NUT-RC: Noisy User-generated Text-oriented Reading Comprehension

Rongtao Huang[1], **Bowei Zou**[2]*, **Yu Hong**[1],
Wei Zhang[3], **Ai Ti Aw**[2], **Guodong Zhou**[1]
[1]School of Computer Scienceand Technology, Soochow University, China
[2]Aural & Language Intelligence Department, Institute for Infocomm Research, Singapore
[3]Alibaba Group, China
rthuang.suda@gmail.com, {zou_bowei,aaiti}@i2r.a-star.edu.sg,
lantu.zw@alibaba-inc.com, {yhong,gdzhou}@suda.edu.cn

Abstract

Reading comprehension (RC) on social media such as Twitter is a critical and challenging task due to its noisy, informal, but informative nature. Most existing RC models are developed on formal datasets such as news articles and Wikipedia documents, which severely limit their performances when directly applied to the noisy and informal texts in social media. Moreover, these models only focus on a certain type of RC, extractive or generative, but ignore the integration of them. To well address these challenges, we come up with a noisy user-generated text-oriented RC model. In particular, we first introduce a set of text normalizers to transform the noisy and informal texts to the formal ones. Then, we integrate the extractive and the generative RC model by a multi-task learning mechanism and an answer selection module. Experimental results on TweetQA demonstrate that our NUT-RC model significantly outperforms the state-of-the-art social media-oriented RC models.

1 Introduction

Reading Comprehension (RC), which aims to answer questions by comprehending the contexts of given passages, is a frontier topic in natural language processing research. Recently, many RC models (Wang et al., 2018; Lin et al., 2018; Zhu et al., 2018; Joty et al., 2018; Weber et al., 2019) have been proposed and have achieved considerable successes. According to answer prediction methods, the RC models can be roughly divided into two major categories: *extractive* and *generative*. For an extractive RC model, the predicted answer is limited to be a consecutive span in the passage. For a generative RC model, it allows the answer to be free-text which can include novel words and phrases not appeared in passages.

However, most of the existing RC models are developed for formal text, such as news articles (Hermann et al., 2015; Trischler et al., 2017) and Wikipedia (Rajpurkar et al., 2016; Joshi et al., 2017), which severely limit their performances on Noisy User-generated Texts (NUT). Meanwhile, an increasing number of people are accustomed to getting real-time information via social media like Twitter. Recently, Xiong et al. (2019) propose a large-scale dataset for question answering over social media texts, TweetQA, which is constructed in a crowd-sourcing way, and recommended annotators to write answers in their own language.

Table 1 shows an example from the TweetQA dataset. In order to answer the question "*what is kerry thankful for?*", an RC model first needs to be positioned to "*Thank u 4 opening this convo*", where "*u, 4, convoy*" is the abbreviation of "*you, for, conversation*", respectively. Furthermore, "*u*" refers to "*@InStyle*". Then the RC model can predict the answer "*instyle opening the conversation*". Such an example indicates that question answering targeting on social media text like tweet is challenging, not only because of its informal nature of the noisy user-generated text (e.g., abbreviation and multiple independent short sentences) but also from the tweet-special representations (e.g., the object whom "Kerry washington" wants to thank is "InStyle" because he is mentioned by "Kerry" at the beginning of the tweet).

[*] corresponding auther

Proceedings of the 28th International Conference on Computational Linguistics, pages 2687–2698
Barcelona, Spain (Online), December 8-13, 2020

Tweet: *@InStyle: On KWs Cover: Beautiful statement. Thank u 4 opening this convo. Its an important 1 that needs to be had.— kerry washington (@kerrywashington) February 5, 2015*
Q: *what is kerry thankful for?* **A**: *instyle opening the conversation*

Table 1: An example of reading comprehension targeting noisy user-generated social media text from the TweetQA dataset. Note that the highlighted terms are the tweet-specific informal tokens.

To address the above challenges, we come up with a Noisy User-generated Text-oriented Reading Comprehension model (NUT-RC), which takes advantage of both the existing extractive and generative RC models for formal texts, and extends to the informal texts by text normalization. In particular, we first convert the noisy user-generated texts in social media to the formal texts via text normalization, which can benefit to improving the theoretical upper bound of performance of the RC models. Second, in TweetQA, many answers are with the free-text format. For example, the answer in Table 1, "*instyle opening the conversation*", is not a contiguous substring of the corresponding tweet, which indicates that there is a ceiling of the TweetQA dataset for an extractive RC model. Therefore, we propose a generative RC model with multi-task learning. Unlike the traditional encoder-decoder framework, followed Dong et al. (2019), we use shared transformer blocks and a specially designed self-attention matrix mask to control what context the answer generation conditions on. Then we enable the combination of the extractive RC model and the generative RC model by a multi-task learning mechanism. By sharing the underlying representations, the generative RC model gains the ability of the extractive RC model. Finally, an answer selection module is used to choose the more appropriate answer from the extractive RC model or the generative RC model.

The experiments are carried out on the TweetQA dataset. The performances of our NUT-RC model achieve to 76.1%, 72.1%, and 77.9% of BLEU-1, Meteor, Rouge-L, respectively, yielding absolute improvements of 14.7% on BLEU-1, 13.5% on Meteor, and 13.8% on Rouge-L over the baseline. Note that the performance is also exceeding the human on both Meteor and Rouge-L metrics.

Our contributions in this paper are three-folds: (1) proposal of a novel noisy user-generated text-oriented RC model which combines the extractive and the generative RC models by an answer selection model; (2) proposal of regarding the extractive RC as an auxiliary task to optimize the generative RC model by multi-task learning; and (3) empirical verification of the effectiveness of the model and achieving the state-of-the-art performance on TweetQA.

2 Related Work

Reading Comprehension Reading comprehension (RC) aims to teach a machine answering questions by comprehending the context of given passages, which is one of the most important tasks in the NLP community automatically. The prior research on RC mainly focuses on either the cloze-style (Hermann et al., 2015; Hill et al., 2015; Seo et al., 2016) or multiple-choice (Richardson et al., 2013; Lai et al., 2017). However, these models are difficult to be directly utilized for real application scenarios. Recently, many large-scale RC datasets constructed by a crowdsourced way (Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018) have been proposed and received widespread attentions (Wang et al., 2017; Min et al., 2019). Besides, to make RC models have the ability of conversation understanding like a human being, more challenging multi-round conversational RC datasets are proposed, such as QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2019). After Transformer (Vaswani et al., 2017) has been released, various pre-trained models (Devlin et al., 2019; Yang et al., 2019; Dong et al., 2019) have sprung up and achieved promising results on most of the RC datasets through purely fine-tuning. The parameters of our NUT-RC model are also initialized by BERT (Devlin et al., 2019).

According to the original formats of the predicted answers, the existing RC model can be roughly divided into two major categories: *extractive* and *generative*. In an extractive RC model, the answers are limited to be a span of the given passages. Most of the mainstream RC models are the extractive RC model. In a generative RC model, the answers can be free-text and do not have to appear in the passages.

Normalizer	Informal type	Example
SPLT	Mixed tokens	*#InTheUnlikelyEvent* (*# In The Unlikely Event*)
EXPN	Abbreviation	*u* (*you*), *convo* (*conversation*), *addr* (*address*)
WDLK	Misspelling word	*preety* (*pretty*), *goverment* (*government*)
MISC	Other	*sh*t* (·), :-) (·)

Table 2: Taxonomy of informal texts and non-standard word normalizers. The revised texts are in parentheses; "(·)" denotes deletion.

With the development of natural language generation technology, researchers focused on using generative models to solve reading comprehension problems. For example, McCann et al. (2018) and Bauer et al. (2018) used RNN-based pointer generation mechanisms to generate answers from a single document. Tan et al. (2018) adopted a pipeline method in multi-document reading comprehension. In this paper, we propose a multi-task learning based generative RC model that integrates the advantages of both the extractive and the generative RC models by share hidden state representations.

Noisy User-generated Text-oriented NLP Recently, due to the increasing number of social media users, many research directions of NLP are required for processing the noisy user-generated texts in social media, such as tweets. The prior works mainly concentrate on part-of-speech tagging and dependency parser in tweets. For example, Foster et al. (2011) annotated 7,630 PoS tags according to the Penn Treebank (Marcus et al., 1993); Kong et al. (2014) built a dependency parser for tweets based on the TWEEBANK, which is the first dataset annotated with syntactic information on tweets. Besides, the ACL-IJCNLP Workshop (Baldwin et al., 2015) provides a shared task on noisy user-generated text processing, including twitter lexical normalization and named entity recognition, which stimulated many studies on this topic (Godin et al., 2015; Flint et al., 2017; Liu et al., 2018). Recently, Xiong et al. (2019) released the first large-scale RC dataset over social media data, TweetQA, which gathered tweets used in news articles and encouraged human annotators to write questions and answers upon these tweets in their language. To the best of our knowledge, there are rare reading comprehension models over social media, and the proposed NUT-RC model is inspired by TweetQA.

3 Noisy User-Generated Text-oriented Reading Comprehension

Figure 1 illustrates the architecture of the noisy user-generated text-oriented reading comprehension model (NUT-RC), which comprises six basic components: 1) *text normalization* to convert raw noisy and informal tweets to formal texts (Subsection 3.1), 2) *lexicon encoder* to map the input into a sequence of input embedding vectors (Subsection 3.2), 3) *transformer with self-attention masks* to map the input embeddings into contextual hidden representations (Subsection 3.3), 4) *extractive RC model* identify the start and end positions of an answer in a tweet (Subsection 3.4), 5) *generative RC model* with multi-task learning to generate an answer from the vocabulary conditioned on the question and the tweet (Subsection 3.5), and 6) *answer selection* to choose the most appropriate answer from the extractive RC model and the generative RC model (Subsection 3.6).

3.1 Text Normalization

One challenge in reading comprehension for social media is its informal nature of the noisy user-generated texts. Following Flint et al. (2017), we first convert the noisy and informal tweets to their corresponding formal format by text normalization. Table 2 shows the taxonomy of informal texts and the normalization of common Non-Standard Words (NSW) in the TweetQA dataset. We apply four kinds of text normalizers to clean informal texts. First, hashtags ("*#InTheUnlikelyEvent*") and user_ids ("*@JasonLloydNBA*") in tweets are often represented as a single mixed-token, thus the SPLT normalizer is used to divide the mixed-token into separated words. Then, because of the quick typing of posters, various abbreviations and typos exist widely in social media applications. For these tokens, we use the EXPN normalizer to replace abbreviations with their associated expansions and apply the WDLK normalizer to correct the misspelled tokens. Finally, there are also many profanities and non-standard punctuation in tweets, which might be useful for emotion analysis rather than factoid reading comprehension. Thus we

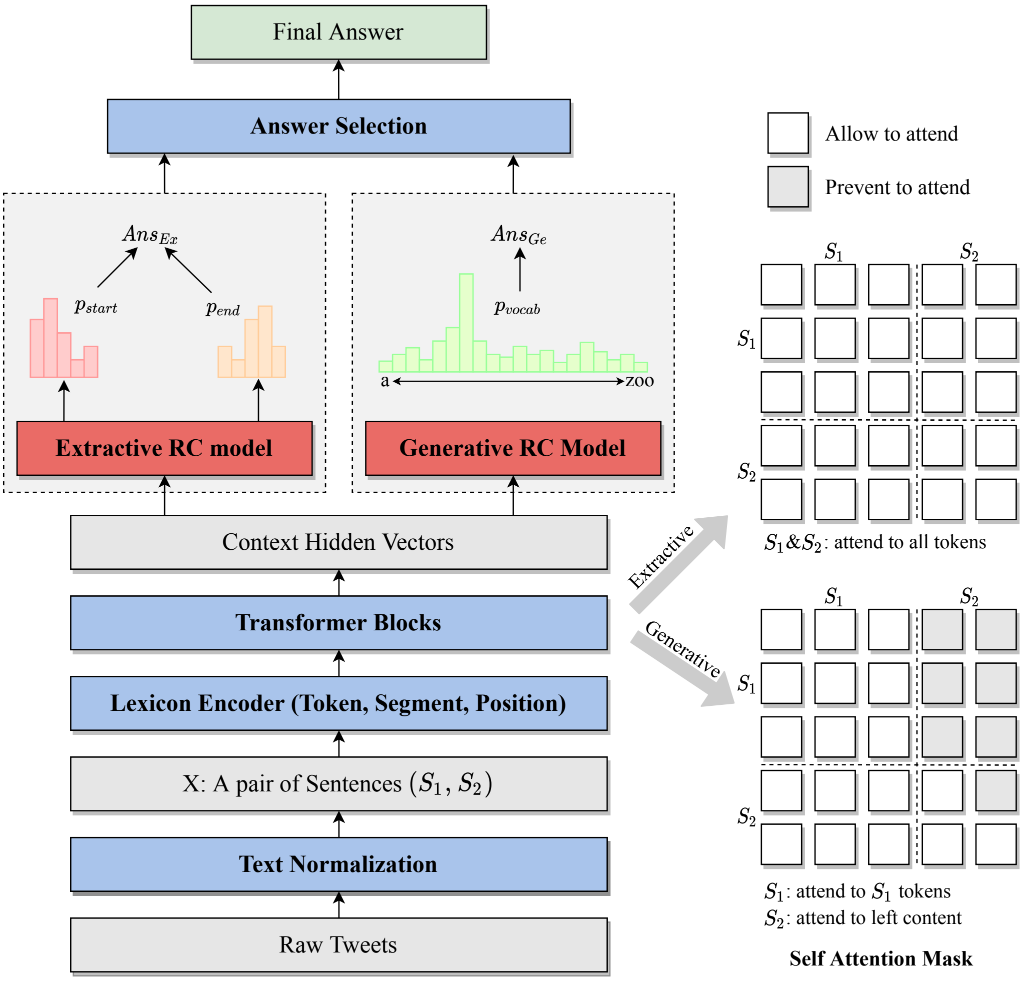

Figure 1: Architecture of NUT-RC.

remove these expressions by the MISC normalizer. Noting that when cleaning tweets, we use the SPLT normalizer ahead of the other three ones.

3.2 Lexicon Encoder

After text normalization, the lexicon encoder generates the input embeddings by a word sequence X consisting of a pair of segments S_1 and S_2. For the extractive RC model, the source segment (S_1) and the target segments (S_2) are the normalized tweet and the question, respectively. For the generative RC model, the normalized tweet and the question are packed together as the source segment S_1, and the generated answer is the target segment S_2. The input X is tokenized to sub-word units $\{x_i\}_{i=1}^n$ by the WordPiece (Wu et al., 2016), where n is the length of the input sequence. Following Devlin et al. (2019), the first token x_1 is always the "[CLS]" token which is encoded as the hidden embeddings by transformers to represent the full input information. Moreover, a special token "[SEP]" is used to separate the segment pair (S_1, S_2) (for the generative QA model, such token is also used as the end tag of the generated answer). For each sub-word piece, its input embeddings are initialized by the sum of the token embeddings, the segment embeddings, and the position embeddings.

3.3 Transformer with Self-attention Masks

Transformer We employ an L-layer Transformer (Vaswani et al., 2017) to map the input embeddings into a sequence of contextual embeddings

$$\boldsymbol{H}^l = Transformer_l(\boldsymbol{H}^{l-1}), \tag{1}$$

where $l \in [1, L]$, $\boldsymbol{H}^l = [\boldsymbol{h}_1^l, ..., \boldsymbol{h}_n^l]$, and $\boldsymbol{h}_i^l$ denotes the contextualized hidden state of the i-th embeddings of the input in the l-th layer. Note that, the transformer blocks are implemented as a neural decoder with a special self-attention mask, and also can share the parameters across the extractive RC model and the generative RC model through multi-task learning (Section 3.5).

Self-attention Masks For each layer of the transformer, the self-attention (scaled dot-product and multi-head attention) is adopted to integrate the output vectors from previous layers. Following Dong et al. (2019), we utilize the special-attention masks $\boldsymbol{M}$ to control what contexts the current token can attend to. The self-attention $\boldsymbol{A}_l$ can be computed as

$$\begin{aligned} \boldsymbol{Q} = \boldsymbol{H}^{l-1}\boldsymbol{W}_l^Q, \quad & \boldsymbol{K} = \boldsymbol{H}^{l-1}\boldsymbol{W}_l^K \\ M_{ij} = & \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \\ \boldsymbol{A}_l = softmax(&\tfrac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}} + \boldsymbol{M})(\boldsymbol{H}^{l-1}\boldsymbol{W}_l^V), \end{aligned} \tag{2}$$

where the $\boldsymbol{H}^{l-1} \in \mathbb{R}^{n \times d_h}$ denotes the linear combination of the triple of (query, key, value) via the parameter matrices $\boldsymbol{W}_l^Q, \boldsymbol{W}_l^K, \boldsymbol{W}_l^V \in \mathbb{R}^{d_h \times d_k}$, respectively, and the self-attention mask matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ represents whether a pair of tokens can be mutually attended.

As shown in Figure 1, for the extractive RC model, the self-attention mask $\boldsymbol{M}$ is set to a zero matrix, denoting that all tokens in tweet and question are allowed to attend over each other. This is consistent with the bidirectional language model like BERT (Devlin et al., 2019). For the generative RC model, the left elements of the mask matrix are all 0s, the upper-right elements are set to $-\infty$, and the bottom-right of the mask matrix is an upper-triangular matrix. While all tokens in the source have access to each other, the tokens in the target can attend to all tokens in the source, the preceding tokens in the target sequence, and themselves. In this way, we implement a generative RC model with a bidirectional encoder and a unidirectional decoder via shared transformer blocks.

3.4 Extractive RC Model

Given a question and the corresponding tweet, the extractive RC model aims to extract the correct answer span from this tweet. By the aforementioned transformer encoder, we first get the last hidden states of both the tweet and the question $\boldsymbol{H}_{Ex}^L = [\boldsymbol{h}_1^L, ..., \boldsymbol{h}_n^L]$. Then, a Point Network (Vinyals et al., 2015) is adopted to get the possibility vectors of both the start position ($\boldsymbol{s} \in \mathbb{R}^{d_h}$) and the end position ($\boldsymbol{e} \in \mathbb{R}^{d_h}$). The probabilities of the word w_i as the start position p_i^s and the end position p_i^e of the answer span are calculated by

$$p_i^s = \frac{e^{\boldsymbol{s} \cdot \boldsymbol{h}_i^L}}{\sum_j e^{\boldsymbol{s} \cdot \boldsymbol{h}_j^L}}, \quad p_i^e = \frac{e^{\boldsymbol{e} \cdot \boldsymbol{h}_i^L}}{\sum_j e^{\boldsymbol{e} \cdot \boldsymbol{h}_j^L}}. \tag{3}$$

The objective of the extractive RC model is maximizing the log-likelihood of the start position and the end position. The loss is calculated by

$$L_{ex} = -\frac{1}{N}\sum_{i=1}^{N}(log\ p_{y_i^s}^s + log\ p_{y_i^e}^e), \tag{4}$$

where N is the number of samples in the training set, and y_i^s, y_i^e are the start position and the end position of the ground truth answer span, respectively. Note that y_i^e is constrained to be bigger than y_i^s during the prediction.

3.5 Generative RC Model

The generative RC model aims to generate a well-formed word sequence as the answer from the vocabulary, conditioned on the question and the corresponding tweet. First, we pack the question, the tweet, and the answer into a single sequence like "[CLS] *tweet* [SEP] *question* [SEP] *answer* [SEP]". The tweet and the question are combined as the source segment S_1, and the answer is the target segment S_2. Then, according to Subsection 3.3, we get the contextual hidden representations.

During training, we follow Dong et al. (2019), which randomly masks tokens in the target segment S_2 with a special token "[MASK]". Then, the model learns how to recover these masked tokens. The probability distribution over all words in the vocabulary $\boldsymbol{p}_{vocab}$ can be produced by a linear and softmax layer:

$$\boldsymbol{p}_{vocab} = softmax(\boldsymbol{V}\boldsymbol{H}^L_{Ge} + \boldsymbol{b}), \tag{5}$$

where $\boldsymbol{H}^L_{Ge}$ denotes the last hidden states obtained by the shared transformer blocks, conditioned on the source segment S_1 and the masked target segment S_2, $\boldsymbol{V}$ and $\boldsymbol{b}$ are the learnable parameters. Thus, we obtain the probability of the predicted word w_t by

$$p(w_t) = \boldsymbol{p}_{vocab}(w_t). \tag{6}$$

The loss function of the generative RC model is calculated by the sum of negative log-likelihood of the masked target word sequence $\{w^i_t\}^T_{t=1}$:

$$L_{ge} = -\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} log\, p(w^i_t), \tag{7}$$

where N denotes the number of training samples, and T is the length of masked tokens in the target segment.

Multi-task Learning As described above, while the extractive RC model requires that the predicted answer must be a span of the given tweet, the generative RC model will generate novel words or phrases that might not appear in the tweet. Both of them have achieved promising results in their respective ways. Is there an appropriate way to combine the advantages of them to further improve the performance for RC?

To justify this idea, we regard the extractive RC as an auxiliary task to help to optimize the generative RC model. In particular, we first unify the inputs of the extractive and the generative RC models into a single sequence "[CLS] *tweet* [SEP] *question* [SEP] *answer* [SEP]". Then, such inputs are fed into the lexicon encoder and the transformer to obtain the last hidden vectors $\boldsymbol{H}^L \in \mathbb{R}^{n \times d_h}$, where n is the length of the single sequence. Next, we distill the representations of the source segment (tweet and question) $\boldsymbol{H}^L_{Ex} \in \mathbb{R}^{m \times d_h}$ from $\boldsymbol{H}^L$ for the extractive RC model, where m denotes the length of the source segment. Finally, the contextual hidden state $\boldsymbol{H}^L_{Ge}$ of the generative RC model is consistent with $\boldsymbol{H}^L$. The loss of multi-task learning naturally turns into

$$L_{MLT} = L_{ex} + \lambda L_{ge}, \tag{8}$$

where L_{ex} and L_{ge} are calculated by Eq.(4) and Eq.(7), respectively.

3.6 Answer Selection

Up to now, we have two manners to get an answer by the extractive QA model (Subsection 3.4) and the generative QA model (Subsection 3.5), respectively. However, there is no one-size-fits-all method for a complex RC task, e.g., the RC task in social media, although we expect that the RC system can choose the most appropriate model to predict answers. Therefore, to further improve the scalability of our RC approach, we regard the problem as a binary classification task, to select the final answer from the outputs of the extractive and generative RC models.

To train the classifier, we first construct a training set by the following steps: 1) selecting the samples from the training set of TweetQA with different predicted answers by the extractive and generative RC models, and 2) setting the label with the higher answer score of the two to 1, and the other to 0. Then, we pack the predicted answer from the extractive RC model or the generative RC model with the corresponding question into a single sequence like "[CLS] *question* [SEP] *answer* [SEP]", and feed the sequence into the BERT (Devlin et al., 2019) to distill the last hidden vector of "[CLS]". Finally, we train the classifier using the "[CLS]" representations to determine whether the answer should be accepted (label 1) or not (label 0).

Item	#Train	#Dev	#Test
Ext	6,931	939	-
Gen	3,761	147	-
Total	10,692	1,086	1,979
Informal text	8,562	862	1,547

Table 3: Statistics of triples (tweet, question, answer) on the TweetQA corpus. Ext: the ground-truth answers are a span in tweets; Gen: the ground-truth answers do not match any exact substring in tweets.

Model	Hyper-parameter	Value	Model	Hyper-parameter	Value
Common	Dropout rate	0.1	Extractive RC model	Learning rate	3e-5
	Warm up	0.1		Epoch	2
	Batch size	12	Generative RC model	Learning rate	2e-5
	Max sequence length	128		Epoch	10
	λ in Eq.(8)	1		Mask probability	0.7
				Vocabulary size	30,522

Table 4: Hyper-parameter settings.

4 Experimentation

4.1 Settings

TweetQA is the first machine reading comprehension dataset on social media (Twitter) (Xiong et al., 2019). Two key properties distinguish it from the standard QA datasets such as the SQuAD dataset (Rajpurkar et al., 2016). First, in tweets, there are a large number of informal expressions, such as the examples in Table 2, which renders the existing RC models useless. As shown in Table 3, there are 8,562 (80%) samples need to be normalized. Second, unlike the extractive RC datasets in which the answers are exactly contained in the given passages, lots of answers in TweetQA are free-form texts by manual annotation. Table 3 shows that about 33% of answers do not match an exact fragment in the corresponding tweet.

Due to the free-form answers, the evaluation metrics for extractive RC tasks, such as exact match and F1, are not fit for our experimentation. Therefore, following the settings of the TweetQA leaderboard, we use three metrics to assess models, including BLEU-1 (Papineni et al., 2002), Meteor (Denkowski and Lavie, 2011), and Rouge-L (Lin, 2004). For data pre-processing, tweets and QA pairs are first tokenized by Stanford CoreNLP[1], which is consistent for all models. Then we utilize four text normalizers described in Subsection 3.1 to convert the noisy and informal tweets into their corresponding formal texts.

The implementations of the extractive RC model and the generative RC model are based on the PyTorch reimplementation BERT[2] (Devlin et al., 2019) and the UNILM[3] (Dong et al., 2019), respectively. All parameters of the systems are initialized by uncased large whole-word-masking BERT pre-trained model and optimized using Adam (Kingma and Ba, 2014). Table 4 shows the hyper-parameter settings in our experiments. As the TweetQA leaderboard makes the test set unseen, our experiments are conducted on the development set if not specified.

In this paper, we demonstrate the following systems for social media RC.

- **Ext-RC**. The extractive RC model described in Subsection 3.4.
- **Gen-RC**. The generative RC model described in Subsection 3.5.
- **NUT-RC**. The system integrates the extractive and generative RC models by the answer selection module.

[1] https://stanfordnlp.github.io/CoreNLP/index.html

[2] https://github.com/huggingface/transformers

[3] https://github.com/microsoft/unilm

Model	BLEU-1		Meteor		Rouge-L	
	Dev	Test	Dev	Test	Dev	Test
HUMAN	76.4	78.2	63.7	66.7	70.9	73.5
Ext-UB	79.5	80.3	68.8	69.8	74.3	75.6
Ext-UB$_{norm}$	87.1	-	84.6	-	86.7	-
BERT	67.3	61.4	56.9	58.6	62.6	64.1
Seq2Seq	53.4	36.1	32.1	31.8	39.5	39.0
Ext-RC	73.1	73.6	68.9	70.3	75.0	75.4
Ext-RC+	75.9	75.6	71.3	**72.1**	77.5	77.4
Gen-RC	77.3	75.9	72.1	71.7	78.6	77.7
NUT-RC	**78.2**	**76.1**	**73.3**	**72.1**	**79.6**	**77.9**

Table 5: Performances of RC models on TweetQA. The results on the test set are our first and only submissions to the TweetQA leaderboard.

Model	BLEU-1	Meteor	Rouge-L
UNILM	65.9	59.5	67.7
UNILM$_{BERT}$	74.9	69.9	76.4
Gen-RC	**77.3**	**72.1**	**78.6**

Table 6: Comparison of the generative RC models on the development set of TweetQA. UNILM: the pre-trained UNILM model (Dong et al., 2019) is directly fine-tuned on the TweetQA corpus without multi-task learning; UNILM$_{BERT}$: initializes its parameters with the BERT large uncased whole-word-masking model (Devlin et al., 2019).

As the top systems on the TweetQA leaderboard[4] do not release their papers or codes, we only compare the following baselines in Xiong et al. (2019).

- **BERT**. An extractive RC system with the uncased base BERT (Devlin et al., 2019).
- **Seq2Seq**. An RNN-based encoder-decoder RC system (Song et al., 2017), which encodes the passage and the question into a multi-perspective memory and decodes the answer with both copy and coverage mechanisms.

4.2 Experimental Results

Table 5 lists the performances of the baselines and our RC models. The performances of the HUMAN and the Ext-UB indicators (Rows 1 and 2) reported by Xiong et al. (2019) show the complexity and extractive upper bounds of TweetQA. Following with the Ext-UB indicator that directly extracts answer candidates in raw texts of tweets, we normalize tweets by the text normalizers (Subsection 3.1) and then extract answer candidates using the same method (Ext-UB$_{norm}$). It improves about 10% of the theoretical upper bounds of the three metrics on the development set, which demonstrates the effectiveness of our text normalization.

Rows 6-9 in Figure 5 show the performances of our RC models, where the Ext-RC+ model is first fine-tuned on the SQuAD dataset (Rajpurkar et al., 2016) based on the Ext-RC model. It shows that the Ext-RC+ model outperforms about 2% of absolute improvements than the Ext-RC model, which indicates that data augmentation from external resources can benefit the extractive RC model to some extent. Moreover, our NUT-RC model achieves the best performances on both the development set and the test set, even better than the human performances. Finally, the results also show that all of our RC models, including the extractive and the generative RC models, and the NUT-RC model, achieve promising results and exceed the baselines (BERT and Seq2Seq).

Table 6 shows the performances of the generative RC models with different pre-trained embeddings and settings. Comparing with the UNILM model (Row 1), the same model updated by BERT (Row 2) performs better, which is probably due to the whole word masking mechanism of BERT. In addition, the Gen-RC model (Row 3) achieves the best performance, which verifies the effectiveness of the multi-task learning of our generative RC model. The improvement might be benefited from the multi-task

[4] https://tweetqa.github.io/

Normalizer	BLEU-1		Meteor		Rouge-L	
	Ext-RC	Gen-RC	Ext-RC	Gen-RC	Ext-RC	Gen-RC
None	67.9	74.1	64.9	68.9	70.1	75.8
+SPLT	73.1	74.9	68.9	69.9	75.0	76.4
+SPLT+EXPN	73.5	75.7	69.6	70.5	75.5	77.1
+SPLT+WDLK	73.9	**75.9**	69.7	70.0	75.8	**77.2**
+SPLT+MISC	**74.1**	75.8	**70.5**	**70.9**	**76.0**	77.1

Table 7: Ablation study of text normalizers when removing one at a time.

learning affected by a regularization via alleviating over-fitting to the generative RC task, thus making the underlying representations transferable from the extractive RC task to the generative RC task.

Text normalization is a vital pre-processing step for the RC on social media texts. To better quantify the contributions of the different text normalizers, we conduct an ablation study. Noted that the SPLT normalizer is a prerequisite component for the other normalizers, thus we add it first and then add the other normalizers respectively. As shown in Table 7, we observe that the SPLT normalizer can lead to noticeable improvements, especially for the extractive RC model. The results also show that the performance improves slightly when adding EXPN, WDLK, and MISC respectively. Therefore, we take the SPLT and the MISC normalizers as our experimental settings.

4.3 Error Analysis

To better understand which types of errors are the most influential factors leading to the failure of the NUT-RC model, we manually analyzed 100 error cases of the NUT-RC model on the development set of TweetQA. The error types mainly can be divided into five aspects.

Synonymous expression (39%). Due to the diversity of language itself, different expressions in texts might correspond with the same meaning. For instance, given the question "*what kind of discussion is this about?*", while the ground truth answer is "*political*", the predicted answer is "*a politics discussion*". Obviously, the two answers are completely equivalent in semantics, but they have inferior match quality by the evaluation metrics (BLEU-1, Meteor, and Rouge-L). Therefore, how to evaluate the generated answers is imperative in future work.

Informal expression in social media (27%). Although we adopt four text normalizers (in Section 3.1) to normalize the informal tweets, there are still many errors concentrating on 1) UserID which often needs to be understood to answer person-related questions; 2) HashTag which is often used to indicate an event; and 3) some informal oral-English expressions, such as "*CHEAH!! USA!!*".

Extractive boundary (17%). All of these error cases come from the extractive RC model. Comparing with the ground-truth answer, the predicted answer is either a longer one (e.g., "*a live look*" vs. "*a live look in*") or a shorter one (e.g., "*he was hit by a pitch*" vs. "*by a pitch*").

Reasoning and inference (12%). Answering this type of error cases requires the inference ability on multiple sentences and commonsense reasoning.

Annotated error (5%). A small part of ground-truth answers are annotated incompletely or incorrectly.

5 Conclusion

In this paper, we come up with a noisy user-generated text-oriented RC model on social media texts. Through text normalizers, the model transforms informal texts into well-formed texts. Moreover, we train the generative RC model with the help of the extractive model by a multi-task learning mechanism. Finally, we apply an answer selection module to choose the more appropriate answer from the generative and the extractive RC models. Empirical results show that the proposed model significantly outperforms the state-of-the-art baselines on the TweetQA dataset. In future work, we plan to justify the proposed model on other free-form QA datasets. The source code of our NUT-RC model is publicly available at `https://github.com/WhaleFallzz/NUT_RC`.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 61703293, No. 61751206, and No. 61672368) and the National Key Research and Development Program of China(2017YFB1002104). This work was also supported by the joint research project of Alibaba and Soochow University. Finally, we would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. *arXiv*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium, October-November. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July. Association for Computational Linguistics.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv*.

Emma Flint, Elliot Ford, Olivia Thomas, Andrew Caines, and Paula Buttery. 2017. A text normalisation system for non-standard English words. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 107–115, Copenhagen, Denmark, September. Association for Computational Linguistics.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef Van Genabith. 2011. # hardtoparse: Pos tagging and parsing the twitterverse. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ ACL WNUT NER shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China, July. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July. Association for Computational Linguistics.

Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2018. Joint multitask learning for community question answering using task-specific embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4196–4207, Brussels, Belgium, October-November. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv*.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012, Doha, Qatar, October. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September. Association for Computational Linguistics.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745, Melbourne, Australia, July. Association for Computational Linguistics.

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

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. 2018. Parsing tweets into universal dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 965–975, New Orleans, Louisiana, June. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy, July. Association for Computational Linguistics.

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*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, March.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv*.

Linfeng Song, Zhiguo Wang, and Wael Hamza. 2017. A unified query-based generative model for question generation and question answering. *arXiv*.

Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. S-net: From answer extraction to answer synthesis for machine reading comprehension. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada, August. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, Vancouver, Canada, July. Association for Computational Linguistics.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714, Melbourne, Australia, July. Association for Computational Linguistics.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. NLProlog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6151–6161, Florence, Italy, July. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy, July. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv*.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *arXiv*.