Microsoft Research Asia at the NTCIR-13 STC-2 Task

Zhongxia Chen
University of Science and Technology,Hefei,P.R.C
czx87@mail.ustc.edu.cn

Hongyan Huang
University of Science and Technology,Hefei,P.R.C
colors@mail.ustc.edu.cn

Dinglong Li
University of Science and Technology,Hefei,P.R.C
lidinglongldl@gmail.com

Ruihua Song
Microsoft Research Asia,Beijing,P.R.C
song.ruihua@microsoft.com

Xing Xie
Microsoft Research Asia,Beijing,P.R.C
xing.xie@microsoft.com

ABSTRACT

This paper describes our approaches in NTCIR-13 on short text conversation(STC) task (Chinese). For retrieval-based method, we propose a response matching and ranking model which takes not only the text information into account, but also considers visual features of images corresponding to the text. For generation-based method, we propose the emotion-aware neural response generation model. Based on the attention-based sequence-to-sequence model, our model generates emotional responses by involving emotion information while decoding. Official results show that both emotion and image information improve the effectiveness of response retrieving or generating, and our best run gains 0.1822 for mean nDCG@1 , 0.3002 for mean P^+ and 0.3241 for mean nERR@10.

Team Name

MSRSC

Subtasks

Short Text Conversation (Chinese)

Keywords

Short Text Conversation, Image to Vector, Fast-rank, Re-ranking, Sequence-to-sequence, Attention Mechanism, Emotion-aware, Result Fusion

1. INTRODUCTION

We participated in the NTCIR-13 Short Text Conversation (STC) subtask. Given a new post, this task aims to retrieve an appropriate comment from a large post-comment repository, or generate an new comment learning from the dataset. All the comments are judged from four facets: **Fluent**, **Coherent**, **Self-sufficient** and **Substantial** [5].

The principle of a suitable comment is that this comment can fit the context to respond the given post. Both retrieval-based methods [2] and generation-based methods [6, 4, 7] in previous works focus on learning or extracting context of the post and obtain the response refer to the context.

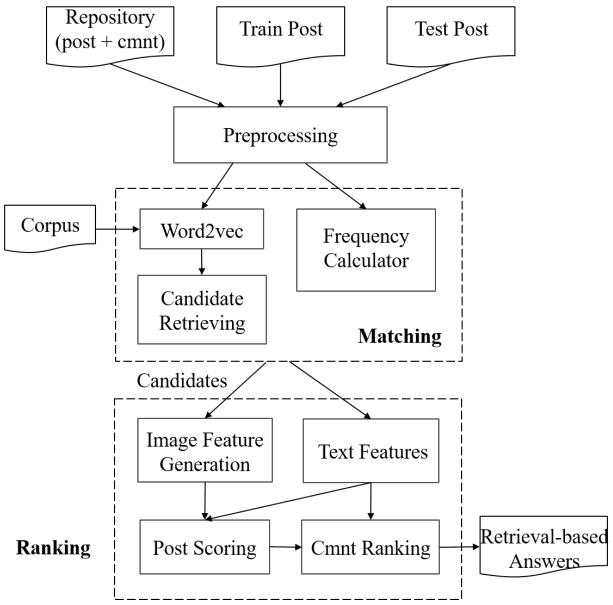
However, all these works focus on the text information of the post. The another important approach to obtain the context of the post, extracting from corresponding images, has hardly been considered in STC task. Yet image also shares vast context information of the post.

To comprehend a short text, image may contains some distinctive features independent of word features. For example, Figure 1 shows corresponding images of the text "How powerful is the consumption ability of 'The Corn'" in Bing image search. From those images, it is easy to understand that the topic of the text is related to the famous Singer *Li Yuchun* which is even not appeared in the text. However, understanding the meaning of "The Corn" - the fans of *Li Yuchun* - with text information seems difficult. Thus we believe that visual features can be effective while learning context.

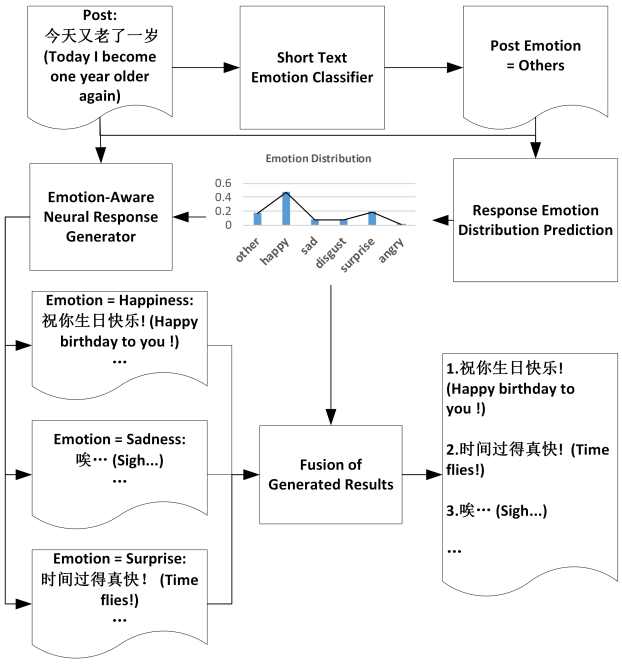
Figure 1: Corresponding images of the text retrieved by Bing image search.

Besides understanding the context, emotion is also an important aspect of human conversation. Though it has hardly been investigated in STC, emotion plays an important role in responding. For example, for the post "Today I become one year older again", here are three different comments: "Happy birthday to you!", "sigh..." and "Time flies!". The first comment expresses happiness about the birthday, the second comment feels sad about that and the last one shares surprise about the fact mentioned in the post. All these responses are relevant and appropriate for responding to the post. If emotion information is ignored and the system chooses randomly from the possible responses, the system is unable to generate responses of appropriate emotions corresponding to the personality.

Thus in this report, we introduce our methods which take visual features and emotion information into account. For retrieval-based method, we propose a matching and ranking model with image information involved. Given a new post, the matching module first obtain similar posts in the repository according to the text similarities and retrieve these



(a) System architecture of retrieval-based method



(b) Workflow of emotion-aware neural response generation model

Figure 2: Structures of our retrieval-based method and generation-based method.

posts and corresponding comments as candidates. Then the ranking module is proposed to rank candidates by both visual features and text features. The visual features are extracted from images returned from Bing image search engine.

For generation-based method, we propose an emotion-aware STC model to generate a response containing appropriate emotion. We first predict the possible emotion(s) to be expressed in the responses. The response will be generated with respect to the required emotion. We apply convolutional neural networks to classify short text emotions and predict suitable comment emotions for a given post. While generating short text responses, based on a basic encoder-decoder model with recurrent neural network (RNN) and attention mechanism, emotion information of comments are fed into decoder to separately generate comments with different emotions. At the end a fusion method ranks the generated comments and determine a response according to the comment emotion predictor.

2. SYSTEM DESCRIPTION

In this section, we introduce our solution of taking visual features into account for retrieval-based method and our emotion-aware neural response generation model for generation method.

2.1 Retrieval-based Method

The whole structure of our retrieval-based method is shown in Figure 2a. Our system consists of pre-processing, matching and ranking. For a given new post, we retrieve the similar posts and corresponding comments using text features from the large repository. Limited by the cost of generating image features, image feature extracting unit is just applied after the matching module. All these features is used to calculate the overall score of post similarity. Then we rank candidate comments with both post similarities and text features of comments.

2.1.1 *Preprocessing*

We use Open Chinese Convert[1] to convert traditional Chinese characters to simplified Chinese characters. For word segmentation, we choose Jieba Chinese word breaker[2] to split the sentences into word sequences. After segmentation, our system filters meaningless words and symbols according to Chinese stop words list.

2.1.2 *Matching*

In this module, our system first extracts semantic feature for post texts and frequency feature for comment texts in repository. Then for each input post, we retrieve a small subset of post-comment pairs in the repository refer to the extracted features for the following ranking module.

- **Semantic Feature**

  We use Google Word2Vec[3] to extract sentence embeddings to represent semantic information of short texts. We use Wikipedia Corpus[4] as the training corpus of word2vec. After computing continuous distributed representations of words, we combine word vectors with summation as the representation of the whole text.

- **Frequency feature**

[1]https://github.com/BYVoid/OpenCC
[2]http://github.com/fxsjy/jieba
[3]https://code.google.com/archive/p/word2vec/
[4]https://dumps.wikimedia.org/

We define the frequency feature according to the Power Law as follow:

$$F_{fre}(p, C_i) = \log\left((C_i)_{frequency}\right)$$

Here $(C_i)_{frequency}$ means the population of comment C_i in the repository.

- **Candidate Retrieving**

  For reducing the size of candidates in the ranking period, we calculate the cosine similarities of posts and only keep the similar post and corresponding comments:

$$F_{sim}(p_1, p_2) = \frac{\overrightarrow{T_1} \cdot \overrightarrow{T_2}}{\|\overrightarrow{T_1}\|\|\overrightarrow{T_2}\|}$$

  Here p_1 and p_2 are short text posts, $\overrightarrow{T_1}$ and $\overrightarrow{T_2}$ are corresponding sentence text feature vectors.

2.1.3 Ranking

In the ranking module, we rank the candidates by taking both text and image information into account.

- **Image Feature Generation**

  Given a short text post, we retrieve images from Bing image search engine. Due to the search efficiency, we split the text P into several short queries $P = (Q_1, ..., Qn)$. Here we use Jieba segmenter to obtain word sequence and split the sequence into queries which each queries have up to eight characters.

  For each query Q_i, we retrieve t images in the search engine. For each image $Pic_{i,j}$, we extract the visual feature vector $V_{i,j}$. Here $V_{i,j}$ is the vector before the output layer of a convolutional neural network model training on ImageNet. We then use clustering method and choose the center of the largest cluster as the image feature vector of query Q_i (denoted as V_{Q_i}).

  So finally we obtain the visual feature vector set $V_P = (V_{Q_1}, V_{Q_2}, ..., V_{Q_n})$.

- **Post Scoring**

  We then re-calculate the similarity between posts with visual feature vectors as the post scores. For the given post p and a candidate post P_i, the similarity score is calculated as:

$$Score(p, P_i) = Ranker(V_p, T_p, V_{P_i}, T_{P_i})$$

  Here *Ranker* is obtained by the Fastrank learning algorithm, V_p and V_{P_i} are visual features, T_p and T_{P_i} are text features.

  We assume that two similar posts has two similar comment sets. While training fastrank regression model, the ground truth of similarity score is defined as the average similarity of two comment sets. And the similarity of comments is calculated using text feature vectors.

- **Comment Ranking**

  Finally we rank all the comments in the candidate set as follow:

$$Score(p, C_i) = Score(p, P_i) \cdot F_{fre}(p, C_i) \cdot F_{sim}(p, C_i)$$

  Here (C_i, P_i) is the corresponding post-comment pair.

Thus we obtain top ten comments as the final result after ranking module.

2.2 Generation-based Method

Figure 2b shows the flowchart of the whole process of our generation model. When responding a post, we first obtain the emotion of post by an emotion classifier and determine the probabilities of possible emotions of comments. When generating comments, we generate short text responses under the condition of different emotions, and then fuse them according to their predicted emotions and the probabilities of possible emotions. We provide more details in the following subsections.

2.2.1 Definition of Emotion

Following Ekman et al. [1], the emotion can be classified into seven categories: *neutrality*, *happiness*, *sadness*, *disgust*, *fear*, *surprise*, and *anger*. We combine *fear* with *neutrality* (named as *others*) and keep the other categories as our definition of emotion.

2.2.2 Short Text Emotion Classifier

A basic component of our work is an emotion classifier of short text. From previous works, we choose Kim-CNN proposed in [3] to implement our emotion classifier.

Given the input word sequence of a sentence $x = (x_1, ..., x_T)$, its corresponding embedding vectors are $\bar{x} = (\bar{x_1}, ..., \bar{x_T})$. For every filter w and window size of filter l, the convolution operation builds a feature map $\bar{c} = (\bar{c_1}, ..., \bar{c_{T-l+1}})$, where:

$$\bar{c_i} = f(w \cdot \bar{x}_{i:i+l-1} + b)$$

with $i = (1, 2, ..., T - l + 1)$. Here b is a bias term, f is a non-linear function and $\bar{x}_{i:i+l-1}$ is the concatenation of embedding vectors $\bar{x}_i, ..., \bar{x}_{i+l-1}$. With multiple filter widths and feature maps, we create multiple feature vectors. We then apply a 1-max pooling operation over each feature vector and keep the maximum value $\hat{c} = max\bar{c}$ as the feature for the particular filter.

After generating multiple features from multiple filters, we feed them into a fully connected softmax layer with dropout to obtain the probability of each emotion class and the emotion class with the maximum probability will be predicted as the emotion of this sentence.

We train the model on a large scale Weibo dataset which contains in total 1,200,000 short texts with emotion labels (200,000 for each emotions, labels are determined by the emoticon to emotion mapping).

2.2.3 Response Emotion Distribution Prediction

The goal of this component is to obtain the emotion distribution of possible responses for a given post. The task is similar to short text emotion classification but with different input and output. The input is a post and its emotion predicted by our short text emotion classifier, and the output is the probabilities of emotion classes that are appropriate for responding the post. We implement the method based on Kim-CNN again. After generating features from convolution layer in previous subsection, we feed them and emotion feature (one-hot vector) into a fully connected softmax layer with dropout to obtain the probability of emotion classes. Here we focus on probabilities of emotion classes, but not the one with maximum probability. We combine the post and

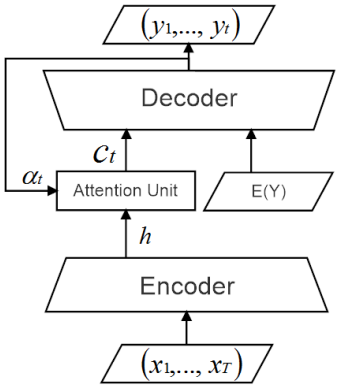


Figure 3: Structure of Encoder-decoder Generative Model

the emotion class of the corresponding comment as a pair for training, and the parameters remain same as emotion classifier.

2.2.4 Emotion-Aware Neural Response Generator

We propose a method to generate comments with a given emotion to a post. Inspired by neural responding machine [4], we employ the encoder-decoder framework to build a neural response generator. As Figure 3 shown, given the word sequence of the input post $X = (x_1, ..., x_T)$, the encoder builds the hidden representations set $h = (h_1, ..., h_T)$, which are fed to the attention unit to calculate the set of context vectors $c = (c_1, ..., c_t)$ with attention signals $\alpha = (\alpha_1, ..., \alpha_t)$ at time t. Then the decoder generates the word sequence $Y = (y_1, ..., y_t)$ as response with $E(Y)$ and context vectors set c respectively.

We use gated recurrent unit recurrent neural network (GRU-RNN) for encoder and decoder to capture long term memory and decrease training difficulty.

- **Encoder**

  We use bidirectional recurrent neural network as the encoder. The hidden state h_j for word x_j is the concatenation of the forward hidden states and backward hidden states: $h_j = [\overrightarrow{h_j}^T; \overleftarrow{h_j}^T]^T$.

- **Attention Unit and Decoder**

  Similar to the traditional attention module, the context vector c_t for generating t-th response word y_t is formed with the set of hidden states $h = (h_1, ..., h_T)$:

  $$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j$$

  Here the weight parameter α_{tj} is

  $$\alpha_{tj} = \frac{\exp(r_{tj})}{\sum_{k=1}^{T} \exp(r_{tk})}$$

  And $r_{tj} = a(s_{t-1}, h_j)$ is the alignment model calculating the importance of post words around x_j when generating response word y_t at time t based on j-th encode hidden state h_j and the previous RNN decoder hidden state s_{t-1} (gained at time $t-1$).

We append the emotion of output text $E(\mathbf{y})$ into the standard decoder of attention model. The probability of generating t-th word y_t is calculated as:

$$p(y_t|y_{t-1}, .., y_1, E(\mathbf{y}), \mathbf{x}) = g(y_{t-1}, s_t, c_t, e_y)$$

where e_y is the emotion embedding of $E(y)$, c_t is the context vector, y_t is the one-hot representation of response word, and g is a softmax activation function after obtaining the linear combination of those inputs. s_t is the decoder hidden state:

$$s_t = f(s_{t-1}, y_{t-1}, c_t, e_y)$$

Here f is the non-linear function, which is GRU in our case.

- **Result Fusion**

  We propose a ranking method to combine all the comments generated with different emotion classes. Given the post $\mathbf{x}$, a candidate response y and its emotion class $E(\mathbf{y})$, we generate the text by maximizing the average log-likelihood:

  $$\hat{l}(\mathbf{y}|E(\mathbf{y}), \mathbf{x}) = \frac{1}{t} \sum_{i=1}^{t} \log p(y_i|y_{i-1}, ..., y_1, E(\mathbf{y}), \mathbf{x})$$

  After obtained the emotion distribution of response $\mathbf{y}$ for the post $\mathbf{x}$ ($p(E(\mathbf{y})|\mathbf{x})$), we calculate the generation score of the result $\mathbf{y} = (y_1, ..., y_t)$ which contains the emotion $E(\mathbf{y})$ as follow:

  $$s(\mathbf{y}, E(\mathbf{y})|\mathbf{x}) = \log p(E(\mathbf{y})|\mathbf{x}) + \lambda \hat{l}(\mathbf{y}|E(\mathbf{y}), \mathbf{x})$$

  While we fuse all generation results into a list of responses, several post-processing methods can be applied. For example, since some words (ie. Xiaoming) stand for the names of persons and are easy to break the context of the post, we filter all these words while generating unless it is appeared in the post sequence; Furthermore, to generate uncommon comments, we can use a RNN language model to learning the common words and decrease their generating probabilities while decoding; We can also diversify the responses to obtain more different comments. We test these methods in the following experiments.

3. EXPERIMENTS

In this section, we first describe our runs for retrieval-based method and generation-based method respectively:

- MSRSC-C-R1: Only use image features in the post scoring. Instead of using fastrank learning model, we use the tf-idf weighted averaging to combine image vectors and calculate the cosine similarity of the given post p and the candidate post P_i.

- MSRSC-C-R2: Use full features and fastrank training in the ranking module, suppose to be a better run.

- MSRSC-C-R3: Only use text features for fastrank training in the post scoring.

- MSRSC-C-R4: Here we submit a run using knowledge obtained from the generation-based model. After matching module, we rank all the candidate comments by the generation score predicted from our emotion-aware generation model and keep top ten comments as the answer.
- MSRSC-C-R5: Only use image features for fastrank training in the post scoring.
- MSRSC-C-G1: Use all the post-processing methods in the fusion module: name filtering, RNN language model adjusting and diversification.
- MSRSC-C-G2: Use name filtering and RNN language model adjusting while decoding. The diversity is not taken into account.
- MSRSC-C-G3: use RNN language model adjusting but keep name words while decoding. Also use result diversification.
- MSRSC-C-G4: Do not use RNN language model adjusting or name filtering while decoding. Only keep result diversification.
- MSRSC-C-G5: We choose the local scheme of neural responding machine proposed by Shang et al. [4] as comparison. Our method is same as this model if we ignore emotion information. Also keep result diversification.

Here are some training details of our generation-based method. For training short text emotion classifier, we initialize the word vectors by random sampling from a uniform distribution between -0.1 and 0.1, and use 3,4,5 filter windows with 128 feature maps each. While training emotion-aware neural response generator, the vocabulary size is set to 40000, which can cover over 95% of words in posts and comments. We replace the words that are not covered by the token "<Unknown>". We implement the short text generator using Chainer[5], in which both encoder and decoder have 512 hidden units and the word embedding length is 200.

STC-2 use three measures the same as STC-1 for evaluation: $nDCG$@1, $nERR$@10 (Expected Reciprocal Rank) and P^+ (the bigger the better)[5].

The experiment results of five retrieval-based runs are shown in following Table 1:

Run	Mean nDCG@1	Mean P^+	Mean nERR@10
MSRSC-C-R1	0.1140	0.2207−	0.2208−
MSRSC-C-R2	0.1300	0.2498	0.2611
MSRSC-C-R3	0.1087	0.2274−	0.2378−
MSRSC-C-R4	**0.1767**	**0.2982**	**0.3104**
MSRSC-C-R5	0.1517	0.2263	0.2202−

Table 1: Official results of our retrieval-based methods. We conduct student t-test between MSRSC-C-R4 and other methods, "−" means that $p < 0.05$.

From the table, we can see that MSRSC-C-R2 outperforms other runs except for MSRSC-C-R4 as we expect. Comparing MSRSC-C-R2 with MSRSC-C-R3, we can infer that image features can bring some distinctive information in the ranking module and improve the overall performances. In the ranking module, MSRSC-C-R3 takes only text information and outperforms two runs which use only image information (MSRSC-C-R1 and MSRSC-C-R5). Here the text features seem to be effective than image features.

Comparing to MSRSC-C-R1, the results of MSRSC-C-R5 are slightly better, which shows that the fastrank algorithm is a better ranking method to some extent.

Among all retrieval-based runs, MSRSC-C-R4 obtains best performances for each metrics, which indicates the effectiveness of our emotion-aware generation model. The generation probabilities can also help to re-rank the comments.

The results of five generation-based runs are shown in following Table 2:

Run	Mean nDCG@1	Mean P^+	Mean nERR@10
MSRSC-C-G1	**0.1133**	0.1720	0.1609
MSRSC-C-G2	**0.1133**	0.1736	0.1659
MSRSC-C-G3	0.0750	0.1348−	0.1351−
MSRSC-C-G4	0.0987	**0.2168**	**0.2174**
MSRSC-C-G5	0.0670	0.1693	0.1604−

Table 2: Official results of our generation-based methods. We also conduct student t-test between MSRSC-C-G4 and other methods, "−" means that $p < 0.05$.

Comparing to baseline MSRSC-C-G5, all the results except for MSRSC-C-G3 improves visibly as we expect.That is to say, emotion information seems to be helpful in this task.

We then compare our four emotion-aware models in detail. We can see that MSRSC-C-G1 outperforms MSRSC-C-G3 for all the three metrics. It seems that name filtering is an effective post-processing method in the fusion module. Comparing MSRSC-C-G3 with MSRSC-C-G4, we may observe that RNN language model adjusting is not useful for three metrics. Furthermore, the similar evaluation results of MSRSC-C-G1 with MSRSC-C-G2 indicates that the effectiveness of diversification is not significant.

4. CONCLUSIONS

We in this report, introduce our methods of short text conversation task. For retrieval-based method, we propose a matching and ranking system to retrieve appropriate comments. First, a matching module is used to reduce the size of candidate comments. Then both text information and visual features are taken into account in the ranking module. For generation-based method, we propose an emotion-aware model to generate a response containing appropriate emotion. We apply neural networks approaches to classify short text emotions and predict suitable comment emotions for a given post. While generating short text responses, emotion information of comments are fed into decoder to separately generate comments with different emotions. At the end a fusion method ranks the generated comments and determine a response according to the comment emotion predictor. Empirical results show that both visual features and emotion information can improve retrieving or generation results.

Comparing to the top tier runs in the official result, our results seems not competitive. The probable reasons are as follows:

For retrieval-based method, we focus on introducing visual features into STC and may lose some effective text features.

[5]A flexible framework of neural networks for deep learning, http://chainer.org

Thus the performance of our matching module will be restricted. If we cannot retrieve appropriate comments from the repository, we obviously can never rank them to the top.

For generation-based method, we have not cleaned the training repository such as filtering advertisements and other meaningless samples. Furthermore, due to the time limit, our models may not be well trained and the more training epochs are necessary. In addition, some post-processing methods we use will lead to long and diverse results but may not be acceptable during evaluation.

We will improve our methods by fixing above weaknesses in the future works.

5. REFERENCES

[1] P. Ekman, W. V. Friesen, and P. Ellsworth. What emotion categories can observers judge from facial behavior? *Emotion in the Human Face*, pages 67–75, 1972.

[2] Z. Ji, Z. Lu, and H. Li. An information retrieval approach to short text conversation. *CoRR*, abs/1408.6988, 2014.

[3] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[4] L. Shang, Z. Lu, and H. Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015.

[5] L. Shang, T. Sakai, H. Li, R. Higashinaka, Y. Miyao, Y. Arase, and M. Nomoto. Overview of the NTCIR-13 short text conversation task. In *Proceedings of NTCIR-13*, 2017.

[6] O. Vinyals and Q. Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[7] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma. Topic aware neural response generation. In *AAAI*, pages 3351–3357, 2017.