Continuity of Topic, Interaction, and Query: Learning to Quote in Online Conversations

Lingzhi Wang[1,2], Jing Li[3], Xingshan Zeng[1,2]*, Haisong Zhang[4], Kam-Fai Wong[1,2]

[1]The Chinese University of Hong Kong, Hong Kong, China

[2]MoE Key Laboratory of High Confidence Software Technologies, China

[3]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

[4]Tencent AI Lab, China

[1,2]{lzwang,xszeng,kfwong}@se.cuhk.edu.hk

[3]jing-amelia.li@polyu.edu.hk, [4]hansonzhang@tencent.com

Abstract

Quotations are crucial for successful explanations and persuasions in interpersonal communications. However, finding what to quote in a conversation is challenging for both humans and machines. This work studies automatic quotation generation in an online conversation and explores how *language consistency* affects whether a quotation fits the given context. Here, we capture the contextual consistency of a quotation in terms of latent topics, interactions with the dialogue history, and coherence to the query turn's existing content. Further, an encoder-decoder neural framework is employed to continue the context with a quotation via language generation. Experiment results on two large-scale datasets in English and Chinese demonstrate that our quotation generation model outperforms the state-of-the-art models. Further analysis shows that topic, interaction, and query consistency are all helpful to learn how to quote in online conversations.

1 Introduction

Quotations, or quotes, are memorable phrases or sentences widely echoed to spread patterns of wisdom (Booten and Hearst, 2016). They are derived from the ancient art of rhetoric and now appearing in various daily activities, ranging from formal writings (Tan et al., 2015) to everyday conversations (Lee et al., 2016), all help us present clear, beautiful, and persuasive language. However, for many individuals, writing a suitable quotation that fits the ongoing contexts is a daunting task. The issue becomes more pressing for quoting in online conversations where quick responses are usually needed on mobile devices (Lee et al., 2016).

To help online users find what to quote in the discussions they are involved in, our work studies how to recommend an ongoing conversation with a quote and ensure its continuity of senses with the existing contexts. For task illustration, Figure 1 displays a Reddit conversation snippet centered around the worthiness to buy a Scuf controller. To argue against $T4$'s viewpoint, we see the query turn quotes *Tusser*'s old saying for showing that buying a controller is a waste of money. As can be observed, it is important for a quotation recommendation model to capture the key points being discussed (reflected by words like "money" and "dumb" here) and align them to words in the quotation to be predicted (such as "fool" and "money"), which allows to quote something relevant and consistent to the previous concern.

[$T1$]: Save your money. Scuf is the biggest ripoff in gaming.
[$T2$]: What would you suggest instead?
[$T3$]: Just use a normal controller.
[$T4$]: Ooooooh, I get it now...you're just dumb.
[Q]: The dumb ones are the people spending over $100 for a controller. *A fool and his money are soon parted.*

Figure 1: A Reddit conversation snippet about buying a Scuf controller. The quotation is in blue and italic. [$T1$] to [$T4$] are history turns while [Q] is for query turn.

To predict quotations, our work explores semantic consistency of what will be quoted and what was given in the contexts. In context modeling, we distinguish the query turn (henceforth **query**) and the other turns in earlier history (henceforth **history**), where topic, interaction, and query consistency work together to determine whether a quote fits the contexts. Here **topic consistency** ensures that the words in quotation reflect the discussion topic (such as "fool" and "money" in Figure 1). **Interaction consistency** is to identify the turns in history to which the query responds (e.g., $T1$ and $T4$ in Figure 1) and guide the

*Xingshan Zeng is the corresponding author.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6640–6650,
November 16–20, 2020. ©2020 Association for Computational Linguistics

quote to follow such interaction. **Query consistency** measures the language coherence of quote in continuing the story started by the query. For example, the quote in Figure 1 is to support the query's argument.

In previous work of quotation recommendation, there are many methods designed for formal writings (Tan et al., 2015; Liu et al., 2019); whereas much fewer efforts are made for online conversations with informal language and complex interactions in their contexts. Lee et al. (2016) use a ranking model to recommend quotes for Twitter conversations. Different from them, we attempt to generate quotations in a word-by-word manner, which allows the semantic consistency of quotes and contexts to be explored.

Concretely, we propose a neural encoder-decoder framework to predict a quotation that continues the given conversation contexts. We capture topic consistency with latent topics (i.e., word distributions), which are learned by a neural topic model (Zeng et al., 2018a) and inferred jointly with the other components. Interaction consistency is modeled with a turn-based attention over the history turns, and the query is additionally encoded to initialize the decoder's states for query consistency. To the best of our knowledge, we are the first to explore quotation generation in conversations and extensively study the effects of topic, interaction, and query consistency on this task.

Our empirical study is conducted on two large-scale datasets, one in Chinese from Weibo and the other in English from Reddit, both of which are constructed as part of this work. Experiment results show that our model significantly outperforms both the state-of-the-art model based on quote rankings (Lee et al., 2016) and the recent topic-aware encoder-decoder model for social media language generation (Wang et al., 2019a). For example, we achieve 27.2 precision@1 on Weibo compared with 24.0 by Wang et al. (2019a). Further discussions show that topic, interaction, and query consistency can all usefully indicate what to quote in online conversations. We also study how length of history and quotation affects the quoting results and find that we perform consistently better than comparison models in varying scenarios.

2 Related Work

Our work is in the line with content-based recommendation (Liu et al., 2019) or cloze-style reading comprehension (Zheng et al., 2019), which learns to put suitable text fragments (e.g., words, phrases, sentences) in the given contexts. Most prior studies explore the task in formal writings, such as citing previous work in scientific papers (He et al., 2010), quoting famous sayings in books (Tan et al., 2015, 2016), and using idioms in news articles (Liu et al., 2019; Zheng et al., 2019). The language they face is mostly formal and well-edited, while we tackle online conversations exhibiting noisy contexts and hence involving quote consistency modeling with turn interactions. Lee et al. (2016) also recommend quotations for conversations. However, they consider quotations as discrete attributes (for learning to rank) and hence largely ignore the rich information reflected by a quotation's internal word patterns. Compared with them, our model learns to quote with language generation, which can usefully exploit how words appear in both contexts and quotations.

For methodology, we are inspired by the encoder-decoder neural language generation models (Sutskever et al., 2014; Bahdanau et al., 2014). In dialogue domains, such models have achieved huge success in digesting contexts and generate microblog hashtags (Wang et al., 2019b), meeting summaries (Li et al., 2019), dialogue responses (Hu et al., 2019), etc. Here we explore how the encoder-decoder architecture works to generate quotations in conversations, which has never been studied in existing work. Our study is also related to previous research to understand conversation contexts (Ma et al., 2018; Liu and Chen, 2019; Sun et al., 2019), where it is shown to be useful to capture interaction structures (Liu and Chen, 2019) and latent topics (Zeng et al., 2019). For latent topics, we are benefited from the recent advance of neural topic models (Miao et al., 2017; Wang et al., 2019a)), which allows end-to-end topic inference in neural architectures. Nevertheless, none of the above work attempts to study the semantic consistency of quotes in conversation contexts, which is a gap our work fills in.

3 Our Quotation Generation Model

This section describes our neural encoder-decoder framework that generates quotations in conversations, whose architecture is shown in Figure 2. The encoding process works for context modeling of turn interactions (described in Section 3.1) and latent topics (presented in Section 3.2). For

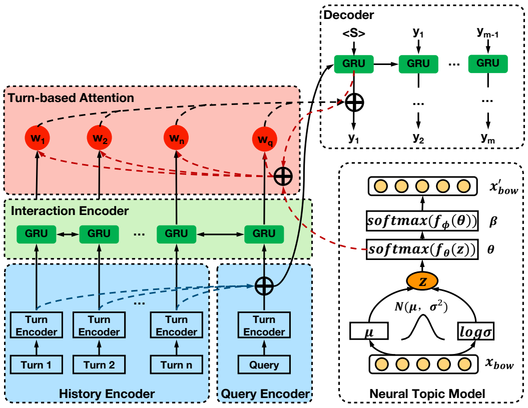


Figure 2: Our encoder-decoder framework for conversation quotation generation. It encodes turn interactions (for both query and earlier history) and latent topics in contexts. The decoder predicts quotes in aware of topic, interaction, and query consistency.

the decoding process to be discussed in Section 3.3, we predict words in quotes taking topic, interaction, and query consistency into consideration. The learning objective of the entire framework will be given at last in Section 3.4.

3.1 Interaction Modeling

To describe turn interactions, we first assume that there are m chronologically-ordered turns given as contexts and each turn T_i is formulated as a word sequence $\langle w_{i,1}, w_{i,2}, ..., w_{i,n_i}\rangle$ (n_i denotes the number of words). We consider the m-th turn as the query while others the history ($T_{history} = \langle T_1, T_2, ..., T_{m-1}\rangle$). Here we distinguish the query and its earlier history to separately explore the quote's language coherence to the query (for query consistency) and its interaction consistency to the earlier posted turns. In the following, we will describe how to encode history and query turns, and how the learned representations work together to explore conversation structure.

History Encoder. Here we describe how to encode turns in history. We first feed each word w_{ij} (the j-th word in the i-th turn) in history into an embedding layer to obtain its word vector $\boldsymbol{c}_{i,j}$. Then word vectors of the i-th turn $C_i = \langle \boldsymbol{c}_{i,1}, \boldsymbol{c}_{i,2}, ..., \boldsymbol{c}_{i,n_i}\rangle$ are further processed with a bidirectional gated recurrent unit (Bi-GRU) (Cho et al., 2014b). Its hidden states are defined as:

$$\overrightarrow{\boldsymbol{h}_{i,j}^{c}} = f_{GRU}(\boldsymbol{c}_{i,j}, \boldsymbol{h}_{i,j-1}^{c}),\ \overleftarrow{\boldsymbol{h}_{i,j}^{c}} = f_{GRU}(\boldsymbol{c}_{i,j}, \boldsymbol{h}_{i,j+1}^{c}) \tag{1}$$

The turn-level representations are hence captured by concatenating the last hidden states of both directions: $\boldsymbol{h}_i^c = [\overrightarrow{\boldsymbol{h}_{i,n_i}^c}; \overleftarrow{\boldsymbol{h}_{i,0}^c}]$. Further, we define the history representations as $\boldsymbol{h}^c = \langle \boldsymbol{h}_1^c, \boldsymbol{h}_2^c, ..., \boldsymbol{h}_{m-1}^c\rangle$, which will be further used to encode the interaction structure (described later).

Query Encoder. Similar to the way we encode each turn in history, a Bi-GRU is first employed to learn query representations $\boldsymbol{q} = \boldsymbol{h}_m^c$. Then, we identify which turns in history the query responds to for learning interaction consistency. To this end, we put a query-aware attention over the history turns and result in a context vector below:

$$\boldsymbol{c} = \sum_{i=i}^{m-1} \alpha_i \cdot \boldsymbol{h}_i^c, \quad \alpha_i = softmax(\boldsymbol{h}_i^c \cdot \boldsymbol{q}) \tag{2}$$

Afterwards, we enrich query representations with the features from history and obtain the history-aware query representations:

$$\tilde{\boldsymbol{q}} = W_q[\boldsymbol{q}; \boldsymbol{c}] + b_q \tag{3}$$

where W_q and b_q are learnable parameters.

Structure Encoder. With the representations learned above for query $\tilde{\boldsymbol{q}}$ and history $\boldsymbol{h}^c$, we can further explore how turns interact with their neighbors (henceforth conversation structure) with another Bi-GRU. It is fed with the $\langle \boldsymbol{h}_1^c, \boldsymbol{h}_2^c, ..., \boldsymbol{h}_{m-1}^c, \tilde{\boldsymbol{q}}\rangle$ sequence and the hidden states sequence $\langle \boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_{m-1}, \boldsymbol{h}_m\rangle$ is further put into a memory bank M for decoder's attentive retrieval in quotation generation (see Section 3.3).

3.2 Topic Modeling

Following the common practice (Blei et al., 2003; Miao et al., 2017), we model topics following the bag-of-words (BoW) assumption. Hence, we form a BoW vector $\mathbf{x}_{bow}$ (over vocabulary V) of the words in context to learn its discussion topic. The topic inference process is inspired by neural topic models (NTM) (Miao et al., 2017). It is based on a variational auto-encoder (VAE) (Kingma and Welling, 2013) involving an encoding and a decoding step to reconstruct the BoW of contexts.

BoW Encoding Step. This step is designed to learn a latent topic variable $\mathbf{z}$ from $\mathbf{x}_{bow}$. Here words in conversation contexts are assumed to satisfy a Gaussian distribution prior on mean μ and standard deviation σ (Miao et al., 2017). They are estimated by the following formula:

$$\mu = f_\mu(f_e(\mathbf{x}_{bow})),\ log\,\sigma = f_\sigma(f_e(\mathbf{x}_{bow})) \tag{4}$$

where $f_*(\cdot)$ is a neural perceptron performing a linear transformation activated with an ReLU function (Nair and Hinton, 2010).

BoW Decoding Step. Conditioned on the latent topic $\mathbf{z}$, we further generate words to form the BoW of each conversation $\mathbf{x}_{bow}$. Here we assume each word $w_n \in \mathbf{x}_{bow}$ is drawn from the conversation's topic mixture θ, which is a distribution vector over the topics. In the following, we show the generation story to decode $\mathbf{x}_{bow}$:

- Draw latent topic $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$.
- Topic mixture $\theta = softmax(f_\theta(\mathbf{z}))$.
- For the n-th word in the conversation:
    - Draw the word $w_n \sim softmax(f_\phi(\theta))$.

Here $f_*(\cdot)$ is a ReLU-activated neural perceptron defined above. The topic mixture θ will be later applied to capture topic consistency when predicting the quotation.

3.3 Quotation Generation

To predict the quotation $\boldsymbol{y}$, we first define the probability of words in it with the following formula:

$$Pr(\boldsymbol{y}|T_{history}, T_{query}) = \prod_{i=1}^{|\boldsymbol{y}|} Pr(y_i|\boldsymbol{y}_{<i}, M, \theta) \tag{5}$$

where $\boldsymbol{y}_{<i} = \langle y_1, y_2, ..., y_{i-1} \rangle$ and $|\boldsymbol{y}|$ denotes the quotation's word number. In prediction, the i-th word is generated with a likelihood $p_i = Pr(y_i|\boldsymbol{y}_{<i}, M, \theta)$, which is jointly determined by the words appearing before it ($\boldsymbol{y}_{<i}$) and the contexts features delivered by M (turn interactions described in Section 3.1) and θ (the discussion topic described in Section 3.2). Below comes more details of how we follow the semantic consistency of contexts to generate quotations.

Query Consistency. To carry on query's senses, the quotation is decoded with an unidirectional GRU initialized based on the encoded query. The initialization and later recursion of decoder's hidden states are given as:

$$\boldsymbol{h}_0^d = W_0\tilde{\boldsymbol{q}} + b_0,\ \boldsymbol{h}_i^d = f_{GRU}(\boldsymbol{v}_i, \boldsymbol{h}_{i-1}^d) \tag{6}$$

where W_0 and b_0 are parameters to be learned. $\boldsymbol{v}_i$ is the embedded decoder input to predict the i-th word in quotation.[1] In decoding, word prediction is conducted sequentially with beam search. It results in a ranking list of output, where we take the top K for quotation matching described later.

Topic and Interaction Consistency. For modeling quote consistency of discussion topics (with θ) and turn interactions (with M), we design a turn-based attention over conversation contexts to decode the quotation. Its attention weights are computed in aware of the structure-encoded turn representations $\boldsymbol{h}_j$ from M and topic distribution θ:

$$\alpha_{ij} = \frac{exp(f_d(\boldsymbol{h}_i^d, \boldsymbol{h}_j, \theta))}{\sum_{j'=1}^{m} exp(f_d(\boldsymbol{h}_i^d, \boldsymbol{h}_{j'}, \theta))} \tag{7}$$

where $f_d(\boldsymbol{h}_i^d, \boldsymbol{h}_j, \theta)$ captures the topic-aware semantic dependency the i-th word in quotation to the j-th turn in contexts and is defined as:

$$f_d(\boldsymbol{h}_i^d, \boldsymbol{h}_j, \theta) = W_d[\boldsymbol{h}_i^d \cdot \boldsymbol{h}_j^\theta] + b_d \tag{8}$$

where $\boldsymbol{h}_j^\theta = W_\theta[\boldsymbol{h}_j; \theta] + d_\theta$, and parameters W_d, b_d, W_θ, and d_θ are all trainable. Then we give the context vector $\boldsymbol{t}_i$ conveying both topic and interaction features for the i-th word to be generated:

$$\boldsymbol{t}_i = \sum_{j=1}^{m} \alpha_{ij}\boldsymbol{h}_j. \tag{9}$$

Finally, we predict the i-th word in quotation following the distribution p_i defined to combine topic, interaction, and query consistency:

$$p_i = softmax(W_p[\boldsymbol{h}_i^d; \boldsymbol{t}_i] + b_p), \tag{10}$$

where W_p and b_p are trainable parameters.

Quotation Matching. Occasionally language generation will "create" a non-existing quotation. To avoid that, we take a post-processing step for the outputs absent in our quotation list. Following previous practice (Liu et al., 2019), we select a quote from the list with the minimum edit distance (by tokens) and consider it as the final output.

3.4 Learning Objective

For the entire framework, we design its learning objective to allow joint learning of latent topics and conversation quotations:

$$\mathcal{L} = \mathcal{L}_{NTM} + \mathcal{L}_{QGM} \tag{11}$$

Here $\mathcal{L}_{NTM}$ is the objective function of neural topic model (NTM) defined as:

$$\mathcal{L}_{NTM} = D_{KL}(p(\mathbf{z})||q(\mathbf{z}\,|\,\mathbf{x})) - \mathbb{E}_{p(\mathbf{z})}[p(\mathbf{x}\,|\,\mathbf{z})] \tag{12}$$

where $D_{KL}(\cdot)$ is the Kullback-Leibler divergence loss and $\mathbb{E}_*[\cdot]$ reflects the reconstruction loss. [2]

[1] In training, we do teacher forcing and feed the gold standard. In test, we feed the predicted left neighbor.

[2] Because of the space limitation, we leave out the derivation details and refer the readers to Miao et al. (2017).

As for $\mathcal{L}_{QGM}$, it is defined as the cross entropy loss over all training instances to train the quotation generation model (QGM):

$$\mathcal{L}_{QGM} = -\sum_{n=1}^{N} log(Pr(\boldsymbol{y}_n|C_n, \theta_n)) \qquad (13)$$

where N is the number of training instances. $C_n = \{T_{history}, T_{query}\}_n$ represents the contexts of the n-th conversation and θ_n is C_n's topic composition induced by NTM.

4 Experimental Setup

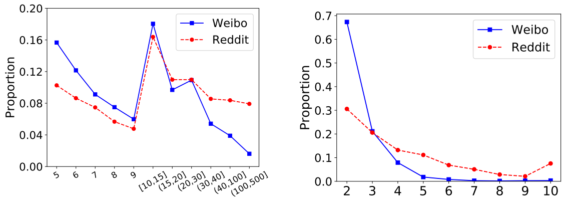
Datasets. For experiments, we construct two new datasets: one in Chinese from Weibo (a popular microblog platform in China and henceforth **Weibo**) and the other in English from Reddit (henceforth **Reddit**), which will be released upon publication. Here the raw Weibo data is released by Wang et al. (2019a) and Reddit obtained from a publicly available corpus.[3] For both Weibo and Reddit, we follow the common practice form conversations with posts and their comments (Li et al., 2015; Zeng et al., 2018b), where a post or comment is considered as a conversation turn.

To gather conversations with quotations, we maintain a quotation list and remove conversations containing no quotation from the list. For the remaining, if a conversation has multiple quotes, we construct multiple instances where one corresponds to the prediction of a quotation therein. On Weibo, we explore the quoting of Chinese *Chengyu*.[4] For Reddit, we obtain the quotation list from Wikiquote.[5] Afterwards, we remove conversation instances with quotations appearing less than 5 times to avoid sparsity (Tan et al., 2015). Finally, the datasets are randomly splitted into 80%, 10%, and 10%, for training, development, and test.

	Weibo	Reddit
# of quotes	1,053	1,111
Avg len of quotes	4.0	10.1
$\|$Voc$\|$ of quotes	1,251	4,111
# of convs	19,081	44,539
Avg # of turns per conv	2.51	4.25
Avg len of turn per conv	21.6	71.8
$\|$Voc$\|$ of convs	44,134	72,375

Table 1: Statistics of Weibo and Reddit datasets. The upper rows are for quotes and the lower rows are for conversations. The "len" refers to the number of tokens contained. "Avg # of turns" means the average turn number in context.

The statistics of the two datasets are shown in Table 1. We observe that the two datasets exhibit different statistics. For example, from the average turn number in contexts, we find Reddit users tend to quote in later turns while Weibo earlier. To further compare users' quoting behavior, we show the distribution of quotation number in Figure 3(a) and position in Figure 3(b). Figure 3(a) shows only a few quotations are commonly used in online conversations, probably because of its informal writing style. While for Figure 3(b), we find only a few Weibo conversations quote 5 turns later while the distribution on Reddit is much flatter.



(a) Frequency Distribution (b) Position Distribution

Figure 3: Quotation distribution over frequency (on the left) and position (right). X-axis: frequency (left) and context turn number (right); Y-axis: proportion of quotations (left) and conversations (right).

Preprocessing. To preprocess Weibo data, we adopted open-source Jieba toolkit[6] for Chinese word segmentation. For Reddit dataset, we employ natural language toolkit (NLTK[7]) for tokenization. In BoW preparation, all stop words and punctuation were removed following common practice to train topic models (Blei et al., 2003).

Parameter Setting. Here we describe how we set our model. In model architecture, the hidden size of all GRUs is set to 300 (bi-direction, 150 for each direction). For encoder, we adopt two layers of bidirectional GRU, and unidirectional GRU for decoder. The parameters in NTM are set up following Zeng et al. (2018a). For input, we set

[3] https://files.pushshift.io/reddit/comments/

[4] https://en.wikipedia.org/wiki/Chengyu Chengyu can be seen as a quotable phrase (Wang and Wang, 2013) — memorable rhetorical figures to convey wit and striking statement (Bendersky and Smith, 2012).

[5] https://en.wikiquote.org/wiki/Main_Page

[6] https://github.com/fxsjy/jieba

[7] https://www.nltk.org

the maximum turn length to 150 for Reddit and 200 for Weibo, and the maximum quotation length 20. Word embeddings are randomly initialized to 150-dimensional vectors. In model training, we employ Adam optimizer (Kingma and Ba, 2015), with $1e-3$ learning rate and the adoption of early stop (Caruana et al., 2001). Dropout strategy (Srivastava et al., 2014) is also used to avoid overfitting. We adopt beam search (beam size $=5$) to generate a ranking list for quote recommendation.

Evaluation Metrics. We first adopt recommendation metrics with popular information retrieval metrics Precision at K (P@K) and mean average precision (MAP) scores (Schütze et al., 2008) used. For P@K, K=1 to measure the top prediction, while for MAP we consider the top 5 outputs. Here we measure the generation models with their predictions after quotation matching (Section 3.3). Then, generation metrics are employed to evaluate word-level predictions. Here we consider both ROUGE (Lin, 2004) from summarization (F1 scores of ROUGE-1 and ROUGE-L are adopted) and BLEU (Papineni et al., 2002) from translation. To allow comparable results, generation models are measured with their original outputs (without quotation matching) while for ranking competitors, we take their top-1 ranked quotes.

Comparisons. We first adopt two weak baselines that select quotations unaware of the target conversation: 1) RANDOM: selecting quotations randomly; 2) FREQUENCY: ranking quotations with frequency. Then, we compared two ranking baselines: 3) non-neural learning to rank model (henceforth LTR) with handcrafted features proposed in Tan et al. (2015). 4) CNN-LSTM (Lee et al., 2016): previous quotation recommendation model (CNN for turn and quotation encoding and LSTM for conversation structure).

Next, we consider the encoder-decoder generation models without modeling conversation structure: 5) SEQ2SEQ (Cho et al., 2014a): using an RNN for encoding and another RNN for decoding; 6) TAKG: Seq2Seq framework incorporating latent topics for decoding. 7) the state-of-the-art (SOTA) model NCIR (Liu et al., 2019) designed for Chinese idiom generation.

Finally, the following of our variants are test: 8) IE ONLY: using interaction modeling results for decoding (w/o topic and query consistency modeling); 9) IE+QE: coupling interaction and query consistency (w/o NTM used for topic consistency); 10) IE+QE+NTM: our full model.

5 Experimental Results

In this section, we first show the main comparison results in Section 5.1. Then Section 5.2 discusses what we learn to represent consistency. Finally, Section 5.3 presents more analysis to characterize quotations in online conversations.

5.1 Main Comparison Results

Table 2 reports the main comparison results on two datasets, where our full model significantly outperforms all comparisons by a large margin. Several interesting observations can be drawn:

- *Quotation is related with context.* The poor performance of weak baselines reveals the challenging nature of quoting in online conversations. It is not possible to learn what to quote without considering context.
- *Generation models outperform Ranking.* Generation models in encoder-decoder style perform much better than ranking. It maybe attributed to generation model's ability to learn word-level mapping from source context to quotation.
- *Interaction, query, and topic consistency are all useful.* We see IE ONLY outperforms SEQ2SEQ, showing that interaction modeling helps encode indicative features from context. Likewise, the results of IE+QE are better than IE ONLY, and IE+QE+NTM better than IE+QE, both suggesting that learning query and topic consistency contribute to yield a better quotation.
- *Quoting in Reddit is more challenging than Weibo Chengyu.* All models perform worse on Reddit than Weibo. The possible reason is that Chinese Chengyu is shorter and renders a smaller vocabulary than English quotes (see Table 1).

5.2 Quotation and Consistency

We have shown our effectiveness in main results. Here we further examine our learned consistency and their effects on quoting. In the rest of this paper, without otherwise specified, *our model* is used as a short form of our full model (IE+QE+NTM). For comparison, we select TAKG for its best performance in Table 2 over all comparison models.

Interaction Consistency. To understand the positions of turns a quote is likely to respond to, we display the turn-based attention weights (Eq. 7) over turn position in Figure 4. Also shown is

Models	Weibo					Reddit				
	P@1	MAP	RG-1	RG-L	BLEU	P@1	MAP	RG-1	RG-L	BLEU
Weak Baselines										
RANDOM	0.2	0.7	2.1	2.1	0.2	0.1	0.7	5.6	4.5	0.1
FREQUENCY	2.3	6.9	3.1	3.1	2.3	1.0	4.7	1.7	1.5	1.0
Ranking Models										
LTR	3.6	9.3	5.1	5.1	3.6	1.7	7.1	4.1	3.6	1.7
CNN-LSTM	7.3	11.3	10.5	10.5	7.3	4.1	5.2	6.8	6.0	3.7
Generation Models										
SEQ2SEQ	19.9	24.1	22.6	22.5	19.9	7.2	9.8	11.7	10.6	4.7
TAKG	24.0	27.3	26.8	26.7	24.0	12.5	16.0	15.7	14.4	6.7
NCIR	22.6	26.5	25.3	25.2	22.6	7.3	12.2	10.9	9.9	4.1
Our models										
IE ONLY	21.5	24.8	24.5	24.4	21.5	11.2	14.6	13.9	12.8	5.7
IE+QE	22.0	24.7	25.2	25.1	22.0	13.5	17.4	17.0	15.5	7.0
IE+QE+NTM	**27.2**‡	**31.6**†	**29.5** ‡	**29.5**‡	**27.2**‡	**17.5**†	**24.0**†	**20.3**†	**18.8**†	**9.5**†

Table 2: Comparison results on Weibo and Reddit datasets (in %). RG-1 and RG-L refer to ROUGE-1 and ROUGE-L respectively. The best results in each column are in **bold**. Our full model IE+QE+NTM achieves significantly better performance than all the comparisons (paired t-test. ‡: $p < 0.05$; †: $p < 0.01$)

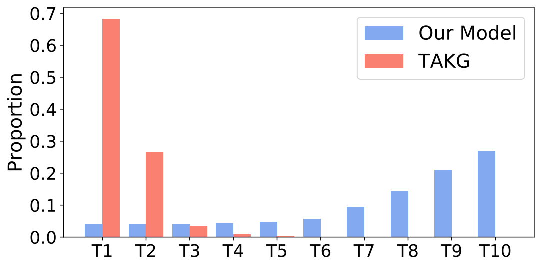


Figure 4: Attention weights over turns. X-axis: turn position; Y-axis: the normalized weight.

	Human 1	**Human 2**
Ground Truth	84	78
IE ONLY	36	32
IE+QE	49	46

Table 3: Human evaluation results for the quote coherent with query (count out of 100).

the attention weights from TAKG (Wang et al., 2019a) for comparison. Here we use Reddit conversations for interpretation because they involve larger turn number (see Table 1). It is seen that TAKG can only attend the first three turns while we assign higher weights to turns closer to query. In doing so, the quotes will continue senses from later history, which fits our intuition that participants tend to interact with latest information.

Query Consistency. We carry out a human evaluation to test the coherence of query and the predicted quotations. 100 conversations are sampled from Weibo and two native Chinese speakers are invited to examine whether a quote carry on the query's senses ("yes") or not ("no"). Table 3 shows the count of "yes" for the ground truth quote and the output of IE ONLY and IE+QE. Interestingly, even ground truth quotations cannot attain over 85% "yes", probably because of the prominent misuse of quotations on social media. Nevertheless, the better performance of IE+QE compared with IE ONLY shows the usefulness to model query consistency for ensuring quotation's language coherence to the query.

Topic Consistency. Here we use the example in Figure 1 to analyze the topics we learn for modeling consistency. Recall that the conversation centers around *price and value* and the quote is used to argue that *only fools will waste the money*. We look into the top 3 latent topics (by topic mixture θ) and display their top 10 words (by likelihood) in Table 4. There appears words like "pay" and "stupied", which might help to correctly predict "fool" and "money" in the quote.

5.3 Sensitivity to Context and Quotations

In this section, we study how varying context and quotations affect our performance.

The Effects of Context. Here we examine whether longer context will result in better results.

Topic 1	*game* property child rights *pay* guy state church guys *paid*
Topic 2	fuck evidence shit guys *stupid* edit nice proof dude *dumb*
Topic 3	car *buy* cops police scrubs gun technology shot crime energy

Table 4: The top 10 words of the 3 latent topics related to the conversation in Figure 1. Words suggesting conversation's focus are in blue and italic.

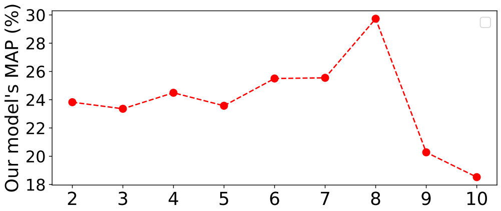


Figure 5: Our MAP scores over conversations with varying turn number. X-axis: turn number; Y-axis: MAP scores. The best results are seen for 8-turn convs.

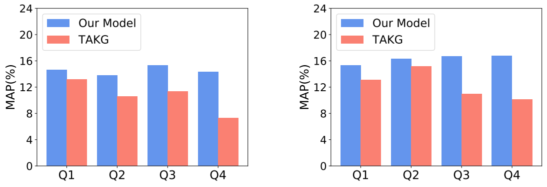
In the following, we measure context length in terms of turn number and token number.

Turn Number. Figure 5 shows our MAP scores to quote for Reddit conversations with varying turn number. Weibo results are not shown here for the limited data with turn number > 4. Generally, more turns result in better MAP, for the richer information to be captured from turn interactions. The scores drop for turn number > 8, probably because of underfitting and a more complex model might be needed for interaction modeling.

To further explore model's sensitivity to turn number, we first rank the conversations with turn number and separate them into four quartiles (Q_1, Q_2, Q_3, Q_4, in order with increasing turn number). We then train and test in each quartile, and compare the results of our model and TAKG in Figure 6(a). As can be seen, our model presents larger margin for quartiles corresponding to larger turn number, indicating our ability to encode rich information from complex turn interactions.

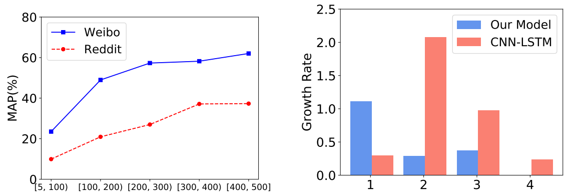
Token number. For context length measured with token number, we follow the above steps to form train and test quartiles for token number. The results are shown in Figure 6(b) where our model consistently outperform TAKG over conversation context with varying token number.

The Effects of Quotation. We further study our results to predict quotations in varying frequency and the MAP scores are reported in Figure 7(a). In general, higher scores are observed for more frequent quotations, as better representations can be extensively learned from training data. We also notice a slower growing rate as the frequency increases. To go into more details, we compare the growing rates with ranking model CNN+LSTM and show the results in 7(b) on Weibo (Reddit results in similar trends). In comparison, we are generally less sensitive to quotation frequency (except for very rare quotes). It is likely to be benefited from quotations' internal structure while ranking models can be largely affected by label sparsity.



(a) Turn Number (b) Token Number

Figure 6: MAP scores (y-axis) over context length (left in turn number and right token number) in varying quantiles. For each subfigure, from left to right shows the results in Q_1 ($[0, 0.25)$), Q_2 ($[0.25, 0.5)$), Q_3 ($[0.5, 0.75)$), and Q_4 ($(0.75, 1)$).



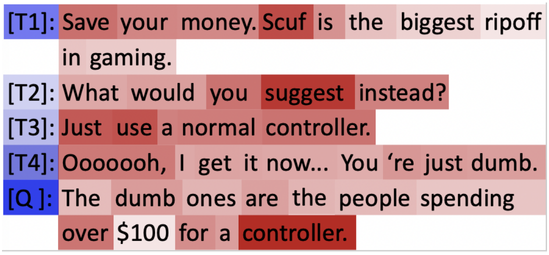
(a) MAP on Freq. Ranges (b) Growth Rate Comparison

Figure 7: Left subfigure: Our MAP scores (y-axis) on different frequency range (x-axis). Right: growing rate (y-axis) on Weibo, where x-axis indicates the order of neighboring ranges.

5.4 Further Discussions

Here we probe into our outputs to provide more insights to quoting in conversations.

Case Study. We first present a qualitative analysis over the example in Figure 1. To analyze what the model learns, we visualize our turn-based attention and TAKG's topic-aware attention over words in Figure 8. As can be seen, TAKG focuses more on topic words "Scuf", "suggest", and "controller", all reflecting the global discussion focus



[T1]: Save your money. Scuf is the biggest ripoff in gaming.
[T2]: What would you suggest instead?
[T3]: Just use a normal controller.
[T4]: Ooooooh, I get it now... You 're just dumb.
[Q]: The dumb ones are the people spending over $100 for a controller.

Figure 8: Attention weights of our model over turns (in blue) and TAKG over words (in red).

while ignoring query's intention. Thus, it mistakenly quote "*A penny saved is a penny earned.*". Instead, we attend the query's interaction with T_1 and T_4, which results in the correct quotation.

Comparing with Human. Finally, we discuss how human performs on our task. 50 Weibo conversations were hence sampled and two human annotators (native Chinese speakers) were invited to quote a Chinese Chengyu in the given context. The two annotators give 7 and 8 correct answers respectively, which shows the task is challenging for human. Our model made 13 correct predictions, exhibiting a better ability to quote in online conversations.

6 Conclusion

We present a novel quotation generation framework for online conversations via the modeling of topic, interaction, and query consistency. Experiment results on two newly constructed online conversation datasets, Weibo and Reddit, show that our model outperforms the previous state-of-the-art models. Further discussions provide more insights on quoting in online conversations.

Acknowledgements

The research described in this paper is partially supported by CUHK RSFS #3133237, HK RGC GRF #14204118, HK ITF #ITS/335/18. Jing Li is supported by the Hong Kong Polytechnic University internal funds (1-BE2W and 1-ZVRH) and NSFC Young Scientists Fund 62006203.

References

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

Michael Bendersky and David A. Smith. 2012. A dictionary of wisdom and wit: Learning to extract quotable phrases. In *Proceedings of the Workshop on Computational Linguistics for Literature, CLfL@NAACL-HLT 2012, June 8, 2012, Montréal, Canada*, pages 69–77.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Kyle Booten and Marti A. Hearst. 2016. Patterns of wisdom: Discourse-level style in multi-sentence quotations. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1139–1144.

Rich Caruana, Steve Lawrence, and C Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014a. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. 2010. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430. ACM.

Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. GSN: A graph-structured network for multi-party dialogues. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5010–5016.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Diederik P. Kingma and Max Welling. 2013. Autoencoding variational bayes. *CoRR*, abs/1312.6114.

Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 957–960, New York, NY, USA. ACM.

Jing Li, Wei Gao, Zhongyu Wei, Baolin Peng, and Kam-Fai Wong. 2015. Using content-level structures for summarizing microblog repost trees. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2168–2178.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2190–2196.

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

Yuanchao Liu, Bo Pang, and Bingquan Liu. 2019. Neural-based Chinese idiom recommendation for enhancing elegance in essay writing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zhengyuan Liu and Nancy Chen. 2019. Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Kaixin Ma, Tomasz Jurczyk, and Jinho D. Choi. 2018. Challenging reading comprehension on daily conversation: Passage completion on multiparty dialog. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2410–2419, International Convention Centre, Sydney, Australia. PMLR.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA. Omnipress.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*, page 260.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *CoRR*, abs/1902.00164.

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

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015. Learning to recommend quotes for writing. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A neural network approach to quote recommendation in writings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16.

Lanchun Wang and Shuo Wang. 2013. A study of idiom translation strategies between english and chinese. *Theory and practice in language studies*, 3(9):1691.

Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. 2019a. Topic-aware neural keyphrase generation for social media language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yue Wang, Jing Li, Irwin King, Michael R. Lyu, and Shuming Shi. 2019b. Microblog hashtag generation via encoding conversation contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1624–1633.

Jichuan Zeng, Jing Li, Yulan He, Cuiyun Gao, Michael R. Lyu, and Irwin King. 2019. What you say and how you say it: Joint modeling of topics and discourse in microblog conversations. *TACL*, 7:267–281.

Jichuan Zeng, Jing Li, Yan Song, Cuiyun Gao, Michael R Lyu, and Irwin King. 2018a. Topic memory networks for short text classification. *arXiv preprint arXiv:1809.03664*.

Xingshan Zeng, Jing Li, Lu Wang, Nicholas Beauchamp, Sarah Shugars, and Kam-Fai Wong. 2018b. Microblog conversation recommendation via joint modeling of topics and discourse. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 375–385.

Chujie Zheng, Minlie Huang, and Aixin Sun. 2019. Chid: A large-scale chinese idiom dataset for cloze test. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 778–787.