Fine-Grained Sentence Functions for Short-Text Conversation

Wei Bi[1], Jun Gao[1,2], Xiaojiang Liu[1], Shuming Shi[1]
[1]Tencent AI Lab, Shenzhen, China
[2]School of Computer Science and Technology, Soochow University, Suzhou, China
{victoriabi, jamgao, kieranliu, shumingshi}@tencent.com

Abstract

Sentence function is an important linguistic feature referring to a user's purpose in uttering a specific sentence. The use of sentence function has shown promising results to improve the performance of conversation models. However, there is no large conversation dataset annotated with sentence functions. In this work, we collect a new Short-Text Conversation dataset with manually annotated SEntence FUNctions (STC-Sefun). Classification models are trained on this dataset to (i) recognize the sentence function of new data in a large corpus of short-text conversations; (ii) estimate a proper sentence function of the response given a test query. We later train conversation models conditioned on the sentence functions, including information retrieval-based and neural generative models. Experimental results demonstrate that the use of sentence functions can help improve the quality of the returned responses.

1 Introduction

The ability to model and detect the purpose of a user is essential when we build a dialogue system or chatbot that can have coherent conversations with humans. Existing research has analyzed various factors indicating the conversational purpose such as emotions (Prendinger and Ishizuka, 2005; Zhou et al., 2018; Shi and Yu, 2018), topics (Xing et al., 2017; Wang et al., 2017), dialogue acts (Liscombe et al., 2005; Higashinaka et al., 2014; Zhao et al., 2017) and so on. This work describes an effort to understand conversations, especially short-text conversations (Shang et al., 2015), in terms of *sentence function*. Sentence function is an important linguistic feature referring to a user's purpose in uttering a specific sentence (Rozakis, 2003; Ke et al., 2018). There are four major sentence functions: *Interrogative*, *Declarative*, *Imperative* and *Exclamatory* (Rozakis, 2003). Sentences with different sentence functions generally have different structures of the entire text including word orders, syntactic patterns and other aspects (Akmajian, 1984; Yule, 2016).

Some work has investigated the use of sentence function in conversation models. For example, Li et al. (2016) propose to output interrogative and imperative responses to avoid stalemates. Ke et al. (2018) incorporate a given sentence function as a controllable variable into the conditional variational autoencoder (CVAE), which can encourage the model to generate a response compatible with the given sentence function.

Considering the importance of sentence function in conversation modeling, it is surprised to find that no large conversation dataset has been annotated with sentence functions. In Ke et al. (2018), they only labeled a small dataset with 2,000 query-response pairs. Sentence function classifiers are trained and tested on this dataset and the best model only achieves an accuracy of 78%, which is unsatisfactory to serve as an annotation model to automatically assign sentence functions for unlabeled conversation data.

The goal of this work is two fold. On one hand, we create a new Short-Text Conversation dataset with manually annotated SEntence FUNctions (STC-Sefun), in which each sentence segment in the query-response pairs is labeled with its sentence functions. Besides the four major sentence functions, we get inspired by the dialogue act tag set (Stolcke et al., 2000) and further decompose each of them into fine-grained sentence functions according to their different purposes indicated in conversations. For example, *Interrogative* is divided into *Wh-style Interrogative*, *Yes-no Interrogative* and other six types. As shown in the first two examples in Figure 1, queries expressed in a *Yes-no Interrogative* sentence and a *Wh-style*

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3984–3993
Florence, Italy, July 28 - August 2, 2019.

Query	聊点啥呢
	What shall we talk about
Sentence Function	Interrogative: Wh-style Interrogative
Response	你想聊聊股市吗?
	Do you want to talk about stock market?
Sentence Function	Interrogative: Wh-style Interrogative
Query	你喜欢唱歌吗
	Do you like singing?
Sentence Function	Interrogative: Yes-no Interrogative
Response	特喜欢听歌, 唱歌
	Especially like listening to songs and singing
Sentence Function	Declarative: Positive Declarative
Query	游戏进不去
	I can’t get into the game
Sentence Function	Declarative: Negative Declarative
Response	是卡的进不去?
	Is it because of the slow network
Sentence Function	Interrogative: Yes-no Interrogative

Figure 1: Query-response pairs in the STC-SeFun dataset. The manually annotated level-1 and level-2 sentence functions are separated by the colon.

Interrogative sentence have divergent word patterns and their corresponding responses are also far different. We have twenty fine-grained sentence functions in total. And we annotate each query-response pair with this two-level sentence function label set.

On the other hand, we investigate how to output a response with the consideration of sentence function to improve the performance of conversation models. We decompose this task into two sub-tasks. First, we perform two sentence function classification tasks on the STC-SeFun dataset to: (1) determine the sentence functions of unlabeled queries and responses in a large corpus of short-text conversations, and (2) predict a target response sentence function for a given test query. Second, we explore various conversation models utilizing sentence function in different manners. These models include information retrieval-based and neural generative models, which are built upon the large automatically annotated corpus and tested with the predicted target sentence function. We show experimentally that the sentence function classifiers on the two classification tasks achieve sufficiently reliable performance, and sentence function can help improve the relevance and informativeness of the returned responses in different types of conversation models. All our code and datasets are available at `https://ai.tencent.com/ailab/nlp/dialogue.html`.

2 Related Work

Research on dialogue systems or chatbots have studied to control the output responses with different signals to improve user satisfaction of the interaction. Various methods consider emotions or topics as the controlling signals. For example, Martinovski and Traum (2003) find that many conversation breakdowns could be avoided if the chatbot can recognize the emotional state of the user and give different responses accordingly. Prendinger and Ishizuka (2005) show that an empathetic responding scheme can contribute to a more positive perception of the interaction. Xing et al. (2017) observe that users often associate an utterance in the conversation with concepts in certain topics, and a response following a relevant topic could make the user more engaged in continuing the conversation. The above studies, involving the control of emotions or topics, often affects a few words in the whole returned response, such as *smiling* for the happy emotion, *moisturizing* for the skincare topic. Different from them, sentence function adjusts the global structure of the entire response, including changing word order and word patterns (Ke et al., 2018).

Modeling dialogue acts such as *statement, question* and *backchannel*, in conversation models has also attracted many researchers’ attention. Higashinaka et al. (2014) identify dialogue acts of utterances, which later contribute to the selection of appropriate responses. Zhao et al. (2017) utilize dialogue acts as the knowledge guided attributes in the CVAE for response generation. Sentence function is similar to dialogue act in that they both indicate the communicative purpose of a sentence in conversation. Moreover, our fine-grained sentence function types are in many ways inspired from the dialogue act tag set (Stolcke et al., 2000) designed for the Switchboard corpus (Godfrey and Holliman, 1997), which consists of human-human conversational telephone speech. However, the conversations in the Switchboard corpus is multi-round, multi-party and aligned with speech signals. In our work, we target for the single-round non-task-oriented short-text conversation data collected from social media platforms. Thus we remove tags that cannot be determined in our setting, i.e. those needed to be determined in multiple rounds, involved multiple parties, or related to speech signals. Then we merge the remaining tags that have no big difference in their sentence

Sentence Function	Frequent Patterns		Sentence Examples	
	Chinese	English	Chinese	English
Wh-style Interrogative	x在哪y?	Where does x y?	周末在哪过啊	Where do you spend your weekend
	谁是x?	Who is x?	谁是天蝎座	Who is a Scorpio
Yes-no question	x是在y吗?	Is x y?	你是在云南吗?	Are you in Yunnan?
	x是指y吗?	Does x y?	你是指昨天的篮球比赛吗?	Do you mean the basketball match yesterday?
Alternative question	x还是y	x or y	狮子和白羊真配还是假配?	Leo and Aries go together or not?
	x y哪个	x y which	香蕉和苹果哪个卖得比较好?	Which sells better, bananas or apples?

Figure 2: Frequent word patterns of three level-2 *Interrogative* sentence functions. x and y are variables to represent the content words. The underlined words in the sentences are those corresponding to the word patterns.

word orders or patterns into one level-2 label. As a result, we have twenty fine-grained sentence functions in our annotation task.

Most existing conversation models can be categorized into two types: the information retrieval (IR)-based models and the neural generative models. The IR-based models search for the most similar query in the repository and directly copy its corresponding response as the result (Ji et al., 2014; Hu et al., 2014). Meanwhile, the generative models learn the mapping from the input query to the output response in an end-to-end manner (Xing et al., 2017; Zhou et al., 2018; Zhao et al., 2017). Specifically, Ke et al. (2018) propose a generative model to deal with the compatibility of controlling sentence function and generating informative content. In our experiments, we use this model as one of the compared methods to analyze the performance on our large conversation corpus.

3 Data Collection

In this section, we describe the annotation process of the STC-SeFun dataset: (1) how we collect high-quality conversation pairs to be annotated; (2) how we annotate sentence functions for these conversation pairs.

3.1 Conversation Data Preparation

We collect a huge number of raw query-response pairs from popular Chinese social media platforms, including Tieba, Zhidao, Douban and Weibo. We first pre-process the raw data to filter out pairs that contain dirty words and other sensitive content. Next, four annotators from a commercial annotation company are recruited to select out high-quality pairs, in which the responses should be not only relevant to the query but also informative or interesting. Each response is assigned to two different annotators and annotated independently. We then select out 100k query-response pairs that both annotators consider high-quality for the sentence function annotation task.

3.2 Sentence Function Annotation

For a given query-response pair, we first segment the sequence of the query/response by its punctuation. Then we hire three annotators from the same commercial annotation company to annotate sentence functions of each sequence segment.

We design a two-level sentence function label set for annotation. For the level-1 sentence functions, we have the typical four labels: *Declarative (DE)*, *Interrogative (IN)*, *Imperative (IM)* and *Exclamatory (EX)*. We further categorize them into the level-2 fine-grained labels due to their different purposes in the conversations. For example, *IN* is further divided into *Wh-style IN*, *Yes-no IN* and other six IN types due to the fact that word patterns in different IN labels differ significantly. Figur 2 illustrates some frequent patterns for these fine-grained IN sentence functions. In total, we have twenty level-2 sentence function labels, which are shown in Table 1. The explanation of each sentence function is provided in Appendix.

For each conversation pair, each query/response segment is annotated with both the level-1 and level-2 sentence function labels. Figure 1 shows three annotated examples. The detailed annotation process consists of two stages:

- We ask three annotators to select at most one level-1 label and two level-2 labels for each sentence segment. During annotation, the annotator should consider the query and response jointly to assign the sentence functions.
- After all annotators finish labeling the same conversation pair, we re-annotate it as follows: (1) if all three annotators assign the same labels, this data pair is not re-annotated; (2) if labels from all annotators have no overlap or the conversation pair has a sentence segment with no annotated label

Sentence Function	Query	Response
Declarative (DE)		
Positive DE	49,223 (48%)	67,540 (57%)
Negative DE	9,241(9%)	18,428(16%)
DE with IN words	887(.9%)	2,660(2%)
Double-negative DE	40(<.1%)	99(.1%)
Other types of DE	2,675(3%)	5,218(4%)
Interrogative(IN)		
Wh-style IN	23,385(23%)	7,652(7%)
Yes-no IN	6,469(6%)	4,046(3%)
A-not-A IN	6,456(6%)	1,055(.9%)
Alternative IN	789(.8%)	279(.2%)
IN with tag question	170(.2%)	271(.2%)
Rhetorical	42(<.1%)	417(.4%)
IN with backchannel	0(0%)	345(.3%)
IN with open question	227(.2%)	11(<.1%)
Imperative(IM)		
IM with request	2,073(2%)	358(.3%)
IM with dissuade	86(<.1%)	58(<.1%)
IM with command	7(<.1%)	4(<.1%)
IM with forbidden	4(<.1%)	2(<.1%)
Exclamatory(EX)		
EX without tone words	241(.2%)	3,948(3%)
EX with interjections	364(.4%)	1,958(2%)
EX with greetings	167(.2%)	285(.2%)
Total sentences	95,898	95,898
Total sentence segments	103,138	117,714

Table 1: Statistics of the SeFun dataset.

at all, we ignore this pair; (3) we present all labels together with the majority-voting results back to the annotator who gives the inconsistent label and ask him/her to check if he/she agrees with the majority-voting results. If this annotator agrees with the majority-voting results, we store this conversation pair with the confirmed results, otherwise we ignore it.

As a result, we have 95,898 conversation pairs remaining and Table 1 shows some statistics.

4 Sentence Function Classification

We are given a query with query segments $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, its annotated sentence function labels $[d_{x,1}, d_{x,2}, \ldots, d_{x,n}]$, its paired response with response segments $[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m]$ and the response sentence function labels $[d_{y,1}, d_{y,2}, \ldots, d_{y,m}]$. However, n and m are 1 for most conversation pairs in our STC-SeFun dataset, which involve short queries and responses generally. We perform two classification tasks:

• Given a query/response sentence segment, we design a model to predict its own sentence function. This model helps us to automatically annotate sentence functions for a large number of unlabeled conversation pairs, which can be used to build conversation models considering with sentence function. We refer this as the **Classification-for-Modeling** task.

• Given a query and its sentence functions, we aim to predict a proper response sentence function for this query. This model allows us to select a target sentence function, which will be considered when we decide the output response from the conversation model for a test query. We refer this as the **Classification-for-Testing** task.

4.1 Classification-for-Modeling Task

Training setup: For this task, we train and test the models using different data setups: (1) train with annotated query segments only and test on query segments only; (2) train with annotated response segments only and test on response segments only; (3) mix annotated query and response segments together for training and test on query and response segments respectively.

Network structures: Our model is a two-level classifier performed in a hierarchical fashion. For the first level, we employ an encoder to obtain a sentence representation $\mathbf{v}_x$ for each input sentence segment, which can be used as high-level features for sentence classification. Specifically, the encoder is followed by a fully-connected (FC) layer and a softmax layer to estimate the probability of each level-1 sentence function. Mathematically, the probability distribution of the level-1 sentence function labels $p(d_x^{l1}|\mathbf{x})$ is computed as follows:

$$\begin{aligned} \mathbf{v}_x &= \text{Encoder}(\mathbf{x}), && (1) \\ p(d_x^{l1}|\mathbf{x}) &= \text{Softmax}(\text{FC}(\mathbf{v}_x)). && (2) \end{aligned}$$

To compute the probability of each level-2 sentence function, we first use an embedding vector $\mathbf{e}_x^{l1}$ to represent the level-1 sentence function d_x^{l1} estimated by Eq. 2 from a converged model. Then $\mathbf{e}_x^{l1}$ is added to the sentence vector $\mathbf{v}_x$ to compute the probability distribution of the level-2 sentence functions as follows:

$$p(d_x^{l2}|\mathbf{x}, d_x^{l1}) = \text{Softmax}(\text{FC}(\mathbf{v}_x + \mathbf{e}_x^{l1})). \quad (3)$$

For encoders, two common implementations are attempted. The first is a CNN-based encoder commonly used for text classification tasks (Kim, 2014). The second is a RNN-based encoder which encodes a sequence using a bidirectional GRU.

Implementation details: The dimension of all hidden vectors is 1024. All parameters are initialized by sampling from a uniform distribution

$[-0.1, 0.1]$. The batch size is 128. We use the Adam optimizer with the learning rate 0.0001 and gradient clipping at 5.

Constructing the STC-Auto-SeFun dataset: The classification results are shown and analyzed in Section 6.2. Based on the obtained results, we conclude that the estimated sentence functions by our trained models are highly reliable. Thus we apply the best model to automatically annotating the sentence functions for queries and responses on a large unlabeled data corpus, which contains over 700 million conversation pairs crawled using similar steps in Section 3.1. This dataset, namely the STC-Auto-SeFun dataset, is later used to build conversation models discussed in Section 5.

4.2 Classification-for-Testing Task

In this task, we aim to predict the response sentence function given a query and its sentence functions, i.e. $p(d_y|[\mathbf{x}], [d_x])$. In this work, we focus on the estimation of a single response sentence function and leave the discussion of multiple response sentence functions in future work.

We employ an encoder for the input query to obtain the query representation, and an embedding layer followed by a BOW layer to obtain the query sentence function representation. Next, we add these two representations up and feed them into a FC layer followed by a softmax output layer to obtain the probabilistic distribution. The parameter setting is the same as in the Classification-for-Modeling task.

5 Conversation Models with Sentence Function

In the following, we show how to utilize the sentence functions in both the IR-based models and the generative models. We use the STC-Auto-SeFun as the retrieval/training corpus in the IR-based/generative models. And we focus to predict the response with one target sentence function.

5.1 IR-based Models

IR baseline: We adopt a simple IR model by first finding the most similar query in the retrieval corpus, then utilizing its response as the result. Similarity is measured by the Jaccard index between two bags of words.

Re-ranked model: We now present a method which demonstrates that sentence function can be used to improve the retrieval-based models. We first obtain a set of candidate responses for the IR baseline. Candidate responses are re-ranked based on whether the candidate is assigned with the target sentence function d_y^*, which is predicted by the Classification-for-Testing model for the current query $\mathbf{x}$. We use the Classification-for-Modeling classifier to predict whether a candidate response is tagged with the target sentence function. If the predicted label is not the target sentence function, this candidate response's score will be penalized with a weight by the Classification-for-Modeling classifier's output probability scaled by a constant.

Specifically, assume the IR baseline $f(x, R) \rightarrow \{s_1, s_2, \ldots, s_{|R|}\}$, where R is the set of candidate responses and the IR baseline outputs a ranked list of scores $\{s_1, s_2, \ldots, s_{|R|}\}$ corresponding to the candidate responses $R = \{r_1, r_2, \ldots, r_{|R|}\}$. We then run the Classification-for-Modeling classifier to predict the sentence function d_{r_i} for each candidate response r_i with the probability p_{r_i}. A penalty weight is computed for each candidate as:

$$s_i^{penalty} = \begin{cases} 0 & \text{if } y_{r_i} = d_y^*, \\ p_{r_i} & \text{otherwise.} \end{cases} \tag{4}$$

That is, if the candidate response r_i is assigned with the target sentence function, $s_i^{penalty}$ is zero. If it is tagged with any other sentence function, $s_i^{penalty}$ is the highest probability of the incorrect sentence function, i.e. p_{r_i}.

The re-ranking score is then computed as:

$$s_i^{re-rank} = s_i - \lambda(s_1 - s_k)s_i^{penalty}, \tag{5}$$

and the candidate responses are sorted according to their $s_i^{re-rank}$'s. Here, hyper-parameters λ and k are used to control sentence function's influence in re-ranking. If the top candidate has a penalty weight of 1.0, then with $\lambda = 1$, it will be moved to the k'th position in the ranking list. Whereas, $\lambda = 0$ corresponds to no re-ranking.

5.2 Generative Models

Seq2seq baseline: We use a one-layer bi-directional GRU for the encoder, and a one-layer GRU for the decoder with soft attention mechanism (Bahdanau et al., 2015). Beam search is applied in testing.

C-Seq2seq (Ficler and Goldberg, 2017): We modify the Seq2seq baseline by adding the sentence function embedding as another input at each decoding position.

Method	level-1 sentence functions			level-2 sentence functions		
	Accuracy	Macro-F1	Micro-F1	Accuracy	Macro-F1	Micro-F1
CNN-encoder (separated)	97.5	87.6	97.5	86.2	52.0	86.2
RNN-encoder (separated)	**97.6**	90.9	**97.6**	87.2	**65.8**	87.1
CNN-encoder (joint)	97.4	87.3	97.3	86.5	51.8	86.4
RNN-encoder (joint)	**97.6**	**91.2**	97.5	**87.6**	64.2	**87.6**

Table 2: Results (%) on 10,000 test query segments on the Classification-for-Modeling task.

Method	level-1			level-2		
	accuracy	macro-F1	micro-F1	accuracy	macro-F1	micro-F1
CNN-encoder (separated)	95.2	76.6	95.1	79.0	43.3	79.0
RNN-encoder (separated)	95.5	85.0	95.5	80.0	**54.2**	80.0
CNN-encoder (joint)	95.2	78.4	95.2	80.3	46.0	80.2
RNN-encoder (joint)	**95.8**	**85.9**	**95.8**	**80.6**	53.4	**80.6**

Table 3: Results (%) on 10,000 test response segments on the Classification-for-Modeling task.

KgCVAE (Zhao et al., 2017): The basic CVAE introduces a latent variable z to capture the latent distribution over valid responses and optimizes the variational lower bound of the conditional distribution $p(\mathbf{y}, z|\mathbf{x})$. To further incorporate the knowledge-guided features l, the KgCVAE assumes that the generation of $\mathbf{y}$ depends on z, $\mathbf{x}$ and l, and l relies on $\mathbf{x}$ and z. The variational lower bound is then revised to consider l jointly. Here, we use the response sentence function d_y of each conversation pair as the knowledge-guided features.

SeFun-CVAE (Ke et al., 2018): This model is specifically designed to deal with the compatibility of the response sentence function d_y and informative content in generation. It optimizes the variational lower bound of $p(\mathbf{y}, z|\mathbf{x}, d_y)$, where z is a latent variable assumed to be able to capture the sentence function of $\mathbf{y}$. Thus a discriminator is added to constrain that the encoding information from z can well realize its corresponding sentence function d_y. The decoder is also revised to generate words among three types: function-related, topic and ordinary words.

5.3 Implementation Details

For the re-ranked IR-based model, we collect the top-20 candidates for re-ranking. We set $\lambda = 1$ and $k = 20$. For all generative models, we use a vocabulary of 50,000 words (a mixture of Chinese words and characters), which covers 99.98% of words in the STC-Auto-SeFun dataset. All other words are replaced with <UNK>. The network parameter setting is identical to the classification task. During testing, we use beam search with a beam size of 5.

6 Experiments on Sentence Function Classification

6.1 Metrics

We report *Accuracy* (the percentage of samples with corrected sentence functions), *Macro-F1* (the F1 score that weights equally all classes) and *Micro-F1* (the F1 score that weights equally all test samples).

6.2 Classification-for-Modeling Task

We randomly sample 10,000 query and response segments respectively from the STC-SeFun dataset for testing. Results on test query and response segments are summarized in Table 2 and 3 respectively. As stated in Section 4.1, we train different models with query/response data only (denoted as separated), as well as query and response data jointly (denoted as joint) and try two sentence encoders: CNN-based and RNN-based. From the results, we can see that:

- The RNN-based encoder is better than the CNN-based encoder on both test query and response segments consistently on all metrics;
- There is very little performance difference between the separated and joint training data setting under the same network structure;
- Accuracy of all models, even for level-2 sentence functions, are much higher than 78% reported in Ke et al. (2018), in which the classifier is for 4-class classification and tested on 250 sentences only. It means our models are more re-

Method	level-1			level-2		
	Accuracy	Macro-F1	Micro-F1	Accuracy	Macro-F1	Micro-F1
CNN-encoder (without query SeFun)	81.2	15.1	81.1	55.7	23.5	55.7
RNN-encoder (without query SeFun)	77.9	**30.3**	77.9	**65.6**	25.8	65.5
CNN-encoder (with query SeFun)	81.2	17.4	81.1	**65.6**	21.1	65.6
RNN-encoder (with query SeFun)	**81.3**	28.5	**81.5**	65.5	**25.7**	**65.7**

Table 4: Results(%) on 5,000 test queries on the Classification-for-Testing task.

liable to assign sentence function labels to unlabeled conversation pairs;

- Macro-F1 scores, especially for level-2 sentence functions, are much lower than Micro-F1 scores. This indicates our models may not perform well on all sentence functions. However, considering that our conversation data are naturally imbalanced and dominated by a few labels, which can be observed from the statistics in Table 1, it is sufficient to discriminate between top classes.

Based on the above analysis, we consider the RNN-encoder(joint) as the best model for this task, and apply it for the construction of the STC-Auto-SeFun dataset and the conversation models.

6.3 Classification-for-Testing Task

We utilize classifiers for this task to estimate the proper response sentence function given the query with/without the query sentence functions. We also implement the RNN-based and CNN-based encoders for the query representation for comparison. Table 4 shows the results on 5,000 test queries by comparing the predicted response sentence function with its annotated groundtrue response sentence function. We can observe that:

- Encoding query sentence functions is useful to improve the performance for both CNN-based and RNN-based encoders.
- The RNN-based encoder again outperforms the CNN-based encoder, except for very few cases.
- Performance on this task decreases significantly compared to the Classification-to-Modeling task. This is because that this task is more subjective and there may be no definite response sentence function to reply to a given query.

Note that in previous work about estimating the next dialogue act from a 33 dialogue act tag set given the context with its dialogue acts (Higashinaka et al., 2014), the models achieve about 28% on Accuracy. Comparing with them, we consider our model has sufficient ability to choose a reasonable target response sentence function for a given test query. Here, we choose to use the RNN-encoder (with query SeFun) in the testing of the conversation models discussed in the next section.

7 Experiments on Conversation Models

7.1 Metrics

Since automatic metrics for open-domain conversations may not be consistent with human perceptions (Liu et al., 2016), we hire three annotators from a commercial annotation company to evaluate the top-1 responses on 200 sampled test queries in four aspects: *Fluency* (whether a response is grammatical), *Relevance* (whether a response is a relevant reply to its query), *Informativeness* (whether the response provides meaningful information via some specific words relevant to the query) and *Accuracy* (whether the response is coherent with the target sentence function). Each aspect is graded independently in five grades from 0 (totally unacceptable) to 5 (excellent). We further normalize the average scores over all samples into $[0, 1]$.

7.2 IR-based Models

Results are shown in Table 5. We can make the following observations:

- The re-ranked models achieve higher accuracy on the target response sentence function than the IR baselines, which means our designed re-ranking score function is effective.
- For both sentence function levels, the re-ranked IR models perform better than the IR baselines on all metrics. This means that considering a proper sentence function into the IR-based models is useful to help select high-quality responses.

Method	Flue	Rele	Info	Accu
IR baseline (level1)	63.4	68.4	61.5	34.3
Re-ranked IR (level1)	**69.6**	**74.4**	**77.2**	**50.5**
IR baseline (level2)	63.0	68.2	61.6	25.0
Re-ranked IR (level2)	**68.0**	**73.4**	**75.3**	**38.6**

Table 5: Results(%) on the IR-based models.

Method	Flue	Rele	Info	Accu
Seq2seq(level1)	55.4	61.5	49.3	32.0
C-Seq2seq(level1)	55.9	**65.0**	**51.6**	33.0
KgCVAE(level1)	**57.6**	62.5	51.4	29.0
SeFun-CVAE(level1)	57.1	63.5	50.9	**34.5**
Seq2seq(level2)	53.0	62.3	48.9	35.0
C-Seq2seq(level2)	**58.9**	**64.7**	**50.9**	**37.2**
KgCVAE(level2)	56.5	63.2	49.4	33.7
SeFun-CVAE(level2)	56.9	63.7	50.2	36.7

Table 6: Results(%) of the generative models.

7.3 Generative Models

Results are shown in Table 6. We have the following observations:

- For level1 sentence functions, the C-Seq2seq achieves the highest scores on relevance and informativeness, the second best score on accuracy and the third score on fluency. For level2 sentence functions, it performs the best on all metrics. Thus, we consider the C-Seq2seq has the best performance on our test set overall.
- The Seq2seq baseline is inferior to C-Seq2seq and SeFun-CVAE on all metrics. This indicates that with proper use of sentence function, the generative conversation models can effectively improve their performance.
- The KgCVAE obtains the lowest accuracy and we conjecture that the KgCVAE can not effectively capture the sentence function information. By contrast, the accuracy obtained by the SeFun-CVAE, in which a type discriminator is added, is higher than that of the KgCVAE. This shows that the added discriminator in the SeFun-CVAE can effectively constrain the encoding information from the latent variable to well represent its corresponding sentence function.
- Compared with two IR-based models, all generative models obtain lower scores on fluency, relevance and informativeness. On accuracy, the best generative models, i.e. SeFun-CVAE(level1) and C-Seq2seq(level2) outperform the IR baselines respectively, but still underperform the re-rank IR models. Thus considering all metrics together, the re-ranked IR model performs the best.

7.4 Case Study

In Figure 3, we present two examples, each of which shows a test query with its target level-2 response sentence function (predicted by the Classification-for-Testing model), and the top-1 responses selected by the IR-based models.

Query	昨天我跟我弟大吵了一架，吵完还打了一架。 I had a quarrel with my brother yesterday and had a fight afterwards.
Target response sentence function	EX with interjections
IR baseline	我倒希望大吵一架再和好 I'd rather have a big fight and make up
Re-ranked IR	我要给我弟弟看哈哈哈笑死我啦 I'm going to show my brother hahaha, you are killing me
Seq2seq	哈哈，好有爱，好有爱。 Haha, so cute, so cute.
C-Seq2seq	笑死我了，太可爱了 You're killing me. It's so cute
KgCVAE	哈哈，我弟弟也这样。 Haha, so does my brother.
SeFun-CVAE	我和我的小伙伴们都惊呆了 My friends and I wereshocked
Query	喜欢发呆的人，心里一定有另一个纯净的世界。 There must be another pure world in the heart of people who like to a daze.
Target response sentence function	Negative DE
IR baseline	为什么喜欢发呆 Why do you like to be in a daze
Re-ranked IR	我的心里藏着一个安静的世界. . . There is a quiet world in my heart
Seq2seq	我也喜欢这样的生活。 I like this kind of life, too
C-Seq2seq	发呆的时候，我也在发呆。 When in a daze, I am also in a daze.
KgCVAE	我喜欢发呆的人。 I like people who are in a daze.
SeFun-CVAE	喜欢发呆的人，一定会有另一个世界 People who like a daze must have another world

Figure 3: Responses of IR-based and generative models. Words in red are related to the target sentence function and words in blue are relevant to the query.

We can see that the IR baseline tends to output responses with more overlapped terms with the query due to the use of Jaccard similarity. However, the obtained responses may not be relevant to the query, as shown in the first case. Whereas, the re-ranked IR model can balance between the compatibility of the target response sentence function and the Jaccard similarity. Thus its selected responses may not have many term overlapped with the query, but the conversations continue more smoothly and coherently.

Responses of the Seq2seq baseline are generic and universal that can be used to reply to a large variety of queries. The three improved methods tend to generate responses with some words related to the target sentence functions and relevant to the query. Thus generative models with the use of sentence function can help improve the response quality, though not as significantly as in the IR-based models.

8 Conclusions

This work introduces the STC-SeFun dataset, which consists of short-text conversation pairs with their sentence functions manually annotated. We first show that classifiers trained on the STC-SeFun dataset can be used to automatically annotate a large conversation corpus with highly reliable sentence functions, as well as to estimate the proper response sentence function for a test query. Using the large automatically annotated conversation corpus, we train and evaluate both IR-based and generative conversation models, including baselines and improved variants considering the modeling of sentence function in different ways. Experimental results show that the use of sentence function can help improve both types of conversation models in terms of response relevance and informativeness.

References

Adrian Akmajian. 1984. Sentence types and the form-function fit. *Natural Language & Linguistic Theory*, 2(1):1–23.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *Proceedings of the 2015 International Conference on Learning Representations(ICLR)*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104.

John J Godfrey and Edward Holliman. 1997. Switchboard-1 release 2. *Linguistic Data Consortium, Philadelphia*, 926:927.

Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards an open-domain conversational system fully based on natural language processing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 928–939.

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*, pages 2042–2050.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating informative responses with controlled sentence function. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1499–1508.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Xiang Li, Lili Mou, Rui Yan, and Ming Zhang. 2016. Stalematebreaker: A proactive content-introducing approach to automatic human-computer conversation. *arXiv preprint arXiv:1604.04358*.

Jackson Liscombe, Giuseppe Riccardi, and Dilek Hakkani-Tür. 2005. Using context to improve emotion detection in spoken dialog systems. In *Ninth European Conference on Speech Communication and Technology*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, 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.

Bilyana Martinovski and David Traum. 2003. Breakdown in human-machine interaction: the error is the clue. In *Proceedings of the ISCA tutorial and research workshop on Error handling in dialogue systems*, pages 11–16.

Helmut Prendinger and Mitsuru Ishizuka. 2005. The empathic companion: A character-based interface that addresses users'affective states. *Applied Artificial Intelligence*, 19(3-4):267–285.

Laurie Rozakis. 2003. *The complete idiot's guide to grammar and style*. Penguin.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics(ACL)*, pages 1577–1586.

Weiyan Shi and Zhou Yu. 2018. Sentiment adaptive end-to-end dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1509–1519.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering output style and topic in neural response generation. In *Proceedings of the 2017*

Conference on Empirical Methods in Natural Language Processing, pages 2140–2150.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

George Yule. 2016. *The study of language*. Cambridge university press.

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 (ACL)*, pages 654–664.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.