Topic Knowledge Acquisition and Utilization for Machine Reading Comprehension in Social Media Domain

Zhixing Tian[1,2], **Yuanzhe Zhang**[1], **Kang Liu**[1,2], **Jun Zhao**[1,2],
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{zhixing.tian, yzzhang, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

In this paper, we focus on machine reading comprehension in social media. In this domain, one normally posts a message on the assumption that the readers have specific background knowledge. Therefore, those messages are usually short and lacking in background information, which is different from the text in the other domain. Thus, it is difficult for a machine to understand the messages comprehensively. Fortunately, a key nature of social media is clustering. A group of people tend to express their opinion or report news around one topic. Having realized this, we propose a novel method that utilizes the topic knowledge implied by the clustered messages to aid in the comprehension of those short messages. The experiments on TweetQA datasets demonstrate the effectiveness of our method.

1 Introduction

As an increasing number of people share and obtain information from social media, social media is now becoming an important real-time information source. The unprecedented volume, variety of user-generated content, and the user interaction network constitute new opportunities for understanding social behavior and building socially intelligent systems. It is important and challenging to teach a machine to automatically understand the content presented in social media.

Although considerable progress has been made in the task of Machine Reading Comprehension (MRC), most of the previous works only focus on the comprehension of the other domains, such as news (Hermann et al., 2015; Chen et al., 2016), story (Sachan et al., 2015; Narasimhan and Barzilay, 2015) and Wikipedia (Seo et al., 2016; Yang et al., 2018), and there are very few works addressing the problem of social media MRC. Table 1 shows an example of social media comprehension. Different from the other domains, in social media domain, one normally posts a message on the assumption that the readers have specific background knowledge. Those messages are generally short and contain limited contextual information as shown in the table. Thus, it is difficult for a machine to understand them thoroughly based only on the text itself. In this example, if only look at the message, without background knowledge, a machine reader would be puzzled about its topic and could not answer the question.

Message: *in case you missed it, here's your first look at jamie bell's the thing:* ***# fantasticfour*** *empire magazine (@ empiremagazine) april 9, 2015*
Question: *what movie is being shared here?* **Answer:** *fantastic four*

Table 1: An example of machine reading comprehension in social media domain.

To obtain background knowledge for a machine reader, one feasible method is to introduce external knowledge from the knowledge base like ConceptNet (Liu and Singh, 2004) and WordNet (Fellbaum, 1998), as the previous works do in other domains (Lin et al., 2017; Wang et al., 2018; Wang and Jiang, 2019a). Unfortunately, due to the nature of informality and diversity of the messages in social media, some key phrases of the short messages cannot be found in those pre-constructed knowledge base. For example, in Table 1, the token *fantasticfour*, which indicates the topic of the message, cannot be found in ConceptNet or WordNet.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, August 13 - 15, 2021.

By studying social media messages, we find that a significant nature of them is clustering. That is to say, on a social media platform, a group of people tend to express their opinion or report news around one topic. Specifically, those topic-relevant messages is commonly clustered by the hashtag, which is marked with "#" symbol (e.g., #fantasticfour in in Table 1) and ubiquitous in social media domain. Thus, given a social media message, we can find a group of relevant messages based on the hashtag. As shown in Table 2, there are a series of topic-relevant messages clustered by the hashtag "#fantasticfour". Through those messages, we would know the topic is a science fiction film, from Marvel, about some superheroes and so on. Those hashtag-clustered messages tend to share a common topic and can be considered as a knowledge source of the topic of the given message. To this end, we propose a novel method, which obtains and utilizes the topic knowledge from the hashtag-clustered messages, to address the problem of lack of background knowledge in the task of social media comprehension.

Message0: ***#FantasticFour*** *was way better than I thought it would be! It's def a science fiction film tho this ain't for the mindless action crowd.*
Message1: *GREAT SCIENCE FICTION FILM!* ***#fantasticFour***
Message2: *Move over, Captain Marvel!* ***#FantasticFour***
Message3: *#2020WillBeTheYearFor the* ***#FantasticFour*** *to unite with all of these cool characters in @Marvel Ultimate Alliance 3!*
Message4: ***#FantasticFour*** *Remains the Worst Superhero Film of the Decade*
Message5: *@MarkHarrisNYC delivers a postmortem of* ***#FantasticFour*** *and the state of superhero films*

Table 2: An example of hashtag-clustered messages in social media.

Given a message and a question, we extract the hashtag from the message and retrieve the relevant messages based on the hashtag. Subsequently, we refine topic knowledge from the retrieved messages. Moreover, we construct a neural network, dubbed as Topic Knowledge Reader (TKR). The refined knowledge will be fused into the TKR model and contribute to the process of reading comprehension and question answering. We conduct experiments on the TweetQA dataset (Xiong et al., 2019). The result shows the effectiveness of our method.

To summarize, the major contributions of this paper are as follows:

- In the task of machine reading comprehension, we investigate the problem of lack of background knowledge in social media domain. We propose to utilize the nature of clustering of social media to obtain the knowledge from the other relevant messages.
- We propose a particular knowledge acquisition approach, which retrieves and refines topic knowledge from those relevant messages clustered by the hashtag which exists generally in the social media messages.
- We build a machine reading comprehension model, TKR, to utilize the refined knowledge in a targeted manner and conduct experiments on the public dataset, which demonstrates the effectiveness of our method.

2 Related Work

Social Media NLP: Over the past few years, social media has revolutionized the way we communicate. Massive amount of information in form of text is continuously generated by the users, which creates enormous challenges for NLP community to analyze and understand those text automatically. In recent years, several NLP techniques and datasets for processing social media text have been proposed. Dos Santos and Gatti (2014) use a deep convolutional neural network that exploits from character-level to sentence-level information to perform sentiment analysis of short texts. Vo and Zhang (2015) splits

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, August 13 - 15, 2021.

the context and employs distributed word representations and neural pooling functions to extract features from tweets. Zhou and Chen (2014) propose a graphical model, named location-time constrained topic (LTT), to capture the content, time, and location of social messages. Singh et al. (2017) develop an event classification and location prediction system which uses the Markov model for location inference. Qian et al. (2019) jointly discover subevents from microblogs of multiple media types—user, text, and image, and design a multimedia event summarization process.

Machine Reading Comprehension: Due to the fast development of deep learning techniques and large-scale datasets, Machine Reading Comprehension(MRC) has gained increasingly wide attention over the past few years. Richardson et al. (2013) build the multiple choice dataset MCTest, and this dataset encourages the early research of machine reading comprehension, and a strand of MRC models (Sachan et al., 2015; Narasimhan and Barzilay, 2015) are inspired by the dataset. Hermann et al. (2015) propose a cloze test dataset CNN & Daily Mail, which is large-scale and more suitable than MCTest for deep learning methods. Based on this dataset, Hermann et al. (2015) propose an attention-based LSTM (Hochreiter and Schmidhuber, 1997) model named Attentive Reader. Moreover, Rajpurkar et al. (2016) release the span extraction dataset, SQuAD, which has become the most popular MRC dataset over recent years. This dataset enlightens a lot of classical MRC model, like BiDAF (Seo et al., 2016) and R-Net (Wang et al., 2017). In addition, the multi-hop MRC dataset HotpotQA (Yang et al., 2018) has gained recent wide attention. This dataset addresses the problem of multiple clues based question answering.

3 Method

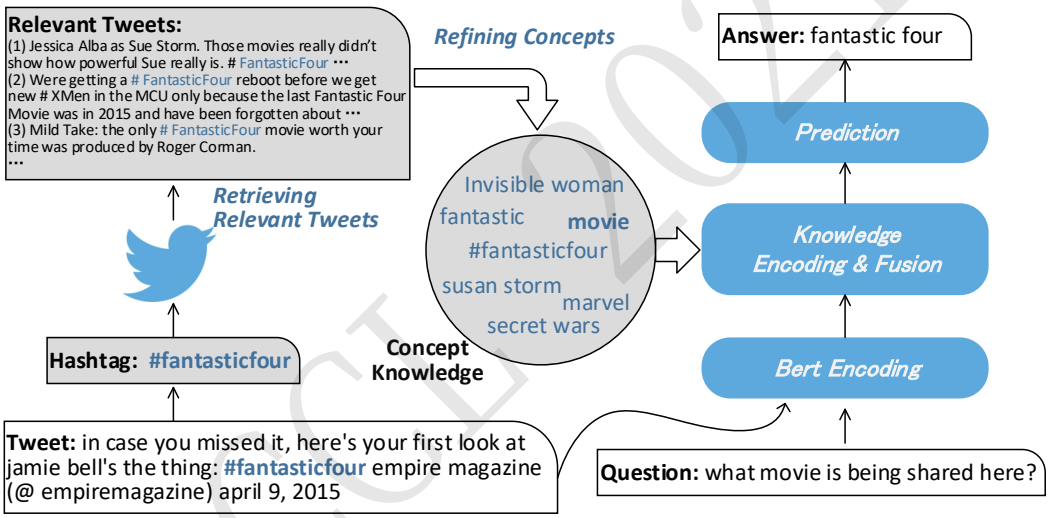

Figure 1: The framework of our proposed method. The left half is the process of obtaining topic knowledge. The right half is the reading comprehension model, Topic Knowledge Reader (TKR)

Figure 1 shows the framework of our method. Note that it is an example of the tweets, but the method is universal for the messages from other social media platforms. Given a tweet and a question, we obtain the answer by the following steps: First, we extract the hashtag from the tweet, meanwhile, we encode the tweet and the question by the BERT encoder. Second, we retrieve relevant tweets that contain the same hashtag. Third, we refine the topic knowledge from the retrieved tweets. Next, the knowledge is encoded and fused with the BERT representation. Finally, the model predicts the answer based on the knowledge aware representation of the tweet and question.

3.1 Knowledge Acquisition

We regard the set of tweets clustered by the hashtag as the resource of knowledge and obtain topic knowledge from them. We first retrieve relevant tweets, then from those tweets, we gather common concepts. Meanwhile, we maintain a hashtag pool to score each concept. Finally, we refine the concepts by select the top-k scored ones as the topic knowledge.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988–999, Hohhot, China, August 13 – 15, 2021.

3.1.1 Retrieving Relevant Tweets

Given a tweet text T, we first extract the hashtag H. Next, we use H as the query and retrieve the relevant tweets. So we have a set S consisting of tweets that contain the same hashtag. The tweets in S tend to share the same topic with the given tweet T, and the information of them is helpful to understand T comprehensively. Next, we remove the non-English tweets from S, and then delete the non-normal strings in the text of S such as the URL which starts with "$http$" and the reference of the picture which starts with "$pic\backslash$".

Exceptionally, for those tweets that contain no hashtag, we utilize a hashtag extractor to extract hashtag words from the tweet. The extractor is composed of a BERT encoder and a span pointer. We input the tweet, T, to the BERT model and obtain the representation $P = \{p_0, p_1, ..., p_n\}$, where p_i is the i-th word of the tweet. Then we extract the hashtag from the tweet by a pointer:

$$Start_i = \frac{exp(w_0^T p_i)}{\sum_j exp(w_0^T p_j)} \qquad End_i = \frac{exp(w_1^T p_i)}{\sum_j exp(w_1^T p_j)} \tag{1}$$

Where w_0 and w_1 are trainable vectors. The pointer labels the probability of each word as the start and the end of the hashtag, respectively. We calculate the score of a span by multiplying those two probabilities and take the span with the max score as the hashtag. Figure 2 shows an example. We train the extractor model on a hashtag extraction dataset proposed by Zhang et al. (2016). Evaluated on the test set of the dataset, our extractor achieves 85.1% accuracy.

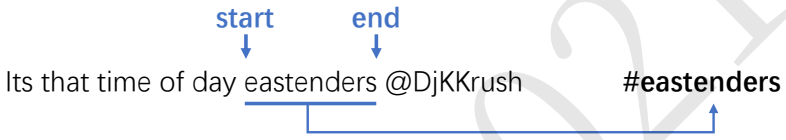

Figure 2: An example of extracting hashtag for those tweets without hashtag

3.1.2 Gathering Relevant Concepts

Having retrieved the tweets with the same topic, we gather the fine-grained knowledge, i.e., concepts that connect to the topic. We tokenize every tweet text in S and obtain a set of tokens. Then we segment each token to get the concept. Due to the nature of informality, some tokens from the tweets could contain multiple words, like the hashtag "#secretwars", thus we conduct a segmentation on each token. After that, we obtain a set C consisting of concepts (e.g., "movie", "marvel", and "secret wars").

3.1.3 Maintaining Hashtag Pool

To further refine the concepts, we maintain a hashtag pool. First of all, a large scale of recent tweets is collected as the original corpus. Based on the corpus, we collect the hashtags, then find the relevant tweets and obtain the concept set C for each hashtag follow above-mentioned process. Those hashtags and their all relevant concepts are added to the empty hashtag pool as the initialization. When a new tweet, T, is given during application, we update the hashtag pool by adding the hashtag of T and the relevant concepts, C, to the pool.

3.1.4 Refining Topic Knowledge

Given tweet T, by above-mentioned steps, we have concepts C, then we apply Term Frequency-Inverse Document Frequency (TF-IDF) to score each concept. The score for $concept_i$ in C is calculated by:

$$score_i = \frac{n_i}{N} log \frac{|P|}{|p_i| + 1} \tag{2}$$

where n_i is the frequency of $concept_i$ in C, N is the total count of concepts in C, P denotes the hashtag pool. Thus, $|P|$ is the total number of the hashtags in the hashtag pool, and $|p_i|$ is the number of hashtags, whose relevant concepts contain the $concept_i$, in the hashtag pool. Finally, the top-k scored concepts are selected as the topic knowledge K of the tweet T.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, August 13 - 15, 2021.

3.2 Topic Knowledge Reader

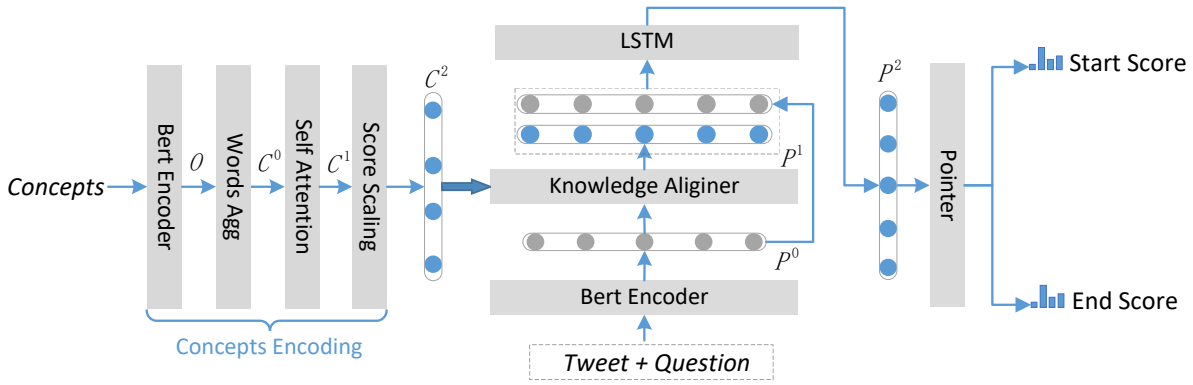

Figure 3: The detail architecture of Topic Knowledge Reader (TKR).

As shown in Figure 3, we propose a reading comprehension model, named Topic Knowledge Reader (TKR), to fuse the refined concepts and then answer the question. The inputs of the model are

- the given tweet $T = \{t_0, t_1, ..., t_{n-1}\} \in \mathbb{R}^n$, where n is the number of words in the tweet, t_i is the *i-th* word in T.
- the question $Q = \{q_0, q_1, ..., q_{m-1}\} \in \mathbb{R}^m$, where m is the number of words in the question, q_i is the *i-th* word in Q.
- the concept knowledge $K = \{k_{00}, k_{01}, ..., k_{ij}, ..., k_{(l-1)x}\} \in \mathbb{R}^y$, where y is the number of words of all concepts, k_{ij} refers to the j-th word of i-th concept.
- the concept score $S = \{s_0, s_1, ..., s_{l-1}\} \in \mathbb{R}^l$.

The output of the model is the predicted answer.

3.2.1 Encoding Tweet and Question

We first concatenate the question Q and the tweet T. The combination passage is

$$D = \{[CLS], t_0, t_1, ..., t_{n-1}, [SEP], q_0, q_1, ...q_{m-1}, [SEP]\} \tag{3}$$

where we add the special word "[CLS]" and "[SEP]", which follows the process in Devlin et al. (2018). Then, we employ BERT (Devlin et al., 2018) to encode the tweet and the question together, thus we have the question-aware representation of the passage:

$$P^0 = BERT\,(D) \in \mathbb{R}^{(m+n+3)\times h} \tag{4}$$

3.2.2 Encoding Concepts

We encode the concepts, before the step of fusion. To obtain the original representation, we apply BERT encoder for the concepts as well. Analogously, we add the special word "[CLS]" to the single sequence of knowledge words,

$$K = \{[CLS], k_{00}, k_{01}, ..., k_{ij}, ..., k_{(l-1)x}\} \tag{5}$$

and then the pre-trained model, BERT, is applied on encoding the concepts:

$$O = BERT\,(K) \in \mathbb{R}^{(y+1)\times h} \tag{6}$$

Words Aggregation: As there are multiple words in some concepts, we aggregate the words of each concept by mean pooling, then obtain the one-vector representation, c_i^0, for each concept:

$$c_i^0 = \frac{1}{N} \sum_{j \in [0,N)} c_{ij}^0 \qquad C^0 = \left\{c_0^0, c_1^0, ..., c_i^0, ..., c_{l-1}^0\right\} \in \mathbb{R}^{l \times h} \tag{7}$$

Self Attention: Though no sequential relation exists among those concepts, they are still interrelated. Thus, we use the self-attention mechanism to perform a non-sequence context encoding on the concepts:

$$c_i^1 = \sum_j \alpha_{ij} c_j^0 \qquad \alpha_{ij} = \frac{exp(\sigma(W_q c_i^0) \cdot \sigma(W_k c_j^0))}{\sum_{j'} exp(\sigma(W_q c_i^0) \cdot \sigma(W_k c_{j'}^0))} \tag{8}$$

where σ is the activation function, $W_q \in \mathbb{R}^{h \times h}$ and $W_k \in \mathbb{R}^{h \times h}$ are trainable matrixes. Thus we have self-aligned concepts $C^1 = \left\{c_0^1, c_1^1, ..., c_{l-1}^1\right\} \in \mathbb{R}^{l \times h}$

Score Scaling: We then scale the concepts by the score $S \in \mathbb{R}^l$ assigned in the step of knowledge refining:

$$C^2 = SC^1 \in \mathbb{R}^{l \times h} \tag{9}$$

C^2 denotes the final representation of the Concepts.

3.2.3 Topic Knowledge Fusion

The Concepts are fused into the passage by:

$$p_i^1 = \sum_j \beta_{ij} c_j^2 \qquad \beta_{ij} = \frac{exp(\sigma(W_p p_i^0) \cdot \sigma(W_c c_j^2))}{\sum_{j'} exp(\sigma(W_p p_i^0) \cdot \sigma(W_c c_{j'}^2))} \tag{10}$$

where σ is the activation function, $W_p \in \mathbb{R}^{h \times h}$ and $W_c \in \mathbb{R}^{h \times h}$ are trainable matrixes. Thus, we obtain the concepts-aware passage representation $P^1 = \left\{p_0^1, p_1^1, ..., p_{m+n+2}^1\right\} \in \mathbb{R}^{(m+n+3) \times h}$. A bidirectional LSTM is applied to conduct an additional sequential context encoding and aggregate the original question-aware passage representation P^0 and the concepts-aware passage representation P^1.

$$P^2 = BiLSTM([P^0; P^1]) \in \mathbb{R}^{(m+n+3) \times h} \tag{11}$$

3.2.4 Prediction

We employ two Linear layers to point the start position and the end position of the answer in the passage, respectively, and then normalize the prediction scores:

$$\tilde{Start}_i = \frac{exp(w_s^T p_i^2)}{\sum_j exp(w_s^T p_j^2)} \qquad \tilde{End}_i = \frac{exp(w_e^T p_i^2)}{\sum_j exp(w_e^T p_j^2)} \tag{12}$$

$w_s \in \mathbb{R}^h$ and $w_e \in \mathbb{R}^h$ are trainable weight vectors. We utilize Negative Log Likelihood (NLL) as the loss function during training. Moreover, during the evaluation, we obtain the score of each span of the tweet by multiplying its start score and the end score and then select the text span with the max score as the answer.

4 Experiment

4.1 TweetQA Dataset

We conduct experiments on the recently released social media MRC dataset, TweetQA. Each instance of the dataset is a triple consisting of a tweet text, a human proposed question, and a list of human-annotated answers. The dataset is composed of 10692 training triples, 1086 development triples, and 1979 test triples. It is the first large-scale MRC dataset over social media data.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, August 13 - 15, 2021.

4.2 Implement Detail

Preprocess: As we employ BERT (Devlin et al., 2018) to encode the text, we tokenize the text by the default tokenizer of BERT. Since the answer spans are not labeled in the train set, we annotate the approximate answer span in each tweet by selecting the span that achieves the best F1 score.

Knowledge acquisition: To simulate the real-world scenario where a social media MRC system works, we regard the train set as the original corpus for the initialization of the Hashtag Pool. During evaluating, we update the Hashtag Pool by the hashtag and the relevant concepts, from the development set and the test set. Based on the experimental analysis, we select top-8 scored concepts for each hashtag at the step of refining the knowledge.

Training: We select the instances that contain the span whose F1 score no less than 0.6 to train the model in the way of weakly supervised. As a result, 8238 instances are used during training. We employ Adam optimizer to train the model. The learning rate is set to 3×10^{-5}, the model is fine-tuned for 3 epochs, and the dropout rate of BERT is set to 0.1. The BERT model we choose is the pre-trained *bert-base* (Devlin et al., 2018) model, distinguished from the *bert-large* model. The hidden size of BERT is 768.

Evaluation: As the answer in TweetQA is not always a span of given tweet, following Xiong et al. (2019), we use the metrics for natural language generation to evaluate the models, namely BLEU-1, Meteor, and Rouge-L . The answers of the test set are not released, so we submit our prediction to the official evaluating platform of TweetQA[1] and receive the response of the performance results.

4.3 Baselines

- **Query Matching**: a simple IR baseline (Kočiskỳ et al., 2018), which is adapted to the TweetQA Task by Xiong et al. (2019).

- **BiDAF**: a popular neural baseline (Seo et al., 2016) of Machine Reading Comprehension, which extract answers from the original tweet text.

- **Gerative QA**: a RNN-based generative model (Song et al., 2017). The model employs both copy and coverage mechanisms during the process of generating.

- **BERT Extraction**: a recently proposed pre-trained model (Devlin et al., 2018). Following (Devlin et al., 2018), we construct a BERT based answer extraction model by inputting the representation of passage, P^0, obtained from Equation 4 directly to the prediction layer formulated by Equation 12.

- **BERT Generation**: Because part of the answers of TweetQA are not a span of the tweet text, we build a BERT based generative model. We use BERT as the encoder same with Equation 4. Following (Tay et al., 2019), we employ a pointer generator, which selects words from both the tweet and the vocabulary, to decode the answer. The generative model is trained on all instances of the train set.

- **Knowledge Concat**: We also introduce another simpler method to fuse the topic knowledge, named "Knowledge Concat". The model directly concatenates topic knowledge (i.e., the selected concepts) with the sequence of tweet and question before BERT encoding and finally, same with TKR, conduct a span prediction.

- **KAR**: Knowledge Aided Reader (KAR) (Wang and Jiang, 2019b) is a recently proposed MRC model, which utilizes the knowledge from WordNet. The model conducts mutual attention and self-attention based on the connections among the words of the question and the passage. The connections are built based on the knowledge from WordNet. For a fair comparison, we change the model by utilizing BERT as the basic encoder instead of the original embedding layers composed of Glove (Pennington et al., 2014), CNN (Krizhevsky et al., 2012), and LSTM.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, August 13 - 15, 2021.

Model	Dev Set			Test Set		
	BLEU-1	Meteor	Rouge-L	BLEU-1	Meteor	Rouge-L
Human	-	63.7	70.9	70.0	66.7	73.5
Extract-UB	-	68.8	74.3	75.1	69.8	75.6
Query Matching	-	12.0	17.0	11.2	12.1	17.4
BiDAF	-	31.6	38.9	34.9	31.4	38.6
Gerative QA	-	32.1	39.5	36.1	31.8	39.0
BERT Generation	48.5	42.0	51.8	49.1	42.1	52.3
BERT Extraction	61.0	58.4	64.2	63.2	60.9	65.8
Knowledge Concat	65.5	61.6	67.9	66.9	63.4	69.1
KAR	66.9	63.2	68.9	67.7	64.1	69.8
TKR	**68.7**	**64.7**	**70.6**	**69.0**	**65.6**	**71.2**

Table 3: The results on TweetQA dataset. Extract-UB denotes the upper bound of extractive methods.

4.4 Main Results

As shown in Table 3, our model, TKR, surpasses the recently proposed Knowledge Aided Reader (KAR) and achieves competitive performance. From our point of view, due to the limitation of the knowledge from the pre-constructed knowledge base (WordNet), KAR suffers from the sparsity problem of knowledge extraction for the diverse and informal expressions in social media domain. Besides, TKR outperforms all of the other baselines significantly, especially the BERT based model, BERT Extraction. The model, BERT Extraction, is exactly the rest architecture of TKR when we ablate topic knowledge from TKR. Thus, the comparison between TKR and BERT Extraction can directly demonstrate the advantages of our methods to acquire and utilize topic knowledge for social media comprehension.

Moreover, Knowledge Concat performs better than BERT Extraction, which also validates the effectiveness of the knowledge. Besides, comparing Knowledge Concat with TKR, we find that TKR performs better. This is because TKR can integrate our refined knowledge to the MRC model in a more targeted manner.

4.5 Different Number of Concepts

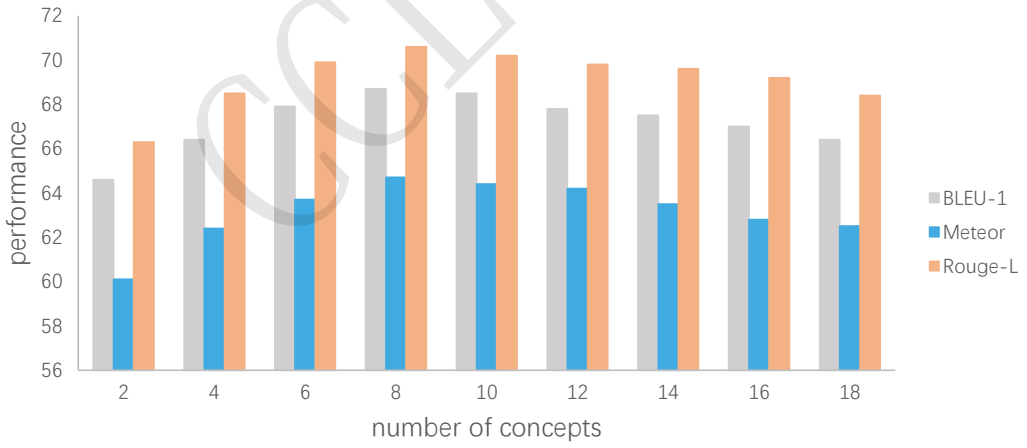

Figure 4: The performance of TKR with different number (k) of concepts on the dev set of TweetQA.

To further verify the effectiveness of the topic knowledge, we study the relationship between the number of employed concepts and the performance of TKR. We choose top-k concepts during the step of refining, where k changes from 2 to 18, and then train and evaluate TKR at different settings of k. As shown in Figure 4, by increasing the number, k, from 2 to 18, the performance of TKR first rises rapidly

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

until k reaches 8 and then drop down slowly. The gain of performance from $k = 2$ to $k = 8$ proves our topic knowledge effective. The loss of performance from $k = 8$ to $k = 18$ is caused by the noise introduced by the concepts with low scores.

Tweet: *Vets saw a fetus on Mei Xiang's ultrasound today. Paws crossed 4 viable pregnancy #PandaStory National Zoo (@NationalZoo)*
Hashtag: *PandaStory*
Concepts: *panda life, national zoo, panda, awesome ip, Mei Xiang, ..., youtube, conservation, giant, yuan meng panda, panda fans on instagram*
Question0: *where is mei xiang located?* **Answer0:** *at the national zoo*
Question1: *what is the name of the panda?* **Answer1:** *mei xiang*

Table 4: Sampled cases which show the effect of the different numbers of concepts. The concepts in blue are the *Top-5* scored ones, and those in black are the *13-18th* scored concepts.

To probe the effect of employing different numbers of concepts more intuitively, we sample and analyze some cases from the development set. Table 4 shows one of them, where *Question0* and *Question1* are two questions proposed based on the same tweet as shown in the table. In this example, we find that the *top-5* concepts describe the topic comprehensively, build a semantic connection between some key concepts in the tweet including *panda*, *Mei Xiang*, and *Nattional Zoo*, and finally contribute to answering the questions. On the contrary, the *13-18th* scored concepts tend to deviate from the topic, and as noisy-like information, they even damage the Reading Comprehension model, TKR, when introduced into the model.

4.6 Ablation Study

Model	BLEU-1	ΔBLEU-1	Meteor	ΔMeteor	Rouge-L	ΔRouge-L
TKR	68.7	-	64.7	-	70.6	-
- score scale	68.0	-0.7	64.3	-0.4	70.0	-0.6
- self attn	67.9	-0.8	64.0	-0.7	69.7	-0.9
- word agg	68.1	-0.6	64.1	-0.6	70.1	-0.5
- LSTM	68.4	-0.3	64.5	-0.2	70.3	-0.3

Table 5: Ablation study on the development set. *-score scale* denotes TKR without the module of score scaling. *-self attn* denotes TKR without self attention. *-word agg* denotes TKR without word aggregation. *- LSTM* denotes TKR that use a dense layer instead of LSTM for the knowledge fusion

To study the effect of some key modules of TKR, we conduct ablation experiments on the development set. As shown in Table 5, all of the three knowledge encoding module, including *score scale*, *self attention* and *word agg*, contribute to the overall performance. The results demonstrate that those modules. which are designed for the topic knowledge in a targeted manner, indeed help the model to encode the knowledge and further to absorb it. Furthermore, the performance of - *LSTM* is slightly behind the original TKR, which proves that the sequential information captured by the additional context encoding is beneficial for the comprehension.

4.7 Extractive vs. Generative

As shown in Table 3, compared with BERT Generation, the extractive models including BERT Extraction and TKR achieve better performance, though there is no identically matching substring in the tweet for part of answers. Table 6 shows two sampled cases which tell the difference between the extractive model and the generative one. As shown in the table, the generative model performs better in some cases where the answer is supposed to be synthesized based on the question and tweet. On the contrary, the

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, August 13 - 15, 2021.

Tweet: *thank you to @ marvel for sending gifts for our patients. they were thrilled to receive them! ...* **Question:** ru*what were patients of Seattle children's thrilled to receive?* **Answer:** *gifts from marvel* **Generative prediction:** *gifts of marvel* **Extractive prediction:** *gifts*
Tweet: *our prayers are with the students, educators & families at independence high school & all the first responders on the scene. # patriotpride ...* **Question:** *at which school were first responders on the scene for?* **Answer:** *independence high school* **Generative prediction:** *the school* **Extractive prediction:** *independence high school*

Table 6: Sampled cases that show the difference between the generative model and the extractive one.

generative model lag behind the extractive one, when the answer is an uninterrupted snippet of the tweet. However, as studying more cases, we find that even in many cases, where the answer need to synthesize, the generative model fails to provide a qualified answer. We consider that much more data is needed to train a qualified generative MRC model.

4.8 Weakly Supervised Training

To train the extractive model, TKR, we annotate the answer span in the tweets by the F1 score. We train the model to locate the annotated span. As the annotated span may not be the true answer, it is a process of weakly supervised training. As shown in Table 7, we study the relationship between the span score of training data and the performance which is evaluated on the development set. As shown in the table, by reducing the threshold of span score, increasing training data is involved, meanwhile the performance first rises until *span score = 0.6* and then drop down. That is to say, in the process of introducing different amount of the weakly supervised training data, *span score = 0.6* is the point where the difference between the benefit from the positive example and the damage from the noise is maximized.

Span Score	data	proportion	BLEU-1	Meteor	Rouge-L
≥ 1	6899	64%	67.0	63.5	69.1
≥ 0.8	7442	69%	67.9	64.3	69.2
≥ 0.6	8238	77%	**68.7**	**64.7**	**70.6**
≥ 0.4	8978	83%	68.2	64.2	70.3
≥ 0.2	9283	86%	67.3	63.3	69.3
> 0	9314	87%	67.1	63.0	69.2

Table 7: The performance on development set with different scale of train data. $SpanScore \geq i$ denotes that the model is trained by the instances containing the span whose F1 score is no less than i. *data* and *proportion* refer to the scale and the proportion of the selected training data.

5 Conclusion

In this paper, we focus on machine reading comprehension in social media domain. We propose a novel method to address the problem of lacking in background knowledge in this task. Utilizing the nature of clustering of social media, we retrieve and refine topic knowledge from the relevant messages, and then integrate the knowledge into an MRC model, TKR. Experimental results show that our proposed method outperforms the recently proposed models and the BERT-based baselines, which proves the method effective overall. By introducing different amount of topic knowledge, we demonstrate the effectiveness of our refined knowledge. Moreover, the ablation study further validates the contribution of the key modules of TKR for utilizing the knowledge.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, August 13 - 15, 2021.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61976211, No.61922085, No.61906196). This work is also supported by the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006), the Open Project of Beijing Key Laboratory of Mental Disroders (2019JSJB06).

References

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany, August. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Cicero Dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with heterogeneous knowledge for commonsense machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2032–2043, Copenhagen, Denmark, September.

H Liu and P Singh. 2004. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, October.

Karthik Narasimhan and Regina Barzilay. 2015. Machine comprehension with discourse relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1253–1262.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Xueming Qian, Mingdi Li, Yayun Ren, and Shuhui Jiang. 2019. Social media based event summarization by user–text–image co-clustering. *Knowledge-Based Systems*, 164:107–121.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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.

Mrinmaya Sachan, Kumar Dubey, Eric Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 239–249.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Jyoti Prakash Singh, Yogesh K Dwivedi, Nripendra P Rana, Abhinav Kumar, and Kawaljeet Kaur Kapoor. 2017. Event classification and location prediction from tweets during disasters. *Annals of Operations Research*, pages 1–21.

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

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy, July.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Chao Wang and Hui Jiang. 2019a. Explicit utilization of general knowledge in machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2263–2272, Florence, Italy, July.

Chao Wang and Hui Jiang. 2019b. Explicit utilization of general knowledge in machine reading comprehension. In *ACL*.

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.

Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. *arXiv preprint arXiv:1803.00191*.

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.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845, Austin, Texas, November. Association for Computational Linguistics.

Xiangmin Zhou and Lei Chen. 2014. Event detection over twitter social media streams. *The VLDB Journal—The International Journal on Very Large Data Bases*, 23(3):381–400.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 988-999, Hohhot, China, Augest 13 - 15, 2021.