sequence to sequence) model, and it predicts the response based on the input sentence. [10] uses machine translation's method by regarding sentences and responses as the source language and the target language respectively. [11] uses Seq2Seq with attention on this task. [12] presents a maximum mutual information objective function. [13] introduces attention models to generate long responses. [14] proposes hierarchical recurrent encoder-decoder networks to better represent the conversation context. As a shortcoming of these models, they don't use any external knowledge to guide the generation of responses, which may generate boring responses like "I don't know".

Adding external knowledge to the conversational models [15], [16], these models are able to generate more semantic responses. [17] uses the Memory Network to store external knowledge which generates the response that related to the knowledge, because the input sentence can interact with knowledge. [18] uses string matching to extract relevant facts to the current dialogue from a knowledge base. [19] extends Pointer-Generator Networks by allowing the decoder to hierarchically attend and copy from external knowledge in addition to the conversation context. [4] uses the posterior knowledge distribution to facilitate conversation generation. [20] obtains knowledge from unstructured texts using a convolutional neural network.

As a disadvantage of the above models, they assume that the words in responses come from the vocabulary of the whole corpus. However, for specific input and knowledge, only a small vocabulary is useful in prediction and other words lead to uncorrelated noise. [6] applies dynamic vocabulary to machine translation, and it uses ambiguity words, source words to construct dynamic vocabulary. [7] uses only one layer of the neural network to predict words in the dynamic vocabulary. For these two methods of dynamic vocabulary, good results have been achieved in their fields.

III. Model

In this section, we formalize the problem and then describe how to construct our model in detail.

A. Problem Formalization

Given a source sentence $X = x_1 x_2 ... x_n$, where x_t is the tth word in X and n is the length of X , with a collection of knowledge $\{K_1, K_2 ... K_m\}$ (each K_i can be a unstructured text or a structured triple), the goal of the conversation model is to predict the response $Y = y_1 y_2 ... y_l$, where y_i is the ith word in the response and l is the length of Y.

B. Architecture Overview

As Fig. 1 shows, we introduce our model from the following components:

- **Sentence and Knowledge Encoder:** The sentence encoder encodes the input sentence into a vector x. The knowledge encoder encodes each knowledge K_i into a vector k_i. If the target Y is available (at training step), it is also encoded into a vector y to guide knowledge selecting.
- **Knowledge Selecting:** With the representation of sentence x and knowledge $\{k_1, k_2 ... k_m\}$, we compute a similarity between x and $\{k_1, k_2 ... k_m\}$. Then we get the knowledge k_s with the highest similarity score and send it to the decoder.
- **Dynamic Vocabulary Construction:** We construct the dynamic vocabulary V_{dy} for current input sentence as the target vocabulary for predicting the response words.
- **Decoder:** With the knowledge k_s obtained from **Knowledge Selecting**, the encoded input sentence x and the target vocabulary V_{dy}, the decoder generates the corresponding response.

C. Sentence and Knowledge Encoder

Given the input sentence and knowledge (with target response at the training step), we use a bidirectional RNN with a gated recurrent unit (Bi-GRU) [21] to encode the $X = x_1 x_2 ... x_n$. The left-to-right gated recurrent unit (GRU) encodes the sentence from left to right, and obtains $\overrightarrow{h_i}$ for each x_i to record the information from its left side. Similarly, the right-to-left GRU encodes the information from its right side to get the $\overleftarrow{h_i}$. These two hidden states are concatenated to get the hidden state h_i of each x_i:

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] = [GRU(x_i, \overrightarrow{h}_{i-1}); GRU(x_i, \overleftarrow{h}_{i+1})], \quad (1)$$

where $[;]$ represents concatenation of vectors. We concatenate the last left-to-right GRU hidden state $\overrightarrow{h_n}$ and last right-to-left GRU hidden state $\overleftarrow{h_1}$ to represent the sentence, which is defined by $x = [\overrightarrow{h_n}; \overleftarrow{h_1}]$. x is used to select the most correlative knowledge at the **Knowledge Selecting** step.

Same as the sentence encoder, the knowledge encoder encodes each knowledge K_i by Bi-GRU, but it doesn't share the same parameters with the sentence encoder. We use the last hidden state of each direction to get the knowledge representation k_i. At the training step, the target response is used to guide the knowledge selecting. We encode the response Y into a vector y by using the same Bi-GRU with the knowledge encoder.

D. Knowledge Selecting

In this section, we aim to select the most relevant knowledge k_s which is most relevant to the input sentence , and use the

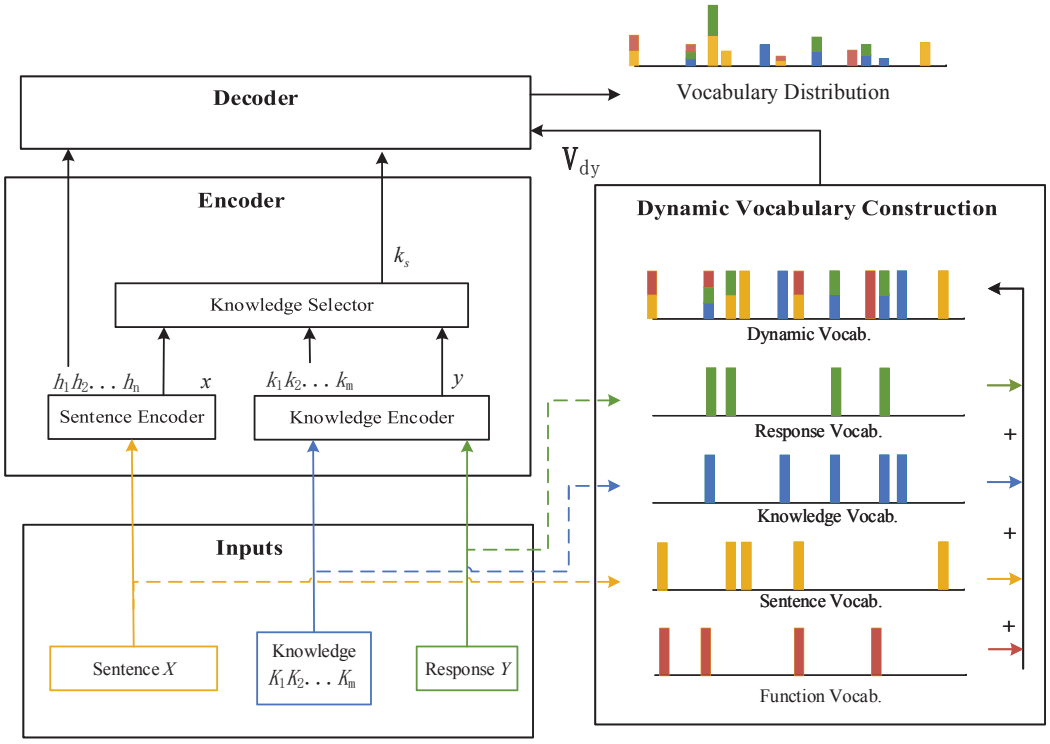

Fig. 1. The overall framework of DVKCM model and it shows the process of generating dynamic vocabulary. The model's inputs are the sentence and knowledge. At the training step, we also add the target response to the inputs. The right half part shows the structure of dynamic vocabulary, and each Vocab represents the words for themselves over the large vocabulary, then these Vocabs are combined to get dynamic vocabulary. Decoder predicts response based on the dynamic vocabulary.

k_s at the Decoder step. The model follows the method of [4] to select knowledge from the knowledge list.

At the training step, the target response y is available, so we compute a conditional probability distribution $p(k_i|y)$ over the knowledge list and use this distribution to get k_i, defined as:

$$p(k_i|y) = \frac{exp(k_i \cdot y)}{\sum_{j=1}^{m} exp(k_j \cdot y)}, \tag{2}$$

where m is the size of the knowledge list. We use dot operation to calculate the similarity between knowledge k_i and target response y. The highest value means that this knowledge is more prefer to be chosen. We use Gumbel-Softmax [22] to extract the knowledge from the knowledge list with the advantage that it can backpropagation.

At the predicting step, the target response y is not available, so we use the x to get k_i, defined as:

$$p(k_i|x) = \frac{exp(k_i \cdot x)}{\sum_{j=1}^{m} exp(k_j \cdot x)}. \tag{3}$$

We plan to transfer the ability of selecting knowledge through target response to input sentence, so that the input sentence can better select relevant knowledge in the testing stage. We define a KL-Loss to make the knowledge distribution between the target response y and the input sentence x similarly:

$$\mathcal{L}_{KL}(p(k_i|x)||p(k_i|y)) = -\sum_i p(k_i|x) ln \frac{p(k_i|x)}{p(k_i|y)}. \tag{4}$$

E. Dynamic Vocabulary Construction

In this section, we introduce the reason to build sentence-level dynamic vocabulary, and how to build dynamic vocabulary.

The motivation of dynamic vocabulary: As shown in Tab. I, given an input sentence, the conversation model needs to generate the response based on background knowledge. The response generated should be more relevant to the knowledge, so the words in the response should be more relevant to the knowledge. Investigating the previous models, we find that the response sometimes contains the wrong words. For example, with the knowledge, one person was born in "Hong Kong", but the response shows the person was born in "United States". We analyze the reason and find that it comes from another knowledge, exactly that another person was born in the "United States". Those issues can be solved by reducing the output vocabulary size, making the output vocabulary only related to the current input sentence and background knowledge, so the response may not be related to the useless words.

In most of the models, the predict vocabulary is always as large as the size of dataset, and it is regarded as the basic vocabulary. For each sentence, we reduce the target vocabulary as much as possible, building dynamic vocabulary for every input sentence. Dynamic vocabulary between different input sentences may not be the same because they do not have

the same background knowledge. At the experiment step, we show the effectiveness of dynamic vocabulary, with the smaller target vocabulary, the better performance.

How to build sentence-level dynamic vocabulary: We plan to build the target vocabulary of each input sentence for generating more meaningful responses. With the background knowledge, the response words have a higher probability of being generated by knowledge words, so we collect the words from the knowledge belong to one input sentence. We define a dictionary D_{k_i}, which represents all words in the knowledge k_i, then all knowledge words are merged:

$$V_k = \bigcup_{i=1}^{m} D_{k_i}, \tag{5}$$

where m is the size of the knowledge list. V_k represents all the knowledge words that belonging to only one input sentence.

Only the knowledge words are not enough. When people are talking in daily life, the words which guarantee grammatical correctness and fluent responses are still needed. These words are called function words. We discover the words, which make the final response fluently such as preposition, conjunctions, auxiliary words, have a higher probability ahead of the frequency words. Based on the words of the whole dataset, we get the top N frequency words. These frequency words are defined as V_d. Different from the V_k, for all the input sentences, V_d is the same.

To make the response relevant to the input sentence, We also collect the words of the input sentence, defined as:

$$V_m = \bigcup_{i=1}^{n} \{w_i\}, \tag{6}$$

where w_i represents the ith word in the sentence, and the length of the sentence is n.

At the training step, the target response is available. To make the model training more accurate, we collect the words of the target response, defined as:

$$V_r = \bigcup_{i=1}^{l} \{r_i\}, \tag{7}$$

where r_i represents the i-th word in the target response and the length of response is l.

Training step: We collect the V_k, V_d, V_m, V_r of an input sentence, defined as:

$$V_i = V_k \cup V_d \cup V_m \cup V_r, \tag{8}$$

where V_i is the dynamic vocabulary for only one input sentence. At the training step, we employ the mini-batch training strategy, and for simplicity, we use the union of all V_i in a batch:

$$V_{dy} = V_1 \cup V_2 \cup ... \cup V_b \tag{9}$$

where b is the batch size. We randomly shuffle the training sentences before each epoch. With the dynamic V_{dy} at each epoch, the model will lead to better coverage of parameters.

Predicting step: Without the target response, we use the

$$V_{dy} = V_k \cup V_d \cup V_m \tag{10}$$

as dynamic vocabulary for an input sentence, and we don't apply the mini-batch at predicting step.

F. Decoder

We use the selected knowledge k_s and the last predicted output y_{t-1} as the decoder input, defined as:

$$s_t = GRU([y_{t-1}; k_s], s_{t-1}), \tag{11}$$

where we concatenate y_{t-1} and k_s as the decoder step t's input, then we perform attention between s_t and the hidden states of sentence encoder $\{h_1, h_2 ... h_n\}$:

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b), \tag{12}$$

$$a_i^t = softmax(e_i^t), \tag{13}$$

where v, W_h, W_s, and b are learned parameters. a_i^t can be regraded as the attention on different sentence words. We perform a weighted sum of the hidden sentence states:

$$h_t^* = \sum_i a_i^t h_i. \tag{14}$$

The context vector h_t^* is the representation of the sentence at step t, regraded as what has read with s_t.

We concatenate the h_t^* and s_t to predict the output word $w_i \in V_{dy}$ at step t:

$$P_{w_i}^t = \frac{exp(s_{w_i}^t)}{\sum_{w_k \in V_{dy}} exp(s_{w_k}^t)}, \tag{15}$$

where s_{w_i} is defined as:

$$s_{w_i}^t = W_{w_i}[s_t, h_t^*] + b_{w_i} \quad \forall w_i \in V_{dy} \tag{16}$$

where W_{w_i} and b_{w_i} are learned parameters. $P_{w_i}^t$ is a probability distribution over all words in dynamic vocabulary V_{dy}.

G. Loss Function

Beside the KL-Loss, we use the NLL-loss to learn reduction of the response predicted by the model with the gold response, defined as:

$$\mathcal{L}_{NLL}(\theta) = -\frac{1}{l} \sum_{t=1}^{l} logp_\theta(y_t | y_{<t}, x, k_s), \tag{17}$$

where θ is the model parameters. l is the length of the response. $y_{<t}$ represents the previous generated words.

So the total loss for the model is defined as:

$$\mathcal{L}_{Loss}(\theta) = \mathcal{L}_{KL}(\theta) + \mathcal{L}_{NLL}(\theta). \tag{18}$$

IV. EXPERIMENTS

In this section, we describe our experiments, including two datasets we used, four baselines for comparison, training details, automatic evaluation performance, and human evaluation results. We change different N for frequency words and different combinations on dynamic vocabulary to prove the validity of dynamic vocabulary, then we show the performance for all baselines compared with our model, and perform ablation experiments to verify the effectiveness of each part of our model. We also apply our dynamic vocabulary method to Seq2Seq, MemoryNet and the latest existing models to prove that dynamic vocabulary can improve the performance of other models.

A. Dataset

We use a multi-turn Chinese dialogue dataset which is released by Baidu recently, named LIC competition dataset [8], and it is composed of knowledge grounded conversations in the movie domain. This dataset consists of thirty thousand sessions, about 120k dialogue sentences, of which 100k are training set, 10k are development set and 10k are test set. The training session includes two parts, which are knowledge and conversation. A pair of crowd workers generate each conversation, one of which plays the agent role and the other plays the user role. Offical has used word segmenter to segment sentences when releasing the dataset. More details can be found in [8].

Also, we perform our experiments on the PERSONACHAT dataset [9], which is collected from a real conversation between two crowd workers. These crowd workers play the role by the persona message that they are given, trying to know each other during the conversation. The training set contains about ten thousand dialogue turns, 160k sentences. In our experiments, we use the persona message as knowledge information.

B. Baselines

We compare our model with the following baselines:

- **S2SK**: For the knowledge-guided conversation, we use the Seq2Seq model with attention mechanism [23] as a baseline. We also add knowledge information into the input sentence to make the response can generate more informative responses.
- **MemoryNet**: The memory network [17], [24] uses several embedding metrics to write knowledge into slots, and reads knowledge by query vectors. In the conversation system, we use the input sentence as the query vector to get relevant knowledge, and the hops in this model is set to 2.
- **PointerNet**: The above models can only generate words in a fixed vocabulary, but PointerNet [25] also copies words from input sentences, which can solve the out of vocabulary problem effectively.
- **PostKS**: A model [4] which employs a novel knowledge selection mechanism where both prior and posterior distributions over knowledge are used to facilitate knowledge selection.

C. Training Details

For all the baselines and our model, we use the pre-trained BERT [26] for knowledge words embedding and conversation words embedding, and the embedding size is 768. The encoder and decoder have 1-layer of GRU [21] with 800 hidden state dimensions. The Adam optimizer [27] is used to update the gradient. Its initial learning rate is set to 0.00005, and gradient clipping is applied with a clip value of 5. We train the model in 20 epochs with a mini-batch size of 64. The size of N in sentence-level vocabulary is uncertain for different datasets. After a lot of experiments and evaluation on two datasets, we consider that it better lower than the size which is half of the different set of the whole corpus and all the knowledge words. We analysis it in Section IV-E.

D. Evaluation Metrics

we evaluate the performance of different models with the following metrics:

F1: following [4], we adopt $\mathbf{F1} = 2 \cdot \frac{\mathrm{Recall} \cdot \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}}$ [28] to measure the unigram score between the predicted response and golden response. Precision is calculated by dividing the number of the same words in the predicted response and the gold response by the length of the predicted response. Recall is calculated through dividing the number of the same words in the predicted response and the gold response by the length of the gold response.

BLEU-1/2: following [29], we use BLEU to match the words between the predicted response and the standard response. BLEU-1 judges the matching of unigrams, while BLEU-2 considers the matching of bigrams. The higher the BLEU score, the better the prediction. Because there are many words from the background knowledge in the standard response, BLEU is useful for matching keywords from the background knowledge.

Distinct-1/2: following [30], we calculate the ratios of distinct unigrams and bigrams in generated responses, and use the metrics to measure how diverse and informative the responses are.

Human: following [4], we find 5 volunteers (majoring in dialogue research) and 5 volunteers (other fields of research). For each response, we randomly select two volunteers from the two categories respectively to score. Volunteers are required to score a given response in the range of 0-2. 0 represents the content of the response is totally irrelevant. 1 represents the content of the response is relevant, but lack of information. 2 represents the content of the response is relevant and informative. We randomly extract 1000 predicted responses for each dataset, resulting in 2000 responses in total for human annotation. The Fleiss' kappa [31] coefficients are 0.61 and 0.65 on LIC and PERSONACHAT datasets.

E. Analysis of Different Dynamic Vocabulary Size

Because the number of words in the input sentence and the number of words in the knowledge are fixed and unchangeable,

TABLE II
PERFORMANCE OF DIFFERENT MODELS ON PERSONACHAT DATASET AND LIC COMPETITION DATASET. HUMAN EVALUATION IS ALSO USED TO EVALUATE THE PERFORMANCE OF THE MODELS (HUMAN).

Models	PERSONACHAT				LIC			
	F1	BLEU-1/2	Distinct-1/2	Human	F1	BLEU-1/2	Distinct-1/2	Human
S2SK	56.6	20.9/12.2	0.003/0.111	1.02	34.0	30.7/18.1	0.041/0.094	0.94
MemoryNet	56.5	22.1/12.8	0.002/0.004	1.12	33.8	32.3/18.7	0.036/0.071	1.10
PointerNet	56.9	20.1/12.0	0.012/0.049	1.20	34.5	18.0/10.9	0.027/0.104	1.22
PostKS (2019)	57.8	22.4/13.0	0.010/0.013	1.38	35.2	33.5/19.4	0.054/0.103	1.26
DVKCM-FIX	57.5	22.6/12.9	0.016/0.078	1.41	37.0	34.8/19.5	0.050/0.154	1.30
DVKCM	**58.3**	**23.7/14.0**	**0.023/0.098**	**1.50**	**39.9**	**36.8/22.7**	**0.081/0.212**	**1.39**

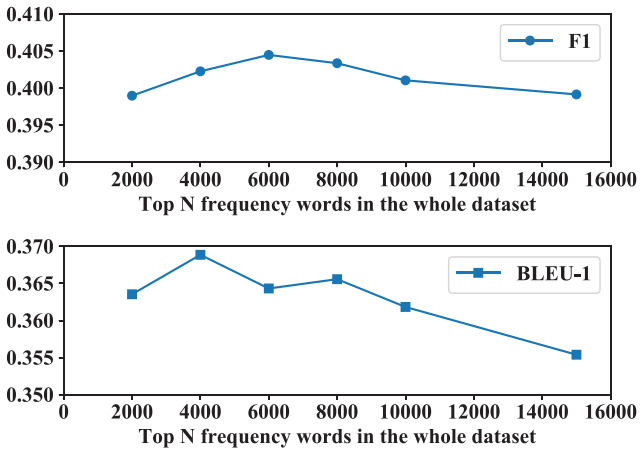

Fig. 2. After training different frequency words in LIC dataset, we find that the performances of both F1 and BLEU-1 by using smaller frequency words are improved. But the performance is reduced when the size of the frequency words N reduces from 6000 to 2000.

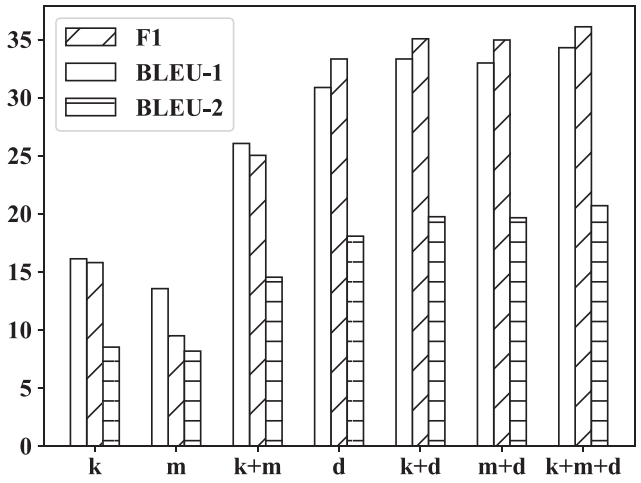

Fig. 3. For proving the validity of each part in dynamic vocabulary, we use different combinations of three data sources (V_k, V_d, V_m). For each combination, we default that using V_r at the training step.

we experiment with different top N frequency words on LIC dataset to display performance with changing N. We need to know in advance that the total vocabulary in the LIC dataset is 50013, and the total number of knowledge words is 43764. As Fig. 2 shows, we set N from 2000 to 15000. With F1 measurement, when N is 6000 the model gets the best score as 40.4, and with a larger or smaller N, the F1 score reduces. We analyze the responses produced by different N and find that with a larger N, the response contains some words that are not relevant to current input sentence or background knowledge, which makes noise. For a smaller N, the response losses some function words, so the model can't generate a smooth response. It is the same reason for BLEU-1 score. The experiment certificates, compared with a large vocabulary, the smaller target vocabulary can improve the performance, but it can't be very small.

F. Analysis of Different Combinations of Dynamic Vocabulary

For proving the validity of each part in dynamic vocabulary, we set frequency words N=6000 and use different combinations of V_k, V_d, V_m. At the training step, we default that V_r is used for each combination. At the predicting step, we remove V_r from the dynamic vocabulary. As Fig. 3 shows, when three parts are used as dynamic vocabulary separately, V_d gets the highest score, which proves that most words of the response are function words. As a disadvantage, there is no word of knowledge in V_d, which leads to the failure to generate knowledge-related information. When combining two vocabulary sources, the response can get a higher score, which proves not only function words but also knowledge words or input sentence words are needed when the model generates the response. The combination of three vocabulary source gets the highest score, which proves the importance of each part in dynamic vocabulary.

G. Overall Performance

We compute automatic evaluation metrics and human evaluation metrics for the baseline models in two datasets mentioned above. The results are shown in Tab. II.

As Tab. II shows, our model (DVKCM) has achieved the best result in all automatic metrics. For BLEU-1, our method is 1.6% higher than the second best one on the PERSONACHAT dataset and 4.5% higher on the LIC competition dataset, which proves the validity of our model. For the S2SK, MemoryNet and PointerNet models, their responses are generated based on the whole knowledge of each input sentence, instead of predicted based on the most related knowledge, and they also

TABLE III
WE PERFORMED ABLATION EXPERIMENTS ON TWO DATASETS, AND DVKCM IS OUR MODEL. WE REMOVE PART OF OUR MODEL AND GET THE CORRESPONDING PERFORMANCE ON TWO DATASETS TO VERIFY THE VALIDITY OF EACH PART OF THE MODEL.

Models	PERSONACHAT				LIC			
	F1	BLEU-1/2	Distinct-1/2	Human	F1	BLEU-1/2	Distinct-1/2	Human
DVKCM	**58.3**	**23.7/14.0**	**0.023/0.098**	**1.50**	**39.9**	**36.8/22.7**	**0.081/0.212**	**1.39**
No KS	57.8	23.0/13.4	0.010/0.011	1.39	35.9	32.3/19.4	0.041/0.099	1.32
No DV	57.6	21.0/12.2	0.012/0.012	1.40	34.2	30.5/18.3	0.042/0.110	1.27
No Embedding	57.0	22.8/12.3	0.009/0.014	1.35	36.1	33.4/21.7	0.041/0.109	1.20

TABLE IV
PERFORMANCE DEMONSTRATION OF ADDING DYNAMIC VOCABULARY TO OTHER MODELS. EXPERIMENTS SHOW THAT OTHER MODELS CAN IMPROVE THEIR PERFORMANCE BY ADDING DYNAMIC VOCABULARY.

Models	PERSONACHAT				LIC			
	F1	BLEU	Distinct	Human	F1	BLEU	Distinct	Human
S2SK	56.6	20.9/12.2	0.003/0.111	1.02	34.0	30.7/18.1	0.041/0.094	0.94
S2SK+DV	**57.8**	**23.0/13.4**	0.003/0.011	**1.12**	**36.0**	**32.3/19.4**	0.041/0.099	**1.05**
MemoryNet	56.5	22.1/12.8	0.002/0.004	1.12	33.8	32.3/18.7	0.036/0.071	1.10
MemoryNet+DV	**58.6**	**22.3** /12.5	0.001/0.003	**1.24**	**34.3**	**32.8/18.8**	**0.038/0.077**	**1.19**
PostKS (2019)	57.8	22.4/13.0	0.010/0.013	1.38	35.2	33.5/19.4	0.054/0.103	1.26
PostKS+DV	**58.0**	**23.4/13.7**	**0.020/0.088**	**1.45**	**39.1**	**34.3/20.1**	**0.060/0.140**	**1.37**

don't reduce the size of the target vocabulary. We think this is the main reason for their poor performance. For human evaluation, our model (DVKCM) can also achieve better performance, which proves the validity of our model.

In section 3.5, when collecting knowledge words, we only receive all knowledge words belonging to the current input sentence, which shown as DVKCM in Tab. II. To prove the effectiveness of this operation, we compare with DVKCM-FIX, which plus all the tokens in the knowledge base of the whole corpus. Although DVKCM-FIX collects all the knowledge words, the vocabulary size of DVKCM-FIX is still smaller because V_r, V_m and V_d reduce it, which leads to better performance compared with baselines. Because DVKCM-FIX collects many knowledge words unrelated to the current input sentence, the responses will lead to more noise than DVKCM, so the performance is lower than DVKCM.

H. Ablation Experiment

We perform ablation experiments to verify the effectiveness of each part of DVKCM model. We remove the following parts from the model separately:

- **No KS**: We remove the knowledge selection part from the model and average all knowledge vectors $\{k_1, k_2 ... k_m\}$ as selected knowledge k_s.
- **No DV**: We remove the dynamic vocabulary from the model and let the prediction of the model based on the whole corpus of words.
- **No Embedding**: We remove BERT embedding from the model and randomly initialize the input sentence and knowledge words vectors in the model.

The experimental results are shown in Tab. III. From the experiment, we can observe that:

(1) Without KS, our decoder part considers all knowledge of current input sentence at the same time, but not all of the knowledge is beneficial to the current response generation, so the noise is generated. The performance of the model is reduced on both datasets.

(2) Without DV, the generation of our responses is based on the whole corpus, which increases the probability of generating irrelevant words, such as words that are irrelevant to current knowledge but have similar features to current knowledge (see Tab. I). This proves that the prediction of the model doesn't need to be based on a large vocabulary. The coverage of words has increased, however, the number of noise words has also increased. If the prediction of the model is based on sentence-level vocabulary, the words in the vocabulary are more related to the response currently generated, which improves the performance while reducing the size of the target vocabulary.

I. Applying Dynamic Vocabulary to Other Models

We also prove that our dynamic vocabulary method can be applied to other models. We not only add the dynamic vocabulary method on S2SK and MemoryNet models, but also add dynamic vocabulary to the latest conversation model PostKS. For both datasets, at the training step, we use V_b as dynamic vocabulary and set V_d as 6000. At predicting step, we use V_p as the target dynamic vocabulary. With the Tab. IV, the result shows dynamic vocabulary can improve the performance on both datasets. We analyze the target responses for both datasets and find the most of the response words are coming from knowledge and function words. When we reduce the target vocabulary size and make sentence-level vocabulary for each input sentence, the responses can contain words that are related to the input sentences with higher probability.

J. Case Study

Tab. V shows an example on the LIC dataset. We can discover that S2SK model is unable to make good use of

TABLE V
SAMPLE RESPONSES FOR MODELS ON LIC DATASET. THE SOURCE IS THE INPUT SENTENCE. WE PRINT THE RESPONSES FOR BASELINES AND THE PROPOSED MODEL.

Source	Do you like watching feature films?
Knowledge	K1. Silent Night is a feature film. K2. Silent Night is released in 2002 year. K3. Inception is ascience fiction film. K4. Inception is released in 2010 year. K5. I like feature films. K6. Inception is interesting.
S2SK	I like films.
MemoryNet	I like Inception.
PointerNet	I like the feature films.
PostKS	I like watching feature films, like Graden.
DVKCM	I like watching feature films, like Silent Night.

knowledge information, resulting in less valuable responses. The MemoryNet model tries to reason to get the right knowledge, but apparently, it chooses the wrong knowledge. Although PointerNet has a mechanism to copy from source, it also gets boring responses, and does not make more use of knowledge information. Through the result of PostKS, we can see that this model well combines the information that some movies are feature films and wants to generate more meaningful response, but the result is the movie in other knowledge lists (Garden is a feature film). The model can't tell the difference between Garden and Silent Night. In DVKCM, by limiting the words to the knowledge belonging to the current source, other feature films' names in the vocabulary will be removed, and the model can output Silent Night very well. In general, DVKCM can generate more informative and input-related responses.

V. CONCLUSION

In this paper, we present a novel idea of applying dynamic vocabulary for the knowledge-guided conversationa model . Compared with the large vocabulary of the whole corpus, we employ dynamic vocabulary construction module to extract the smaller sentence-level vocabulary for each input sentence, which is related to the current input sentence and background knowledge, and only use the smaller vocabulary to execute the decoding part, which can reduce uncorrelated noise and improve the semantic relevance between the knowledge and response. Experiments on two datasets demonstrate the effectiveness of our model over all baselines. We also analyze each part of the dynamic vocabulary and verify the validity of each part. Dynamic vocabulary can be applied to other models, so the performance of other models can be better improved. A case study shows the advantages of the proposed model in solving some specific problems over all baselines.

VI. ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (No. 2018YFB1003804)

REFERENCES

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014.
[2] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *ACL*, 2015.
[3] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention," in *IJCAI*, 2018.
[4] R. Lian, M. Xie, F. Wang, J. Peng, and H. Wu, "Learning to select knowledge for response generation in dialog systems," *CoRR*, 2019.
[5] X. Zhang, "Mc\^2: Multi-perspective convolutional cube for conversational machine reading comprehension."
[6] H. Mi, Z. Wang, and A. Ittycheriah, "Vocabulary manipulation for neural machine translation," in *ACL*, 2016.
[7] Y. Wu, W. Wu, D. Yang, C. Xu, and Z. Li, "Neural response generation with dynamic vocabularies," in *AAAI*, 2018.
[8] W. Wu, Z. Guo, X. Zhou, H. Wu, X. Zhang, R. Lian, and H. Wang, "Proactive human-machine conversation with explicit conversation goals," *CoRR*, 2019.
[9] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "Personalizing dialogue agents: I have a dog, do you have pets too?" in *ACL*, 2018.
[10] A. Ritter, C. Cherry, and B. Dolan, "Unsupervised modeling of twitter conversations," in *NAACL*, 2010.
[11] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *ACL*, 2015.
[12] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," in *NAACL*, 2015.
[13] L. Shao, S. Gouws, D. Britz, A. Goldie, B. Strope, and R. Kurzweil, "Generating long and diverse responses with neural conversation models," *CoRR*, 2017.
[14] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *AAAI*, 2016.
[15] K. Lu, S. Zhang, and X. Chen, "Goal-oriented dialogue policy learning from failures," in *AAAI*, 2019, pp. 2596–2603.
[16] L. Xu, Q. Zhou, K. Gong, X. Liang, J. Tang, and L. Lin, "End-to-end knowledge-routed relational dialogue system for automatic diagnosis," in *AAAI*, 2019.
[17] M. Ghazvininejad, C. Brockett, M. Chang, B. Dolan, J. Gao, W. Yih, and M. Galley, "A knowledge-grounded neural conversation model," in *AAAI*, 2018.
[18] S. Liu, H. Chen, Z. Ren, Y. Feng, Q. Liu, and D. Yin, "Knowledge diffusion for neural dialogue generation," in *ACL*, 2018.
[19] S. Yavuz, A. Rastogi, G. Chao, and D. Hakkani-Tür, "Deepcopy: Grounded response generation with hierarchical pointer networks," *CoRR*, 2019.
[20] Y. Long, J. Wang, Z. Xu, Z. Wang, B. Wang, and Z. Wang, "A knowledge enhanced generative conversational service agent," in *DSTC6 Workshop*, 2017.
[21] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*, 2014.
[22] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*, 2017.
[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.
[24] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *NIPS*, 2015.
[25] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *ACL*, 2017.
[26] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, 2018.
[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
[28] D. M. W. Powers, "Visualization of tradeoff in evaluation: from precision-recall & PN to lift, ROC & BIRD," *CoRR*, 2015.
[29] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002.
[30] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," in *NAACL*, 2016.
[31] Fleiss and J. L., "Measuring nominal scale agreement among many raters." *Psychological Bulletin*, 1971.