Open Domain Response Generation Guided by Retrieved Conversations

Anonymous ACL submission

Abstract

Open domain response generation is the task of creating a response given a user query in any topics/domain. Limited by context and reference information, responses generated by current systems are often "bland" or generic. In this paper, we combine a response generation model with a retrieval system that searches for relevant utterances and responses, and extracts keywords from the retrieved results to guide the response generation. Our model uses a keyword extraction module to extract two types of keywords in an unsupervised fashion: (1) keywords in the *query* not found in the *retrieved utterances* (DIFFKEY), and (2) overlapping keywords among the *retrieved responses* (SIMKEY). Given these keywords, we use a two-stage transformer that first decides where to insert the keywords in the response, and then generates the full response given the location of the keywords. The keyword extraction module and the two-stage transformer are connected in a single network, and so our system is trained end-to-end. Experimental results on Cornell Movie-Dialog corpus, Douban and Weibo demonstrate that our model outperforms state-of-the-art systems in terms of ROUGE, relevance scores and human evaluation. Source code of our model is available at: ANONYMISED.

1 Introduction

Open domain response generation aims to develop conversational agents that can interact and communicate in a variety of topics (Sordoni et al., 2015; Li et al., 2016a; Vinyals and Le, 2015; Serban et al., 2015), and it differs from task-oriented dialogue systems which are designed to work towards a specific goal in a particular domain (e.g. finding a restaurant). There are generally two approaches for open domain response generation: retrieval and generation methods. Retrieval approaches search for answers from an existing corpus of dialogues to use them as response. Responses created by retrieval methods tend to be partially relevant and often do not directly address the queries, as the corpus is unlikely to have full coverage for all queries. Generation methods, on the other hand, are able to create fitting and natural responses but they tend to be short and generic (Li et al., 2016a). Combining both approaches would allow us to generate responses that are more diverse, interesting and relevant. Although there are a number of studies that explore combining both approaches, recent studies train the retrieval and generation component separately, with each component requiring different training data (Cai et al., 2019a,b; Tian et al., 2020; Gao et al., 2021). Existing studies also tend to use hidden representations as additional signals — e.g. that of skeleton words (Cai et al., 2019a) or abstract (Tian et al., 2019) — while our method uses keywords *directly*, and so is more interpretable as we can analyse specifically what keywords lead to a particular generated response.

Query (q)		What's your suggestion about holiday ? How about going to the seaside ?
Retrieved	**Utterance1** (u'_1)	What's your suggestion for tomorrow? How about outdoor sports?
	Utterance2 (u'_2)	Tomorrow is the weekend. I give your husband a suggestion to have a rest.
	DIFFKEY	holiday / going / to / seaside /
	Response1 (r'_1)	Leave me alone. I **just** want to **have** a **rest at home** and do some housework.
	Response2 (r'_2)	You are right. He **just at home** for one day last month and didn't have enough **rest**.
	SIMKEY	**just / have / rest / at / home**
Response (r)		I **just** want to **rest at home** on holiday , not go to the seaside .

Table 1: An example of query (q) and its retrieved utterances (u') and responses (r'). DIFFKEY is highlighted, while SIMKEY is **bolded**.

In this paper, we introduce a novel end-to-end system that combines retrieval and generation methods for open domain response generation. Given a query (q), our model first searches for relevant utterances (u') and responses (r') in a corpus of existing dialogues, and extracts two sets of keywords: DIFFKEY and SIMKEY. Using Table 1 as an example, DIFFKEY corresponds to words in the query (q) that are not found in the retrieved utterances (u'_1 and u'_2), e.g. *holiday* and *seaside*; while SIMKEY are overlapping words in the retrieved responses (r'_1 and r'_2), e.g. *rest* and *home*. Intuitively, DIFFKEY are keywords that are not captured by existing dialogues, and they are extracted so that the generated response would include them to improve its relevance to the query. SIMKEY can be interpreted as guiding keywords — the fact that they are frequently mentioned in the retrieved responses indicate that they are useful keywords to incorporate in the generated response. To capture word similarity beyond their surface forms, our system leverages transformer's attention mechanism to extract these keywords.

Given these keywords, we use a two-stage transformer to generate the final response. The first transformer takes the keywords as (unordered) input and decides where to insert them in the final response, creating a sentence where the predicted positions contain the keywords and other positions are masked tokens (e.g. "[mask] *just* [mask] [mask] *rest at home* [mask] *holiday* [mask] [mask] *go to* [mask] *seaside*"). The second transformer works like a text infilling model (Donahue et al., 2020), where it takes the masked sentence as input and "fill in the blanks" to generate the final response. The keyword extraction module and the two-stage transformer are connected in a single network, and as such the full model is trained end-to-end, requiring only a dialogue corpus as training data.

We conduct experiments on English and Chinese dialogue datasets and demonstrate that our system outperforms benchmark systems consistently across three datasets based on ROUGE, relevance scores and human evaluation, creating a new state-of-the-art for open domain response generation.

2 Related Work

Response generation can be broadly categorised into retrieval-based, generation-based and hybrid methods, which we review below.

Retrieval-based methods. Given an utterance, retrieval-based methods relies on matching algorithms to find the most relevant utterance in the conversation history to use its response as the output. The key is in developing matching algorithms that can measure textual relevance between two utterances (Hu et al., 2014). Early studies mainly focus on response selection for single-turn conversations (Wang et al., 2013). More recently, multi-turn retrieval-based conversation methods are also explored (Zhang et al., 2018).

Generation-based methods. By and large, generation methods use the sequence-to-sequence framework (Sutskever et al., 2014) for response generation. Attention (Bahdanau et al., 2015) and copy (Gu et al., 2016) mechanisms have been widely used to improve the performance of the original sequence-to-sequence framework. As generated responses tend to be generic, several methods are proposed to improve the diversity of the generated responses, e.g. by incorporating topic information (Wu et al., 2019b) or using latent variable models (Serban et al., 2017). Li et al. (2016c) experiment with reinforcement learning to further improve generation quality, and Liu et al. (2020) incorporate adversarial learning to reduce gender bias in its response generation. Xu et al. (2021) incorporate a keyword decoder to generate keywords based on the dialogue history and feed these keywords to the response generator.

Hybrid methods. Song et al. (2016) propose combining both generation and retrieval methods for generating responses. Pandey et al. (2018) retrieve similar conversations and weight them to guide generation. Miao et al. (2019); Cao et al. (2018); Wu et al. (2019a) develop retrieve-then-edit techniques for text generation improve the quality of the generated response. Cai et al. (2019b); Tian et al. (2020); Kazemnejad et al. (2020) treat the retrieval and generation as disjointed components and train them separately, but this means plenty of additional data is needed. Unlike other studies that largely focus on improving the generation component, Wu et al. (2020) propose improving the performance of the retrieval component through entity alignment. Compared to previous studies, our proposed method is unique in how it extracts keywords from the retrieved conversations (most studies only use the retrieved conversations as additional input without keyword extraction (Yang et al., 2019; Tian et al., 2019; Cai et al., 2019b)); the closest work to ours is Wu et al. (2018), while only using the top-1 retrieved utterance-response pairs and so do not consider overlapping keywords among the responses.

3 Model Architecture

3.1 Model Overview

The overall architecture of our system is presented in Figure 1, which consists of a retrieval model and a generation model. Given a query q, the retrieval model first retrieves top-N utterance-response pairs (u'_i, r'_i) from corpus $\mathcal{D}$. The generation model then extracts the keywords (DIFFKEY and SIMKEY) using the semantic alignment keyword extraction (SAKE) module, and the extracted keywords are fed to the two-stage transformer to generate the final response $\hat{r}$.

3.2 Retrieval Model

We use Lucene[1] to index and find top-K ($K = 2$) best utterances in corpus $\mathcal{D}$ based on Jaccard similarity:[2]

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are the bag-of-words of utterances.

3.3 Semantic Alignment Keyword Extraction (SAKE)

In SAKE, the goal is to extract DIFFKEY and SIMKEY, given the query q and retrieved utterance-response pairs (u'_i, r'_i). We first use a 1-layer transformer (Vaswani et al., 2017) to encode the text:

$$\boldsymbol{e} = \mathbf{TF}(S, P) \tag{1}$$

where S represents either q, u'_i, or r'_i; and P is the corresponding positional embeddings. $\boldsymbol{e} \in \mathbb{R}^{L \times D}$ is the contextualised word embeddings (where L denotes the length of sentence and D the embedding dimension).

To align two sentences, we adapt the alignment approach by Tsai et al. (2019), which was proposed to align sequences of different modalities (e.g. text with audio). We first provide a generic description of the alignment method, and come back to explain how to extract DIFFKEY and SIMKEY based on q, u'_i, and r'_i.

Single-source Alignment. Given two sentences, α and β, our goal is to align words in α to the words in β via query/key/value attention. After encoding the sentences with a transformer (Equation 1), we have $\boldsymbol{e}_\alpha \in \mathbb{R}^{L_\alpha \times D}$ and $\boldsymbol{e}_\beta \in \mathbb{R}^{L_\beta \times D}$. We then project the embeddings to query, key and value vectors, i.e. $Q_\alpha = e_\alpha W_Q$, $K_\beta = e_\beta W_K$ and $V_\beta = e_\beta W_V$, where W_Q, W_K and $W_V \in \mathbb{R}^{D \times H}$ and H is the projected dimension. $Y \in \mathbb{R}^{L_\alpha \times H}$, the semantic alignment output is computed as follows:

$$\begin{aligned} Y &= \mathbf{SA}(\boldsymbol{e}_\alpha, \boldsymbol{e}_\beta) \\ &= softmax\left(\frac{Q_\alpha K_\beta^T}{\sqrt{d_k}}\right) V_\beta \end{aligned}$$

Multi-source Alignment. Here we extend the alignment of one sentence to N sentences, i.e. α is one sentence but β is now a group of sentences $\beta = \{\beta_1, ..., \beta_N\}$, by aligning a pair of sentences iteratively and summing up their outputs:

$$Y^{[N]} = \sum_i^N \mathbf{SA}(\boldsymbol{e}_\alpha, \boldsymbol{e}_{\beta_i})$$

Figure 2 presents an illustration of the multi-source alignment method. Note that the key and value projection matrix (W_K and W_V) are shared by all $\{\beta_1, ..., \beta_N\}$.

Recall that DIFFKEY are keywords in query q that are not found in the retrieved utterance u'_i. In this case, $\alpha = q$, and $\beta = \{u'_i\}_{i=1}^N$. For SIMKEY, they are the overlapping words between the top-1 retrieved response r'_1 and other retrieved responses $\{r'_i\}_{i=2}^N$, and so $\alpha = r'_1$ and $\beta = \{r'_i\}_{i=2}^N$. Formally:

$$\begin{aligned} Y_{\text{DIFFKEY}}^{[N]} &= \sum_{i=1}^N \mathbf{SA}(\boldsymbol{e}_q, \boldsymbol{e}_{u'_i}) \\ Y_{\text{SIMKEY}}^{[N]} &= \sum_{i=2}^N \mathbf{SA}(\boldsymbol{e}_{r'_1}, \boldsymbol{e}_{r'_i}) \end{aligned} \tag{2}$$

To calculate the attention weights for each words in the query (q) for DIFFKEY or in the first response (r'_1) for SIMKEY, we compute $M = softmax\left(Y^{[\mathrm{N}]} \cdot \mathrm{W}_{\mathcal{S}} + \mathrm{b}_{\mathcal{S}}\right)$, where $Y^{[\mathrm{N}]}$ represents either $Y_{\text{DIFFKEY}}^{[N]}$ and $Y_{\text{SIMKEY}}^{[N]}$.

After obtaining the attention weights, we use them to weight the word embeddings as a soft approach to 'extract' the keywords.[3] Using Table 1 as an example for DIFFKEY, we would weight all

[1] https://lucene.apache.org/.

[2] We consider only utterances that have Jaccard similarity between 0.5–0.9.

[3] Strictly speaking they are subword embeddings, but for ease of interpretation we use the term "word embeddings" here.

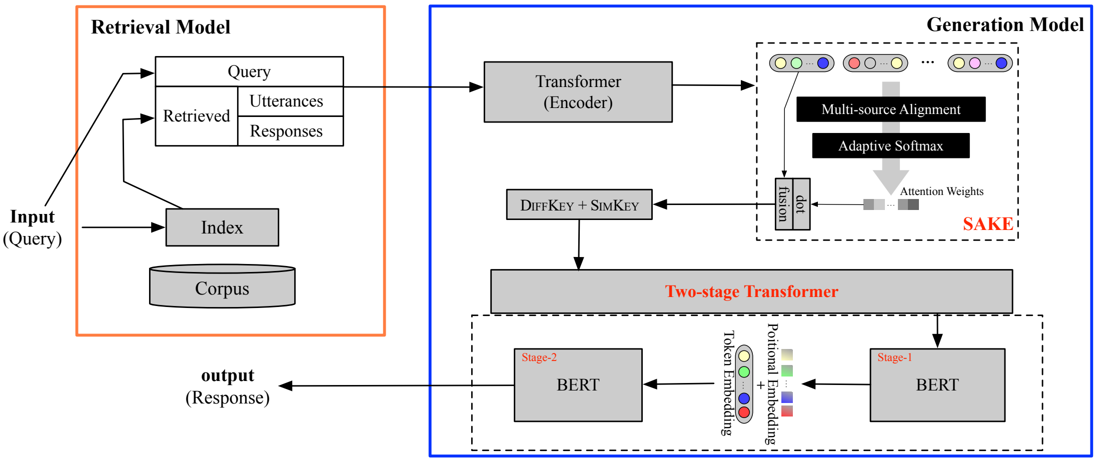

Figure 1: The architecture of proposed retrieval-generation model.

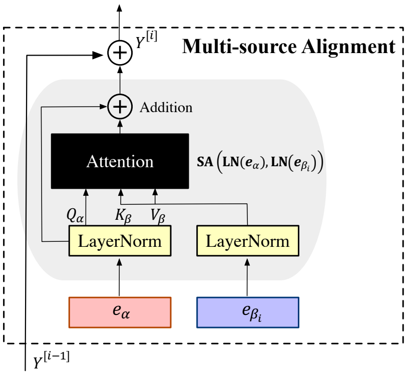

Figure 2: Multi-source alignment α (single sentence) and β (N sentences).

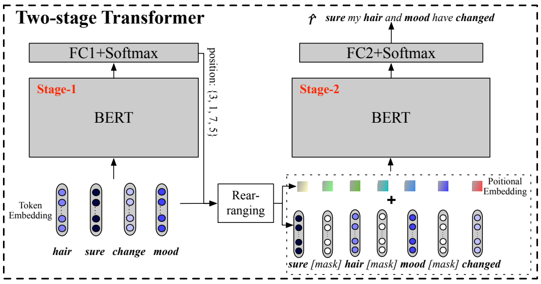

Figure 3: Input and output of the two-stage transformer at test time.

the word embeddings in the query text (q) "*What's your suggestion about holiday ? How about going to the seaside ?*", and words that receive low attention weights (such as *what*) would be effectively masked out. Note that at test time, we use an $argmin$ and $argmax$ operator to extract ηL words from q for DIFFKEY or r'_1 for SIMKEY respectively, where η is a scaling hyper-parameter that controls how many keywords to extract based on the original length of q or r'_1.

3.4 Two-stage Transformer

The DIFFKEY and SIMKEY produced by SAKE are keywords without ordering or positional information. To use them as input to guide the response generation,[4] we first use a BERT model (Devlin et al., 2019a) to predict their positions in the response, and then use another BERT to generate the final response. Note that this second BERT is *not* a fill-in-the-blanks model, as the final response is constructed by taking the highest probability word at every position. Also, during training we update only the second BERT (first BERT parameters are kept static).

Stage-1 Transformer. To imbue the keywords with positional information, we feed them to BERT to predict their positions:

$$
\begin{aligned}
g &= \mathbf{BERT}_1([\text{DIFFKEY}; \text{SIMKEY}]) \\
p_i &= softmax(\mathrm{W}_1\, g_i + \mathrm{b}_1) \\
q_i &= \sum_j p_{i,j} P_j
\end{aligned}
\tag{3}
$$

where P_j is the positional embedding for position j. Intuitively, for a word in DIFFKEY or SIMKEY, p_i represents its probability distribution over different positions, and q_i its weighted positional embedding.

Stage-2 Transformer. The second transformer is also a BERT, and similarly takes DIFFKEY and SIMKEY as input and its goal is to generate the final

[4] To clarify, during training DIFFKEY consists of the whole query (q) and SIMKEY the first response (r'_1) (noting that their embeddings are weighted by SAKE), but at test time DIFFKEY and SIMKEY contain only a subset of words (selected by the $argmin$ and $argmax$ operators respectively).

response. Here, we add the weighted positional embeddings (q) from the stage-1 transformer to the input:

$$\begin{aligned} h = &\ \mathbf{BERT}_2([\textsc{DiffKey}; \textsc{SimKey}] + \\ &\ Q([\textsc{DiffKey}; \textsc{SimKey}])) \\ \hat{r}_i = &\ softmax\Big(\mathrm{W}_2 h_i + \mathrm{b}_2\Big) \end{aligned}$$

where Q is a function that applies the weighted positional embeddings (Equation 3) to each input word, and $\hat{r}_i$ is the output word probability distribution, and the whole model is optimised end-to-end based on cross-entropy loss: $\mathcal{L} = -\sum_i \log P(\hat{r}_i)$.

At test time, instead of computing the weighted positional embeddings (Equation 3), we use $argmax$ to select the best position for each keyword, and introduce an additional step to re-arrange the keywords before feeding them to the stage-2 transformer; see Figure 3 for an illustration. We also truncate the generated response after <EOS> is produced at test time (i.e. all words to the right of <EOS> are discarded).

Intuitively, our stage-2 transformer can be interpreted as a text infilling model (Donahue et al., 2020), where it takes a masked sentence (that contains only important keywords) and learns how to "fill in the blanks" to create the response. As such, the generation process does not involve any decoding algorithms.

3.5 End-to-end Training

There are two components in the architecture that have non-differentiable operations that prevent end-to-end training: (1) keyword (DiffKey and SimKey) generation in SAKE; and (2) position prediction of keywords by stage-1 transformer. For (1), we do not use the attention weights to extract keywords; rather we use the weights to score the words in query and responses, and feed their weighted word embeddings to the two-stage transformer. For (2), we similarly do not commit to an argmax predicted position for the keywords — we instead compute a weighted positional embedding for each word based on the probability distribution over different positions. These tricks allows us to avoid using any non-differentiable operations, and so the network can be trained end-to-end using standard cross-entropy loss of ground truth response.

4 Experiments

4.1 Datasets and Evaluation Metrics

We use two Chinese datasets (Douban (Wu et al., 2018) and Weibo[5]) and an English dataset (Cornell Movie-Dialog corpus[6] (Danescu-Niculescu-Mizil and Lee, 2011)) to evaluate our response generation system. All three datasets consist of human conversations in the form of utterance and response pairs. For Douban, there are 19,623,374 original pairs. After removing pairs with high proportion of symbols (e.g. punctuations and emoticons) and very long sentences ($>$ 50 words), we retain 11,321,313 pairs. Weibo and Cornell each has 4,281,692 pairs and 430,579 respectively after undergoing the same preprocessing. We release the source code of our experiments and the Weibo dataset to facilitate replication and research.[7] To evaluate our model, we use the following metrics:

ROUGE (Lin, 2004): This metric measures the similarity between the generated response and ground truth response by evaluating their n-gram overlap.

Relevance: This also measures similarity, but uses cosine similarity of word embeddings instead to evaluate textual relevance. To aggregate the word embeddings of a sentence, we follow Liu et al. (2016) by taking the mean embeddings ("Average") and max-pooled embeddings (i.e. maximum value over words for each dimension; "Max") before computing the cosine similarity. We also compute another variant where we do not pool the word embeddings but greedily find the best matching words in the text pairs ("Greedy").

Diversity (Li et al., 2016b): This measures the repetitiveness of the generated response, and is computed based on the ratios of distinct unigrams (Dist-1) and bigrams (Dist-2). This metric does not use the ground truth response.

Human: Thirty annotators are invited to judge the generated responses of different systems on two aspects on a 4-point ordinal scale: fluency (F) and relevance (R). Details of the crowdsourcing experiments are provided in the Appendix.

[5] We collect the data by scraping microblog posts from https://weibo.com/

[6] https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html

[7] ANONYMISED.COM.

Types			Models	ROUGE			Relevance			Diversity		Human	
Pretrained	Rtv	Gen		R-1	R-2	R-L	Average	Max	Greedy	Dist-1	Dist-2	F	R
		✓	S2S+Attn	37.82	17.87	33.73	0.314	0.157	0.327	0.049	0.088	2.81	3.04
		✓	CVAE	41.89	20.86	39.49	0.339	0.182	0.357	0.076	0.145	3.14	3.13
		✓	KW+S2S	47.14	24.05	42.86	0.378	**0.214**	0.387	0.133	0.242	3.51	3.52
✓		✓	BERT	42.81	21.49	39.92	0.364	0.211	0.366	0.104	0.172	3.43	3.57
✓		✓	UniLM	44.24	23.07	40.27	0.373	0.205	0.371	0.121	0.189	3.47	3.38
✓		✓	GPT-3	47.11	23.92	41.77	0.378	0.209	0.389	0.137	0.211	3.58	3.56
	✓		Retrieval	30.81	13.87	27.33	0.252	0.131	0.269	0.103	**0.249**	3.81	3.20
	✓		Rtv+Rank	34.57	18.23	32.44	0.327	0.157	0.336	0.129	0.212	3.84	3.25
	✓	✓	Edit	45.81	21.99	43.01	0.369	0.198	0.376	0.112	0.207	3.44	3.51
	✓	✓	Reranker	45.16	21.04	42.71	0.357	0.194	0.380	0.094	0.182	3.47	3.51
	✓	✓	MemDistill	46.76	22.59	43.67	0.374	0.204	0.386	0.107	0.212	3.48	3.49
	✓	✓	SkelGen	46.69	21.13	43.09	0.369	0.211	0.375	0.116	0.231	3.43	3.51
✓	✓	✓	RAG	46.81	22.19	42.78	0.375	0.211	0.382	0.104	0.192	3.52	3.57
✓	✓	✓	Ours	**48.72**	**24.45**	**43.28**	**0.381**	**0.214**	**0.393**	0.121	0.231	3.55	**3.60**

Types			Models	ROUGE			Relevance			Diversity		Human	
Pretrained	Rtv	Gen		R-1	R-2	R-L	Average	Max	Greedy	Dist-1	Dist-2	F	R
		✓	S2S+Attn	36.77	20.14	35.07	0.346	0.179	0.358	0.026	0.084	3.12	3.08
		✓	CVAE	44.15	23.12	41.39	0.361	0.187	0.374	0.086	0.142	3.30	3.25
		✓	KW+S2S	50.32	25.23	48.01	0.390	0.241	0.392	0.163	0.222	3.45	3.49
✓		✓	BERT	48.64	27.71	48.73	0.391	0.262	0.397	0.147	0.285	3.58	3.57
✓		✓	UniLM	50.33	**30.19**	49.81	0.403	0.267	0.408	0.142	0.342	3.51	3.56
	✓		Retrieval	32.41	15.21	28.03	0.302	0.153	0.322	0.111	0.472	3.82	3.41
	✓		Rtv+Rank	38.29	18.17	35.14	0.361	0.174	0.378	0.167	**0.494**	**3.87**	3.40
	✓	✓	Edit	50.82	26.71	48.38	0.394	0.253	0.401	0.152	0.158	3.44	3.56
	✓	✓	Reranker	50.64	26.60	47.81	0.386	0.236	0.397	0.125	0.161	3.41	3.52
	✓	✓	MemDistill	50.82	26.71	48.38	0.394	0.253	0.401	0.152	0.158	3.47	3.55
	✓	✓	SkelGen	51.14	26.79	49.03	0.401	0.261	0.407	0.158	0.181	3.45	3.56
✓	✓	✓	RAG	52.02	27.69	50.31	0.409	0.276	0.413	0.152	0.271	3.50	3.57
✓	✓	✓	Ours	**52.43**	29.87	**50.41**	**0.416**	**0.296**	**0.421**	**0.203**	0.314	3.56	**3.64**

Types			Models	ROUGE			Relevance			Diversity		Human	
Pretrained	Rtv	Gen		R-1	R-2	R-L	Average	Max	Greedy	Dist-1	Dist-2	F	R
		✓	S2S+Attn	33.74	18.16	30.62	0.331	0.172	0.345	0.061	0.081	3.08	3.22
		✓	CVAE	42.32	21.83	38.78	0.362	0.189	0.368	0.076	0.201	3.31	3.29
		✓	KW+S2S	47.92	29.11	45.98	0.374	0.208	0.378	0.142	0.281	3.52	3.48
✓		✓	BERT	45.11	27.78	46.03	0.371	0.216	0.379	0.135	0.206	3.46	3.51
✓		✓	UniLM	48.76	31.89	47.09	0.382	0.223	0.394	**0.202**	0.364	3.51	3.49
	✓		Retrieval	30.19	14.88	28.36	0.274	0.142	0.289	0.131	**0.466**	3.75	3.38
	✓		Rtv+Rank	36.49	17.67	35.44	0.354	0.181	0.361	0.137	0.431	3.84	3.41
	✓	✓	Edit	48.27	29.81	46.53	0.393	0.216	0.385	0.134	0.189	3.48	3.50
	✓	✓	Reranker	48.21	29.91	47.37	0.378	0.206	0.383	0.128	0.237	3.46	3.46
	✓	✓	MemDistill	48.82	30.21	47.71	0.381	0.212	0.387	0.131	0.231	3.52	3.57
	✓	✓	SkelGen	49.18	30.31	48.72	0.384	0.220	0.393	0.102	0.251	3.54	3.56
✓	✓	✓	RAG	50.28	30.18	49.62	0.387	0.238	0.402	0.137	0.302	3.54	3.58
✓	✓	✓	Ours	**51.79**	**32.07**	**51.35**	**0.393**	**0.266**	**0.411**	0.188	0.297	3.59	**3.61**

Table 2: Results on Cornell (top), Weibo (middle) and Douban (bottom). **Boldfont** indicates optimal performance for a metric in a dataset. For model types, "Rtv" = retrieval method, "Gen"= generation method, and "Pretrained" = whether it uses pretrained models.

4.2 Baselines/Benchmarks

We compare our method against the following baseline/benchmark systems (which covers sequence-to-sequence, retrieval and hybrid methods):

S2S+Attn: Recurrent network-based sequence-to-sequence with attention model (Bahdanau et al., 2015).

CVAE: Conditional variational auto-encoder proposed by Zhao et al. (2017) to improve the diversity of generated responses.

KW+S2S: A generation-based model that uses a keyword encoder-decoder to generate keywords given the dialogue history, which are then concatenated with the dialogue history to generate the response (Xu et al., 2021). KW+S2S is trained end-to-end and the ground truth keywords are extracted using TF-IDF.

UniLM, **BERT**, **GPT-3**: These are pre-trained language models fine-tuned for response generation (Dong et al., 2019; Devlin et al., 2019b; Brown et al., 2020).[8] We only have English results for

[8] Note that for BERT we generate the full sentence by selecting the highest probability word in each position in one step and do not do left-to-right decoding (as it does not have a decoder).

GPT-3 as it does not support Chinese.

Retrieval: Baseline retrieval model that searches for the most relevant utterance (Lucene) and returns its response as the result.

Rtv+Rank: Retrieval method that searches for top-20 utterances based on **Retrieval**; two LSTM models are trained to encode pairs of utterances to select the most relevant response (Lowe et al., 2015).

Edit: Hybrid method that retrieves the most relevant utterance and computes two edit vectors to represent novel words in the query and the retrieved utterance to guide response generation (Wu et al., 2018). Note that this method retrieves only 1 relevant utterance, and as such does not capture similarity among relevant responses like our model.

Reranker: Hybrid method that has 2 components: (1) a generator that takes encoded representations of conversation context and retrieved responses as input to generate a response; and (2) a neural reranker that selects the best response among generated and retrieved responses (Yang et al., 2019).

MemDistill: Hybrid method that first clusters training query-response pairs and stores them in memory, and trains a generator to retrieve the most relevant query-response cluster from the memory to guide its generation (Tian et al., 2019). The method is unique in that it uses query-response cluster as a guide (rather than individual responses like our and other benchmark systems).

SkelGen: Hybrid transformer-based method that reranks a set of retrieved responses to select the best response as input for the generator to create a response (Cai et al., 2019b). The reranker is trained separately (using ground truth query-response pairs) to the generator, and the framework does not extract any keywords (the best response only serves as additional sequence to generator).

RAG: End-to-end hybrid model that uses BERT as the neural retriever and BART as the generator. RAG is designed as a general purpose retrieval-augmented generation system, and so uses Wikipedia as the knowledge source (Lewis et al., 2020).

4.3 Experimental Settings

We set word embedding dimension to 512, transformer hidden state dimension to 1024, and dropout rate to 0.3. We use a vocabulary size of 30,004 (30,000 words and 4 special symbols). For SAKE, the number of retrieved results $K = 2$, the projected dimension $H = 5$ and $\eta = 0.2$. We use a batch size of 512 and train for 30 epochs for all baselines and our model, and halve the learning rate when development performance worsens. We use the base model for BERT, and the uncased variant for English. All baseline/benchmark models use their default recommended hyper-parameter configuration.

Models	ROUGE	Relevance	Diversity	Human	
				F	R
Ours	45.07	0.357	0.243	3.59	3.61
−SimKey	37.12	0.289	0.136	-	-
−DiffKey	38.54	0.301	0.148	-	-
−Stage-1	43.21	0.347	0.191	-	-

Table 3: Ablation results where we measure the impact of DiffKey, SimKey and stage-1 transformer.

4.4 Results

Overall Experiments. Table 2 presents the full results, where the top, middle and bottom subtables are Cornell, Weibo and Douban results, respectively. Generally, we see that the hybrid systems are better models compared to pure generation and retrieval systems. Our model shows a strong performance: it substantially outperforms most baselines and benchmark systems in ROUGE and relevance scores across all 3 datasets, creating a new state-of-the-art.

In terms of human evaluation, the generated responses of our model are also more fluent and relevant than all generation and hybrid systems, although they are admittedly less fluent compared to retrieval systems (as their output are human-written responses). For diversity, we see a similar trend where retrieval systems tend to have an upper hand, although when compared to non-retrieval systems, our model outperforms all these systems by a comfortable margin.

Ablation Study. To study the influence of the individual components (e.g. the impact of the number retrieved results K, SAKE and two-stage transformer) in our system, we perform several ablation studies based on Douban (test set). All studies present the average scores of the different variants of ROUGE, relevance and diversity.

We assess the effectiveness of our keyword extraction module by removing either SimKey and DiffKey and present the results in Table 3. It ap-

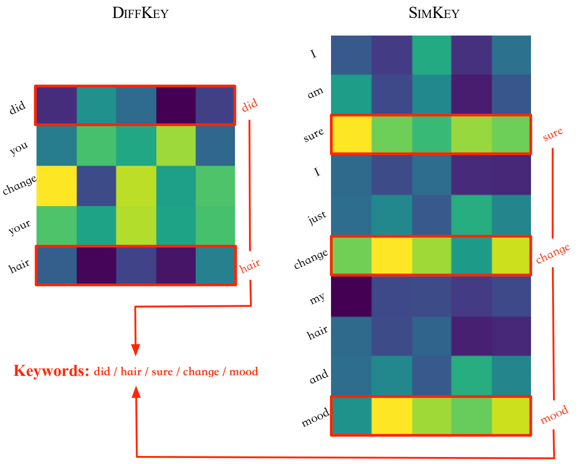

Figure 4: Multi-source alignment output produced by SAKE (Equation 2) for extracting DIFFKEY from the query (left) and SIMKEY from the first retrieved response (right). Darker colour indicates lower magnitude/strength.

Query		Did you change your hair
Retrieved	**Utterance1**	What happened about your hair
	Utterance2	Are you sure you won't change your mind
	Response1	I am sure I just change my hair and mood
	Response2	Sure, my mind change with mood
Keywords		did / hair / sure / change / mood
Generated Response		Sure my hair and mood have changed
Query		我 明天 想 出去 晒晒 I want to go out tomorrow
Retrieved	**Utterance1**	我 想 出去 晒晒 太阳 I want going out to sunbathe
	Utterance2	我 想 明天 出门 因为 在家 太 久 I'm going out tomorrow because I've been at home too long
	Response1	明天 是 个 好天气 It 's a fine day tomorrow
	Response2	天气 预报 说 明天 是 多云 没有 太阳 The weather forecast says it will be cloudy tomorrow and there will be no sun
Keywords		明天 / 出去 / 好天气 tomorrow / go out / fine weather
Generated Response		明天 会 是 个 适合 出去 的 好天气 Tomorrow is a fine day to go out

Table 4: Generated responses and retrieved conversations for two utterances from the Cornell Movie-Dialog corpus (top) and Douban (bottom).

pears that removing either keywords degrades the response substantially across all metrics, indicating the importance of both keywords. That said, SIMKEY seems to be marginally more effective than DIFFKEY in guiding the response generation.

We also test the impact of ordering the keywords by creating a variant where we remove the stage-1 transformer and feed DIFFKEY and SIMKEY to stage-2 transformer without ordering information (i.e. positional embeddings are not added to the input). Results are in the last row of Table 3. We see a dip in performance across all metrics, suggesting it is beneficial to decompose the generation task into a two-step process where we predict the order of the keywords before using them to drive the response generation.

Qualitative Analysis. We present the generated responses and the retrieved conversations by our system for two queries from the Cornell Movie-Dialog corpus (top) and Douban (bottom) Table 4. We can see the retrieved utterance and response pairs provide additional context for the query, and the generated responses are largely driven by the extracted keywords (SIMKEY and DIFFKEY).

To qualitatively understand the output of the SAKE through multi-source alignment, we present the alignment output $Y^{[N]}_{\text{DIFFKEY}}$ and $Y^{[N]}_{\text{SIMKEY}}$ (produced by Equation 2) in Figure 4. The query and top-2 retrieved utterance-response pairs are presented at the top of Table 4. Here we can see that words such as *did* and *hair* are selected as DIFFKEY from the query due to their low alignment strength with the retrieved utterances (see Table 4), while *sure*, *change* and *mood* are extracted as SIMKEY from the first retrieved response as these words are also mentioned in the second retrieved response.

Seeing function words such as *did* and *sure* are being selected as keywords (which seem counter-intuitive), we did another experiment where we use a stopword list to filter these words in SAKE. We found that the results worsen, and hypothesise that these words could be more important than they appear as we are working with response generation for casual conversation/dialogue.

5 Conclusion

We introduce an end-to-end response generation model that extracts keywords from retrieved conversations to guide the response generation. Our system combines the benefits of retrieval and generation methods, and utilises modern pre-trained language models and their attention mechanism for keyword extraction and response generation. We evaluate our system on 3 datasets over two languages (English and Chinese), and demonstrate that it outperforms benchmark systems in ROUGE, relevance scores and human evaluation, creating a new state-of-the-art.

Acknowledgments

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2019a. Skeleton-to-response: Dialogue generation guided by retrieval memory. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1219–1228, Minneapolis, Minnesota. Association for Computational Linguistics.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019b. Retrieval-guided dialogue response generation via a matching-to-generation framework. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1866–1875, Hong Kong, China. Association for Computational Linguistics.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pages 152–161.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4171–4186, Minneapolis, Minnesota.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2492–2501, Online.

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 33rd Conference on Neural Information Processing Systems (NeurIPS 2019).

Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2021. Meaningful answer generation of e-commerce question-answering. ACM Trans. Inf. Syst., 39(2).

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1631–1640, Berlin, Germany.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In Advances in Neural Information Processing Systems, volume 27, pages 2042–2050. Curran Associates, Inc.

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6010–6021, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Advances in Neural Information Processing Systems, volume 33, pages 9459–9474. Curran Associates, Inc.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Michel Galley, and Bill Dolan. 2016b. A diversity-promoting objective function for neural conversation models. In Proceedings of the 2016 Conference of the North American Chapter of the

Association for Computational Linguistics: Human Language Technologies, pages 110–119, San Diego, California.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016c. Deep reinforcement learning for dialogue generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 1192–1202. The Association for Computational Linguistics.

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

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2122–2132, Austin, Texas.

Haochen Liu, Wentao Wang, Yiqi Wang, Hui Liu, Zitao Liu, and Jiliang Tang. 2020. Mitigating gender bias for neural dialogue generation with adversarial learning. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 893–903. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 6834–6842.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pages 1329–1338.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, and Joelle Pineau. 2015. Building end-to-end dialogue systems using generative hierarchical neural network models. arXiv preprint arXiv:1507.04808.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, pages 3295–3301. AAAI Press.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. arXiv preprint arXiv:1610.07149.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 196–205, Denver, Colorado.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, page 31043112, Cambridge, MA, USA.

Zhiliang Tian, Wei Bi, Dongkyu Lee, Lanqing Xue, YIPING SONG, Xiaojiang Liu, and Nevin L. Zhang. 2020. Response-anticipated memory for on-demand knowledge integration in response generation. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Zhiliang Tian, Wei Bi, Xiaopeng Li, and Nevin L. Zhang. 2019. Learning to abstract for memory-augmented conversational response generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3816–3825, Florence, Italy. Association for Computational Linguistics.

Yao-Hung Hubert Tsai, J. Zico Bai, Louis-Philippe Morency, and Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30, pages 5998–6008. Curran Associates, Inc.

O. Vinyals and Q. Le. 2015. A neural conversational model. In Proceedings of ICML Deep Learning Workshop.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 935–945.

Sixing Wu, Ying Li, Dawei Zhang, and Zhonghai Wu. 2020. Improving knowledge-aware dialogue response generation by using human-written prototype dialogues. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1402–1411, Online. Association for Computational Linguistics.

Yu Wu, Furu Wei, Shaohan Huang, Zhoujun Li, and Ming Zhou. 2018. Response generation by context-aware prototype editing. arXiv preprint arXiv:1806.07042.

Yu Wu, Furu Wei, Shaohan Huang, and Ming Zhou. 2019a. Response generation by context-aware prototype editing. In The Thirty-Third AAAI Conference on Artificial Intelligence, 2019, Honolulu, Hawaii, USA, January 27, 2019, pages 7281–7288.

Yu Wu, Wei Wu, Chen Xing, Can Xu, Zhoujun Li, and Ming Zhou. 2019b. A sequential matching framework for multi-turn response selection in retrieval-based chatbots. Computational Linguistics, 45(1):163–197.

Heng-Da Xu, Xian-Ling Mao, Zewen Chi, Fanshu Sun, Jingjing Zhu, and Heyan Huang. 2021. Generating informative dialogue responses with keywords-guided networks. In Natural Language Processing and Chinese Computing, pages 179–192, Cham. Springer International Publishing.

Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W. Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A hybrid retrieval-generation neural conversation model. CIKM '19, page 13411350, New York, NY, USA. Association for Computing Machinery.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3740–3752, Santa Fe, New Mexico, USA.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 654–664, Canada.

Appendix

Anonymous ACL submission

1 Human Evaluation

We use the same methodology to collect human annotations for all three datasets. For each dataset, we randomly sample 200 generated dialogues (original query+generated response) and divide them into four batches (50 dialogues each batch). Sixteen native speakers (Chinese or English depending on the dataset) were invited to rate the generated responses on a 4-point scale;[1] Table 1 presents an example. The judges are broken into four groups, and each batch of dialogues is annotated by two groups of judges. For each dialogue, we have 2 ratings for each aspect (fluency or relevance) and we take the average as the final rating. Within a batch, if the ratings differ substantially between the two groups of judges, a third group of judges will be invited to annotate the batch. The judges do not have access to the ground-truth response, and see only the query and system-generated responses. Each worker is paid USD $0.15 for annotating a query. For fluency evaluation, the 4-point scale is described as follows:

1: *hard to read*;

2: *not quite fluent and has several grammatical errors*;

3: *fluent response with few errors*

4: *fluent response without errors.*

For relevancy:

1: *totally irrelevant*;

2: *marginally relevant*;

3: *somewhat relevant but not directly related to the query*

4: *relevant.*

Original Query	Would you be willing to relocate if required?	
	Generated Response	
	Fluency	Relevance
1	location I course not.	I like apple best.
2	I of course for it.	Shanghai is the most international city in China.
3	No preference for I.	Shanghai is good for me.
4	Of course, I have no preference for location.	Of course, I have no preference for location.
原始问句	今天外面的天气不错，我们出去吃饭好不好?	
	生成回答	
	流畅度	相关性
1	这好你是。	你今天看起来真漂亮。
2	我这好想好啊。	今天天气真好！
3	好啊，我这好想。	我今天想吃面。
4	好，我正好想出去吃。	没问题我们出去吃。

Table 1: An example of scoring criteria.

[1] We use the Tencent online document platform for conducting the crowdsourcing experiments: https://docs.qq.com/