# Building Empathetic Transformer-based Chatbot: Deepening and Widening the Chatting Topic

**Junze Li, Yubo Xie**
School of Computer and Communication Sciences, EPFL
Lausanne, Switzerland
{junze.li,yubo.xie}@epfl.ch

## Abstract

Human-machine interaction, particularly via dialogue system, was a popular research area in the past decade. One of the challenges for dialogue systems is to recognize feelings and topics in the multi-turn conversation partner and reply accordingly, which is a key communicative skill in human-human interaction. This project aims at recognizing and acknowledging the speakers' feelings and topics in conversations, by leveraging the novel EmpatheticDialogues dataset proposed by Facebook AI Research. We mainly focus on incorporating topic information into the Transformer framework to generate more informative and interesting responses. Firstly, we implement the Topic-prepend model, which inserts the topic information at the beginning of dialogue sentences based on the pre-trained topic classifier. Secondly, we implement the Topic-aware model which adopts the joint attention mechanism and biased generation probability. The topic words are obtained by a pre-trained LDA topic model. Finally, we conduct experiments to compare our models with several baselines by both automatic and human evaluation. The results indicate that our models yield better performance by deepening and widening the chatting topic.

## 1 Introduction

Conversational agent has been widely implemented in many applications and is still a challenging task in natural language processing research. The existing conversational agents can be divided into task-oriented dialogue systems and non-task-oriented chatbots.

- Task-oriented dialogue systems are used to help people complete specific tasks in vertical domains (Young et al., 2013).
- Non-task-oriented chatbots are designed for chatting with people regarding to a wide range of topics in open domains (Shang et al., 2015).

With the boom of the social media and conversation corpus, there is more and more research about chatbot in recent years. The common methods to build a chatbot are rule-, retrieval-, or generation-based.

- The rule-based method restricts the diversity of the dialogue under the pre-defined rules. And this kind of rules are difficult to make.
- The retrieval-based method depends on the dialogue database heavily. The response can only be retrieved in the database.
- By contrast, the generation-based method can produce more flexible and interesting response by training a model within a machine translation framework (Ritter et al., 2011; Sutskever et al., 2014; Sordoni et al., 2015) based on the large scale social conversation corpus.

Therefore, the generation-based method is preferred in chatbot applications. The single-turn generation-based method (Xing et al., 2017; Shang et al., 2015) ignores the information of the historical context which is important for the following conversation. To solve this problem, multi-turn generation-based method (Serban et al., 2016) is proposed, which involves in both word-level and utterance-level. Especially, after self-attention mechanism and Transformer structure (Vaswani et al., 2017) proposed, the performance of chatbots improves significantly. However, there

are still some remaining problems. The traditional Sequence-to-Sequence with attention models (Cho et al., 2015) and the Transformer structures frequently generate dull responses like "I don't know.", "I am sorry for hear that." or "Me too." (Li et al., 2015), which are not informative or meaningless due to the high frequency of these sentences in the conversation corpus. Such responses may quickly lead the end of the conversation, and severely affect the performance of the chatbot.

In our research, we focus on generating informative and interesting responses that can help chatbot interact with users in a more human-like way. We consider involving topic information into the decoder of Transformer in order to generate more related responses. Given an input multi-turn context, we try to predict the possible topic words that related to the context, and generate responses based on these topic words. This process is like the real human-human conversation, people always change their responses regarding to the concept or the topic they are talking about. For example, when people say "I love holidays.", the other one may think about different festivals and holidays. Based on this related topic, he may response "Christmas is my favorite time of the year. What is your favorite thing about it?" rather than "That's great" or "Me too". This more informative response will help the dialogue continue. They may talk about the things that they usually do in the holidays. This kind of prior knowledge is useful for empathetic dialogues.

We simulate the way that people response regarding to topics and implement two kinds of models based on Transformer structure. The first one is Topic-prepend model. A pre-trained topic classifier predicts the topic of the context, and prepend this possible topic word in front of the input context in order to add the supervised topic information into the chatbot. The other one is the Topic-aware model. Topic-aware model modifies Transformer structure. In the encoder part, the model transforms an input context into hidden vectors under self-attention mechanism, and acquires embeddings of the topic words extracted from a pre-trained LDA model. These topic words can be seen as the simulation of topical concepts in people's minds, and this LDA model is pre-trained with large scale social media conversations outside the training data. In the decoder part, each word is generated according to both the input context and the predicted topic words through a joint attention mechanism. In joint attention, hidden vectors of the input context are summarized as context vectors by self-attention which follows the existing attention techniques in Transformer, and embeddings of topic words are synthesized as topic vectors by topic attention. Different from existing self-attention mechanism, in topic attention, the weights of the topic words are calculated by taking the final state of the input context as an extra input in order to strengthen the effect of the topic words relevant to the input sentence. The joint attention lets the context vectors and the topic vectors jointly affect response generation, and makes words in the generated response not only relevant to the input sentence, but also relevant to the correlated topic information of the sentence. To model the behavior of people using topical concepts as "building blocks" of their responses, we modify the generation probability distribution of a topic word by adding another probability item which biases the overall distribution and further increases the possibility of the topic words appearing in the response.

We implement an empirical study on the EmpatheticDialogues dataset from Facebook (Rashkin et al., 2019), and compare different models with both automatic evaluation and human evaluation. The results on both automatic evaluation metrics and human annotations show that Topic-prepend and Topic-aware models can generate more informative, diverse, and topic relevant responses and significantly outperform state-of-the-art methods for response generation.

## 2 Related work

Chatbot has been advanced due to the fast development of deep neural network technologies and the increase of public datasets and benchmarks. In the introduction section, we know that generation-based method is adopted by most dialogue systems, which is inspired by statistical machine translation. At first, researchers implement encoder-decoder framework to build single-turn dialogue systems (Ritter et al., 2011; Shang et al., 2015). After the attention mechanism proposed in machine translation (Bahdanau et al., 2014), it has been incorporated into the encoder-decoder framework to improve the model performance (Xing et al., 2017). Later, more researchers focus on modeling the conversational history in multi-turn

dialogue systems. Dynamic-context generative model (Sordoni et al., 2015) encodes the context into fixed-length vectors and feeds them into the RNN language model for response generation. HRED (Serban et al., 2016) models the hierarchy of contexts with two RNNs: one for word level and the other one for utterance level. Based on the HRED, VHRED (Serban et al., 2017b) introduces the latent stochastic variables into the model and MrRNN (Serban et al., 2017a) considers multiple parallel sequences by factorizing the joint probability over the sequences. Dynamic attention mechanism in RNN is proposed to increase the scope of attention on the conversation history (Mei et al., 2017). Deep reinforcement learning can also be integrated into the Sequence-to-Sequence structure in order to simulate the dialogue between two agents (Li et al., 2016b).

The above methods are used to generate proper and fluent responses. However, in order to generate more informative and characteristic responses, Xing et al. extract topic words of input sentences and involve these topic words into joint attention mechanism based on Sequence-to-Sequence model (Xing et al., 2017). Li et al. build a personalized dialogue system by adding character information (Li et al., 2016a). Gu et al. introduce the CopyNet to simulate the repeating behavior in human-human conversation (Gu et al., 2016). Yao et al. add an extra RNN in Sequence-to-Sequence model to represent the dynamics of the intention process.

## 3 Background

Before introducing our backbone models, we first briefly overview the basic Transformer structure and the Latent Dirichlet Allocation (LDA) model.

### 3.1 Transformer structure

The encoder-decoder framework is the common structure of most neural response generation models. The encoder transforms an input sequence of words $\mathbf{X} = (x_1, x_2, ..., x_n)$ into a sequence of continuous representations $\mathbf{z} = (z_1, z_2, ..., z_n)$. Given $\mathbf{z}$, the decoder then generates an output sequence of tokens $\mathbf{Y} = (y_1, y_2, ..., y_m)$ one token at each time step. The decoder is auto-regressive at each time step, given the previous generated token as additional input when generating the next. The objective function can be written as:

$$p(\mathbf{Y}|\mathbf{X}) = p(y_1|\mathbf{z}) \prod_{t=2}^{m} p(y_t|\mathbf{z}, y_1, ..., y_{t-1}) \quad (1)$$

The Transformer follows this encoder-decoder framework using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder respectively.

#### 3.1.1 Encoder and Decoder Stacks

The encoder stack consists of 6 identical layers. Each identical layer has one multi-head attention sub-layer and one feed forward sub-layer. There is also a residual connection around each of the two sub-layers, followed by layer normalization (Ba et al., 2016).

The decoder stack also consists of 6 identical layers. But each identical layer of decoder has three sub-layers. There is an additional multi-head attention over the output of the encoder stack. Similar to the encoder, there are also residual connections around each of the sub-layers, followed by layer normalization. The first multi-head attention layer is also modified to ensure that the current generation only depends on the previous generated tokens, which is called masked multi-head attention layer.

#### 3.1.2 Self-attention mechanism

Each embedding vector is mapped to three vectors, query($\mathbf{q}$), keys($\mathbf{k}$) and values($\mathbf{v}$) by three corresponding mapping matrix. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. In practice, we pack all the queries, keys and values vectors into $Q$, $K$ and $V$ matrices. We compute the self-attention $\alpha$ as:

$$\alpha(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (2)$$

where $d_k$ denotes the dimension of $\mathbf{q}$. The multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions, which is computed as,

$$\beta = concat(\alpha_1, ..., \alpha_h)W^O \quad (3)$$

$$\alpha_i = \alpha(QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

where $\beta$ denotes the multi-head attention, $W^O$, $W_i^Q$, $W_i^K$ and $W_i^V$ are all projection matrices,

which are corresponding to dimension of $\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$. In multi-head attention, different $\alpha_i$ may focus on different part of the input sentences in order to extract more features of the input sentences.

#### 3.1.3 Point-wise Feed-Forward Networks

In addition to the multi-head attention layers, both encoder and decoder contain a fully connected feed-forward network. The function of this feed-forward network can be written as:

$$\gamma(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (5)$$

where $\gamma$ denotes the output, $W_1$, $W_2$ are weight matrices and $b_1$, $b_2$ are biases. This layer consists of two linear transformations with a ReLU activation function.

#### 3.1.4 Positional encoding

Up to now, we do not make use of the sequence order of the input sentence. We implement the positional encoding to extract the position information of the input sentence. The computation of the positional encoding is,

$$C_{pos,2i} = sin(pos/10000^{2i/d_{model}}) \quad (6)$$

$$C_{pos,2i+1} = cos(pos/10000^{2i/d_{model}}) \quad (7)$$

where $pos$ denotes the position and $i$ denotes the dimension of encoding. $d_{model}$ is the dimension of the input vector and output vector.

### 3.2 LDA model

Latent Dirichlet Allocation (LDA) is the common method used for text modeling and topic modeling. LDA model intuitively supposes that documents are generated from multiple topics, and topics are generated from words in the dictionary. First of all, the dictionary should be pre-defined, and the words appearing in the documents will not exceed the scope given by the dictionary. For example, the topic "festival" contains a number of words about festival with a high probability in the dictionary.

For each document in the document collection, the LDA model in Figure 1 generates every topic word in dictionary as follows:

1. Randomly choose the document-topic distribution $\vartheta_m$ among $M$ documents from a symmetric Dirichlet prior distribution $\alpha$;

2. Choose a topic $z_{m,n}$ to generate $n$ word in the $m$ document, drawn from the $\vartheta_m$ distribution;

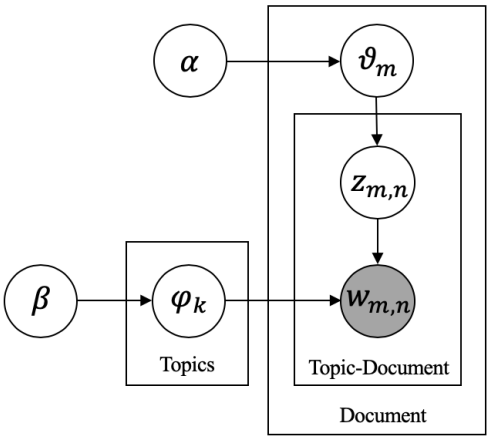


Figure 1: LDA model structure

3. For each of $K$ topics, choose the topic-word distribution $\varphi_k$ where $k = z_{m,n}$ from a symmetric Dirichlet prior distribution $\beta$;

4. Choose word $w_{m,n}$ from the topic-word distribution $\varphi_k$ corresponding to $z_{m,n}$.

This hierarchical Bayesian model estimate $\varphi_k$ and $\vartheta_m$ provides information about the topics that participate in a corpus and the weights of those topics in each document respectively. The core computational problem of topic modeling is the use of observed documents to infer hidden topic structures. Gibbs sampling is used to estimate these parameters. This can also be seen as the inverse of the generative process.

## 4 Backbone Model

After the basic introduction about the Transformer structure and LDA model, we introduce two improved models, which are Topic-prepend and Topic-aware. The Topic-prepend model needs additional information from supervised predictors, and the Topic-aware model is based on the unsupervised LDA model.

### 4.1 Topic-prepend Model

Some machine learning methods have been proposed on supervised tasks that may be relevant to empathetic responding, for example sentiment analysis and text classification. If we introduce these methods into the basic transformer architecture, the previous training process and some other training resources may improve the overall performance without retraining the model or requiring access to the data involved in the pre-training process, which is user-friendly for the future imple-

mentation. We experiment with adding predicted label in the topic prediction task, which may be effective for generating more topic-related replies.


Figure 2: Topic-prepend model

As shown in Figure 2, this is a very simple way to add supervised information to the input sentence, which requires no additional modification of the basic transformer architecture and can be used with pre-trained classifiers based on different supervised learning methods. An input sentence (either a dialogue context or a candidate) is run through a pre-trained topic classifier, and the predicted topic label is prepended to the sentence, which is then as the input of the transformer encoder. The original and processed sentences are as below:

**Original**: "I participate in a software development project."

**Prepend**: "*technology* I participate in a software development project."

In our experiment, we use the fastText model (Joulin et al., 2016) as the prediction classifier. The fine-tuning process of the transformer is implemented similarly as before, but using these processed inputs. We use the 20-Newsgroup dataset (Joachims, 1996) as the external source to train the fastText model for topic classification. Similar methods have been also used for controlling the style of generated text (Niu and Bansal, 2018). This Topic-prepend model may be considerably argued about the effective capacity, as well as the source and the amount the external training data used overall, but the purpose of this method is to get an empirical sense of how supervised information will improve the robust performance of the dialogue generation.

### 4.2 Topic-aware model

Compared to the Topic-prepend model which combines the supervised information, the Topic-aware model combines the unsupervised information based on the LDA model.

Suppose that we have a dataset $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{K}_i, \mathbf{Y}_i)\}_{i=1}^{N}$, where $\mathbf{X}_i$ denotes the input sentence, $\mathbf{K}_i$ denotes the topic words of $\mathbf{X}_i$ and $\mathbf{Y}_i$ denotes the corresponding response. The training of the Topic-aware model is based on the dataset $\mathcal{D}$ and we can use this model to generate more empathetic responses for the input sentences with topic words. Firstly, we introduce how to acquire these topic words and then we explain the main mechanism in the Topic-aware model.

#### 4.2.1 Topic words acquisition

We acquire the topic words of input sentences from a pre-trained LDA model, which is a kind of probabilistic topic models and represents the state-of-the-art model for social texts and dialogues. Figure 1 shows the graphical model of LDA.

The basic assumption is that each input sentence corresponds to one topic, and each topic has different probability distribution of the topic words in this particular topic. We estimate the parameters in the LDA model by Gibbs sampling algorithm. After the estimation, one topic is assigned to the input sentence $\mathbf{X}$, and choose the top $n$ words with the highest probabilities as the topic words $\mathbf{K}$. When we construct the dictionary of the LDA model, the stop words such as "the" and "me" are removed. The computation of the probability distribution of the topic words is as follow,

$$p(z|w) \propto \frac{C_{wz}}{\sum_{z'} C_{wz'}} \tag{8}$$

where $z$ denotes the topic, $w$ denotes the word and $C_{wz}$ denotes the number of times that topic $z$ contains word $w$. Then we can use this distributions as the vector representations of the topic words in training.

In our experiment, we train the LDA model based on the OpenSubtitles dataset and part of Twitter data. The scenarios of these two kinds of data are similar to that of the EmpatheticDialogues dataset. In daily human-human conversations, people always extract the topics of the context and say something related to this topic as the response. Similarly, these external dataset can be

seen as the prior topic knowledge provided to the transformer in order to generated empathetic responses.

Some other social media corpus or web documents can also be used to train LDA model. Different data source may let the model learn different methods to extract different style topics. In addition to LDA model, one can also implement other techniques like tag recommendation to generate topic words (Wu et al., 2016).

#### 4.2.2 Model

Figure 3 shows the structure of the Topic-aware model, which is built on the Transformer framework. The joint attention mechanism and biased generation probability are implemented into the model to leverage the topic information in generated responses.

In encoder, each word in the input sentence $\mathbf{X}$ obtains its word vector by word embedding layer and then this word vector is mapped to query, key and value vectors respectively by three corresponding weight matrices, which can be written as $\{\mathbf{q}_{in}^i, \mathbf{k}_{in}^i, \mathbf{v}_{in}^i\}_{i=1}^T$. These hidden vectors are transformed to a context vector $\mathbf{c}_i$ at time step $i$ by the multi-head attention layer. This context vector is the output of the encoder and external input of the decoder.

In decoder, The topic words $\mathbf{K}$ of this input sentence $\mathbf{X}$ also obtain the corresponding word vectors by word embedding layer, which can be written as $(\mathbf{k}_1, ..., \mathbf{k}_n)$. Each word vector is also mapped to query, key and value vectors, which can be written as $\{\mathbf{q}_k^i, \mathbf{k}_k^i, \mathbf{v}_k^i\}_{i=1}^n$. Each word in the output sentence is also mapped to query, key and value vectors $\{\mathbf{q}_{out}^i, \mathbf{k}_{out}^i, \mathbf{v}_{out}^i\}_{i=1}^m$ in the same way. The multi-head attention combines the hidden vectors of each topic word as a topic vector $\mathbf{o}_i$ at time step $i$. Compared to the traditional multi-head attention, the topic attention leverage the relevance between output sentence and topic words and highlight the importance of relevant topic words. As a result, the topic vectors $(\mathbf{k}_1, ..., \mathbf{k}_n)$ are more correlated to the content of the input sentence and noise in topic words is controlled in generation.

The context attention and the topic attention consist the form the joint attention mechanism together, which allows context vector and topic vector jointly affect the generation probability. The advantage of the joint attention is that it makes words in responses not only relevant to the message, but also relevant to the topics of the input sentence.

The final probability distribution $p(y_i)$ should also be modified. Here, we define $p(y_i) = p_V(y_i) + p_K(y_i)$ and the computation of $p_V(y_i)$ and $p_K(y_i)$ is as follow,

$$p_V(y_i = w) = \begin{cases} \frac{1}{Z} e^{\eta_V}, & w \in \mathbf{V} \cup \mathbf{K} \\ 0, & w \notin \mathbf{V} \cup \mathbf{K} \end{cases} \quad (9)$$

$$p_K(y_i = w) = \begin{cases} \frac{1}{Z} e^{\eta_K}, & w \in \mathbf{K} \\ 0, & w \notin \mathbf{K} \end{cases} \quad (10)$$

where $\mathbf{V}$ denotes the dictionary of the generated sentences and $Z = \Sigma_{v \in \mathbf{V}} e^{\eta_V} + \Sigma_{v' \in \mathbf{K}} e^{\eta_K}$ is a normalizer. Here, we define two functions $\eta_V$ and $\eta_K$, which denote two full connected layer and can be written as,

$$\eta_V = \sigma(\mathbf{w}^T(\mathbf{W}_V^s \cdot \mathbf{s}_i + \mathbf{W}_V^y \cdot y_{i-1} + \mathbf{b}_V)) \quad (11)$$

$$\eta_K = \sigma(\mathbf{w}^T(\mathbf{W}_K^s \cdot \mathbf{s}_i + \mathbf{W}_K^y \cdot y_{i-1} + \mathbf{W}_K^c \cdot \mathbf{c}_i + \mathbf{b}_K)) \quad (12)$$

where $\sigma(\cdot)$ denotes the tanh function, $\mathbf{s}_i$ denotes one hidden vector of the decoder at time step $i$ and the others are parameters of the full connected layer.

Therefore, the generation probability $p(y_i)$ tends to be biased to topic words. For non topic words in the dictionary, the generation probability $p_V(y_i)$ is not biased but related to the topic words by joint attention mechanism. For the topic words, the generation probability $p_K(y_i)$ is non-zero, which increases the possibility of the topic words appearing in the generated sentence. From equation (10) and (12), we can see that $p_K$ is determined by the current hidden vector, the previous generated word and the context vector, which means that the more relevant the topic word is, the more possible it will appear in the output sentence based on the input sentence and the generated part of the output sentence.

The first word of the output sentence will be more accurate in Topic-aware model. The first word plays a key role in the generated sentence, since the following part of the sentence is based on the first word. The more proper the first word is, the more fluent and accurate the sentence is. In the tradition transformer architecture, the choice of the first word is only determined by the context vector, because there is no word generated already. However, by the joint attention mechanism, the choice of the first word is not only based on the context vector, but also the topic vector, which

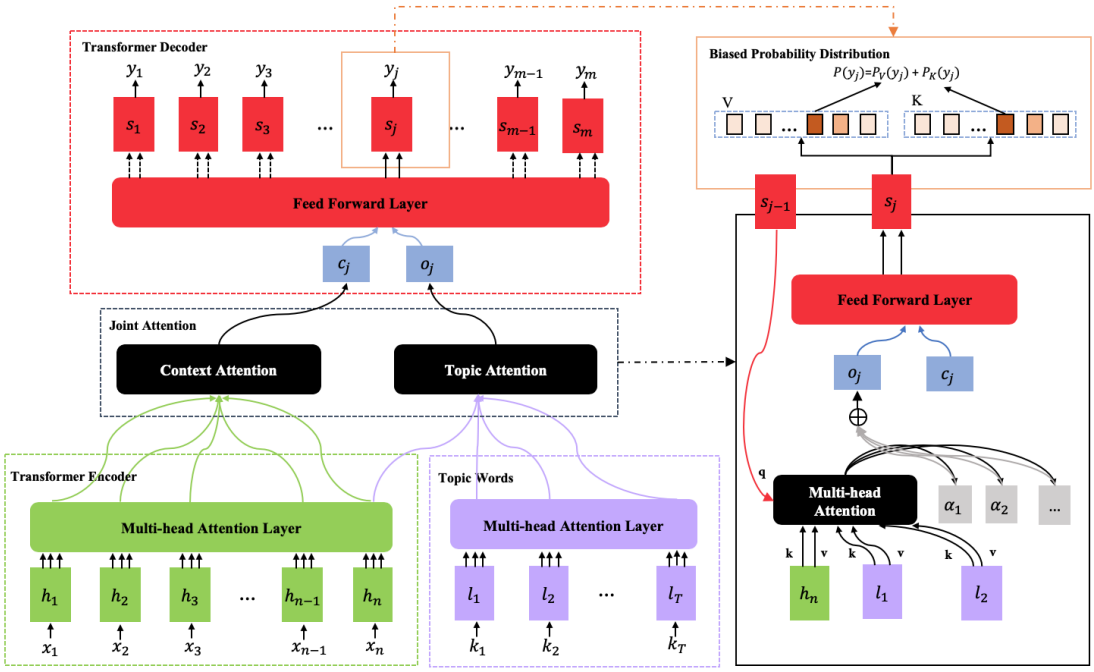


Figure 3: Topic-aware model structure

makes sure that the first word is more accurate and related to the dialogue topic.

Compared to the previous topic related structure VHRED (Serban et al., 2017b), the topic learning part and the output sentence generating part are separated in the Topic-aware model, which means that the topic learning model can be changed or retrained on other data without affecting the structure of the generating part. In this way, people can change different topic learning style for their preference.

Through this method, our model allows appearance of multiple topic words rather than merely fixing a single key word in responsed like what Mou et al. did in their work (Mou et al., 2016). So the appearance of the topic words is in a more flexible way.

## 5 Experiment

We compare Topic-prepend and Topic-aware model with the-state-of-the-art dialogue generation model by both automatic evaluation and human evaluation.

### 5.1 Experiment setup

In our experiment, we use the EmpatheticDialogues dataset from Facebook, which comprises 24,850 conversations about a situation description, gathered from 810 different participants. This dataset is splited into 80% train, 10% validation and 10% test partitions. To prevent overlap of discussed situations between partitions, we split the data so that all sets of conversations with the same speaker providing the initial situation description would be in the same partition. The final train/val/test split is 19533/2770/2547 conversations, respectively. For each conversation, the input sentences are in multi-turn order to extract more context information. We implement the Transformer Tokenizer to tokenize the sentences.

We use the corpus from Open Subtitles website to train the LDA model. We set the number of topics as 100 and the hyperparameters as $\alpha = 0.01$, $\beta = 0.01$. For each topic, we choose top 50 words as topic words. We remove the stop words from the word dictionary of LDA model.

### 5.2 Automatic evaluation

To measure the performance of the Topic-prepend and Topic-aware model, we follow existing studies and adopt several standard metrics: perplexity (PPL) (Vinyals and Le, 2015), BLEU (Papineni et al., 2002), and diversity-based Distinct-1 (Li et al., 2015). We compare the Topic-prepend and Topic-aware model with the baselines in terms of these metrics. In particular, PPL describes how well a probability model predicts the target samples. BLEU quantifies n-gram overlaps between

the generated sentence and the ground-truth. In some way, Distinct-1 reflects the diversity of the generated sentences.

**PPL**: PPL is widely used in probability models to quantify their performance. PPL is defined as follow,

$$PPL = exp\left\{-\frac{1}{N}\Sigma_{i=1}^{N} log(p(\mathbf{Y}_i))\right\} \quad (13)$$

A lower PPL score indicates better generation quality. The PPL score can also be used to decide when to stop training. If the PPL score is not decreasing any more, the training can be stopped.

**BLEU**: BLEU is widely used in machine translation. Formally, BLEU-N score is calculated by,

$$BLEU = exp(min(1 - \frac{r}{c}, 0) + \Sigma_{n=1}^{N} w_n log p_n) \quad (14)$$

where $r$ and $c$ respectively denote the lengths of the reference response and candidate one, $p_n$ presents the modified n-gram precision, N means using n-gram up tp length N and $w_n = \frac{1}{N}$. Higher BLEU score indicates the better performance.

**Distinct-1**: Distinct-1 is calculated as the number of distinct unigrams in the generated sentences scaled by the total number of generated tokens. This metric measures how informative and diverse the generated sentences are. High numbers and high ratios mean that there is much information in the generated sentences, and the high numbers further indicate that the generated sentences are long.

Table 1 shows the results of the automatic evaluation. The bolded score indicates the best performance.

| Model | PPL | BLEU-1 | Distinct-1 |
|---|---|---|---|
| Pretrained | 43.82 | 0.1837 | 0.0375 |
| Fine-tuned | 35.28 | **0.2722** | 0.0462 |
| Topic-prepend | 35.04 | 0.2621 | 0.0625 |
| Topic-aware | **33.28** | 0.2675 | **0.0732** |

Table 1: Automatic evaluation results

### 5.3 Human evaluation

In addition to the automatic evaluation, we also invited human annotators to judge the quality of the generated sentences from different baseline models. We recruited 3 students from our university and 2 students from other universities (2 males and 3 females, aged 23-24). All of them are fluent English speaker and have experience of studying in English-teaching universities. We carried out our experiment on Tencent Questionnaire platform. We randomly shuffled 50 dialogues in the test set as contexts and obtained the respective responses from different baseline models. The annotators judge the quality of the generated responses according to the following criteria:

**+2**: The response is not only topic-related and natural, but also fluent and informative.

**+1**: The response is appropriate to the context, but it is less informative and universal.

**0**: The response can not be used as the response to the context. It is either irrelevant or not fluent.

For this experiment, the questionnaires were released on 4 January 2020, and collected by 8 January 2020. Table 2 shows the results of human evaluation. We visualize the proportion of each score for each baseline model. The highest proportions of different scores are bolded.

| Model | +2 | +1 | 0 |
|---|---|---|---|
| Pretrained | 10.6% | 27.3% | **62.1%** |
| Fine-tuned | 33.5% | 36.3% | 30.2% |
| Topic-prepend | 36.2% | **37.2%** | 26.6% |
| Topic-aware | **40.8%** | 33.7% | 25.5% |

Table 2: Human evaluation results

### 5.4 Evaluation results

From Table 1, we can see that Topic-aware model has the lowest PPL score and highest Distinct-1 score. Plain Fine-tuned Transformer model has the highest BLEU-1 score. Topic-prepend model also performs better than plain Transformer model on PPL score and Distinct-1 score. Therefore, the responses generated from our topic related models are more fluent and diverse.

In the human evaluation, both Topic-prepend and Topic-aware models have better performance than fine-tuned Transformer (higher proportion of +2 and lower proportion of 0). Comparing these two topic related model, we can see that Topic-aware model can generate more high quality topic related responses (higher proportion of +2 score).

### 5.5 Case study

Table 3 presents 3 contexts and the corresponding responses from 4 baseline models. From the comparison, we can find that there are more topic related words appearing in the responses generated by Topic-aware model. However, the responses from Topic-prepend model are sometimes similar to the plain Transformer model. Because in

Topic-prepend model, only one predicted topic label is added to the sentence, which is trivial for long input sentences. However, the Topic-aware model implements joint attention mechanism and the topic information can be involved in a more global way for input sentences regardless of the sentence length. So the responses from Topic-aware model are more unique and informative.

## 6 Conclusion

We implement two topic related models (Topic-prepend and Topic-aware) to incorporate topic information into the Transformer structure. Topic-prepend model consists a topic label classifier based on supervised learning. Topic-aware model consists a LDA topic model and adds joint attention mechanism and biased generation probability mechanism based on unsupervised learning. Empirical study on both automatic and human evaluation shows that these two model can generate empathetic and informative responses, and Topic-aware model performs better.

## References


Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Thorsten Joachims. 1996. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2017. Coherent dialogue with attention-based language models. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. 2017a. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017b. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

| Context 1 |
| --- |
| A: Last semester was probably the largest challenge of my life.<br>I go to school for electrical engineering, and I took 6 courses.<br>B: That is a lot.. most I ever took at one time was 5 classes and I cried at least twice that semester.<br>A: Oh my gosh then you know! I have to say though, it all came out in the wash - 4 As and 2 Bs! |
| Response 1 |
| **Pretrained**: I'm sorry to hear that.<br>**Fine-tuned**: I know right? I am so proud of myself!<br>**Topic-prepend**: That's great, I'm glad you were able to make it through all that hard work.<br>**Topic-aware**: I know right. I am glad you were able to go through a tough semester. |
| Context 2 |
| A: The economy has been doing well. The future looks bright.<br>B: I agree! Things look really good going forward.<br>A: Hopefully prosperity will carry over into the next few generations. |
| Response 2 |
| **Pretrained**: I am so sorry to hear that.<br>**Fine-tuned**: I hope so too!<br>**Topic-prepend**: Yep. It's going to be a good year for all of us.<br>**Topic-aware**: I hope so too. The economy will get better. |
| Context 3 |
| A: My youngest child started high school this year. It really made me think back to her starting kindergarten, and how time really goes by. I can just see her learning to read!<br>B: It is so hard and so rewarding to see them grow up.<br>A: Yes, it really is bittersweet. But, the older they get, the easier it is to think back to the younger days! |
| Response 3 |
| **Pretrained**: I am so sorry to hear that.<br>**Fine-tuned**: I know exactly what you mean.<br>**Topic-prepend**: I know exactly what you mean.<br>**Topic-aware**: They will grow up someday. |

Table 3: Case study

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

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. *arXiv preprint arXiv:1506.06714*.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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

Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. 2016. Improving recommendation of tail tags for questions in community question answering. In *Thirtieth AAAI Conference on Artificial Intelligence*.

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*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.