Entity-Aware Social Media Reading Comprehension

Hao Liu[1], Yu Hong[1(✉)], and Qiao-ming Zhu[1,2]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
tianxianer@gmail.com, qmzhu@suda.edu.cn

[2] BigData Intelligence Engineering Lab of Jiangsu Province, Soochow University, Suzhou, China

Abstract. Social media reading comprehension (SMRC) aims to answer specific questions conditioned on short social media messages, such as tweets. Sophisticated neural networks and pretrained language models have been successfully leveraged in SMRC, accompanying with a series of deliberately-designed data cleaning strategies. However, the existing SMRC techniques still suffer from unawareness of various entity mentions, i.e., the successive tokens (words, sub-words or characters) that fully or briefly describe named entities, such as abbreviated person names. This unavoidably brings negative effects into question answering towards the questions of "who", "where", "which organization", etc. To address the issue, we propose to enhance the capacity of a SMRC model in recognizing entity mentions and, more importantly, construct an entity-aware encoder to incorporate latent information of entities into the understanding of questions and tweets. In order to obtain a self-contained entity-aware encoder, we build a two-channel encoder-shareable neural network for multitask learning. The encoder is driven to produce distributed representations that not only facilitate decoding of entity mentions but prediction of answers. In our experiments, we employ 12-layer transformer encoders for multi-task learning. Experiments on the benchmark dataset TweetQA show that our method achieves significant improvements. It is also proven that our method outperforms the state-of-the-art model NUT-RC, yielding improvements of 2.5% BLEU-1, 3% Meteor and 2.2% Rouge-L, respectively.

Keywords: Social media reading comprehension · Named entity recognition · Multi-task learning

1 Introduction

Machine Reading Comprehension (MRC) is a task of question answering conditioned on the semantic understanding of question and paragraph-level context. A variety of MRC datasets have been constructed to support related research in this field, including SQuAD [1], CoQA [2], NarrativeQA [3]), etc.

S. Khanna et al. (Eds.): PRICAI 2022, LNCS 13630, pp. 197–210, 2022.
https://doi.org/10.1007/978-3-031-20865-2_15

Table 1. An example of unrecognized named entities in social media domain.

Tweet: *This forecast is deflated as much as New England Patriots footballs! I apologize.* ***W NJ*** has the most to lose. Dave Curren(@DaveCurren)January 27,2015
Question: *Who has the most to lose?*
Gold Answer: *W NJ*
Predict Answer: *New England Patriots*

Recently, TweetQA[1] [4] is released for the evaluation of MRC techniques, which limits the available contexts to tweets. It raises an intensive interest in exploring effective MRC solutions towards short and informal texts. The task defined on this dataset is referred to Social Media Reading Comprehension (abbr., SMRC). Table 1 illustrates an example, where a specific SMRC model is required to predict the answer "*W NJ*" given the question "*Who has the most to lose*?". The clues that support the prediction can be merely mined from the single tweet.

Neural networks have been utilized for SMRC, which produced substantial improvements so far (Sect. 2). In particular, large pretrained language models were used to strengthen encoders of current SMRC models, such as BERT [5], UniLM [6] and T5 [7]. Due to extensive learning over large-scale data for semantic representation, the pretrained models significantly improve the understanding of questions and tweets, and therefore, boost SMRC performance. It is noteworthy that such pretrained models need to be fine-tuned over TweetQA in the mode of transfer learning, and necessarily accompanied with proper data cleaning strategies [8]. Transfer learning is applied for enhancing adaptation to domain-specific characteristics of tweets, such as that for the idiom where the stop words "*Down Under*" actually serve as the alternative name of "*Australia*". Data cleaning is used to recover or filter grammatical errors, such as the removal of redundant spaces "*did n't*" into "*didn't*".

Briefly, the existing neural SMRC models achieve promising performance when transferring pretrained models to tweets and coupling them with data cleaning. However, our empirical findings show that entity-oriented SMRC fails to perform perfectly, where the state-of-the-art model such as NUT-RC [8] obtains an error rate of 40.94%[2]. Though, the most noticeable fact regarding data distribution is that the proportion of entity-type answers is up to 29.13%[3] in all SMRC instances in TweetQA dataset.

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

[2] We reproduce NUT-RC [8] and evaluate it on the development set. On the basis, we verify the error rate for entity-oriented SMRC.

[3] We employ an off-the-shelf Named Entity Recognition (NER) toolkit Twitter-Stanza to automatically determine whether gold SMRC answers are the ones containing named entities. The toolkit has been well-trained on the TweeTbank-NER dataset (https://github.com/social-machines/TweebankNLP).

Entity-oriented SMRC instances refer to the ones whose ground-truth answers are entity mentions, such as names of person (PER), organization (ORG) and location-type (LOC) entities. The reason why SMRC models fall into the misjudgement for some of them is because of the unawareness of entity knowledge [9]. For example, the clue for reasoning in the case in Table 1 is evident (i.e., the text "*W NJ has the most to lose*" which is even consistent with the question in morphology and pragmatics), though SMRC models fail to identify the entity "*W NJ*" (i.e., the abbreviated mention of "*West New Jersey*") in it as the answer. It reveals the possibility that, to the end, SMRC models are unaware of what the mention "*W NJ*" is, or even regard it as a sequence of meaningless characters instead of the closely related entity to the "*Who*"-type question.

To address the issue, we propose to enhance the awareness of entity mentions during encoding questions and tweet contexts. The two-channel multi-task learning is utilized, where SMRC and NER tasks are considered. The shareable encoder across the two learning channels is trained to perceive interaction between question and tweet context, as well as the latent information of various entity mentions. This contributes to the construction of a self-contained entity-aware SMRC model. We experiment on the benchmark dataset TweetQA [4]. Experimental results show that our method yields substantial improvements, and it outperforms the published state-of-the-art model NUT-RC [8].

2 Related Work

A variety of innovative approaches have been proposed for SMRC. Huang et al. (2020) [8] design heuristic rules to standardize informal texts in tweets. More importantly, Huang et al. (2020) bridge generative and extractive SMRC by answer selection mechanisms. Tian et al. (2021) [10] enhance the representation learning of questions and tweets using concepts. Hashtags are used as concepts. They are extracted from the closely-related tweets, the ones retrieved and highly-ranked in terms of topic-level relevance. BERT-based pointer network is utilized for extracting concepts. Xue et al. (2022) [11] demonstrate the effectiveness of character-level models in dealing with noisy and informal datasets, such as TweetQA [4]. Instead of using the limited vocabulary to tokenize words, they directly take the UTF-8 bytes as the input. On the basis, a character-level pre-trained model is developed based on T5 [7] architecture.

Our approach is different from aforementioned approaches. We capture the exclusive characteristics that some of entity mentions play an important role for reasoning answers in tweets, or even serve as answers themselves. Accordingly, we intend to enhance the awareness of entity knowledge when encoding questions and tweets. To pursue the goal, we utilize the entity recognition as an auxiliary task, so as to drive the encoder to perceive and represent entity mentions.

3 Approach

In this section, we present the components of our SMRC model step by step, including preprocessing over tweets, entity-aware encoding by multi-task learning, as well as answer prediction.

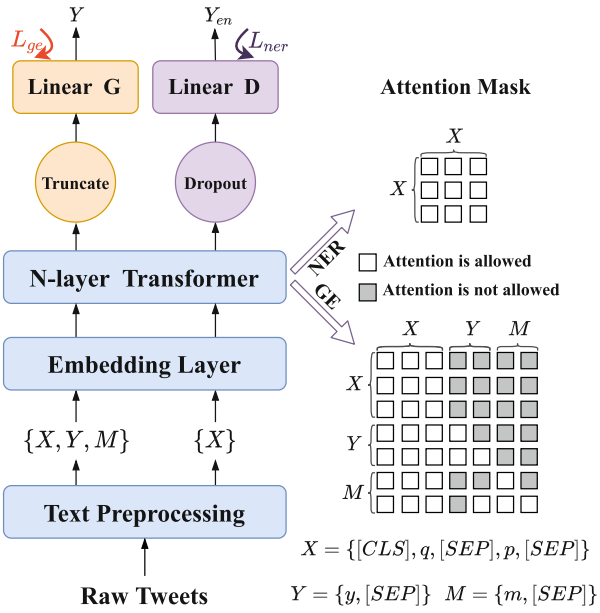

Fig. 1. Architecture of our multi-task learning model, which is used to enhance the awareness of entity information during encoding.

3.1 Text Preprocessing

We employ Huang et al. (2020)'s heuristic rules [8] to separate the mixed-tokens in tweets. Specifically, we split both `Hashtags` and User-Ids into formal texts (e.g., "*#WhiteHouse*"→"*# White House*"), so as to avoid the misunderstanding or omission of entity mentions.

3.2 Entity-Aware Encoding Grounded on Multi-task Learning

We conduct two-channel multi-task learning, where shareable multi-layer transformer encoders are used. One learning channel aims to train the encoder for generative SMRC (primary task), while the other performs for NER (auxiliary task). Different attention masks are utilized in the two learning channels. The neural network we used is constructed with embedding layer and transformer encoders, as well as two separate decoding layers coupled with truncation and dropout operations. Figure 1 shows the architecture of our learning model.

Embedding Layer: Given an SMRC instance in the training set, we construct two kinds of input sequences for SMRC and NER tasks respectively. For NER, we concatenate the token sequence of question q and that of tweet p, where the special tokens "*[CLS]*" and "*[SEP]*" are used (see the concatenation mode in Fig. 1). The resultant input sequence is denoted as X. For generative SMRC, we concatenate X with two additional sequences Y and M. Y comprises tokens of the ground-truth answer y and "*[SEP]*". M serves as the shuffled version of Y. Specifically, the tokens in M are duplicated from Y, though at the initialization stage, they are randomly masked by special character "*[MASK]*" or replaced with other words in the vocabulary [12]. M is primarily used for pseudo-masked fine-tuning, which contributes to the alleviation of exposure bias.

Following Devlin et al. (2019)'s practice [5], we obtain the input embeddings by conducting element-wise aggregation over token, segment and position embeddings. It is noteworthy that the embedding layer is trainable.

Encoder Layers: We apply N-layer Transformer encoders of UniLM v2.0 to convert the input embeddings to contextual semantic representations, no matter whether the input is {X,Y,M} or {X} ("{∗}" denotes concatenation operation).

$$H^l = Transformer(H^{l-1}) \tag{1}$$

where, $l \in [1, N]$ signals the l-th transformer layer which produces the hidden states H^l. H^l contains the token-level hidden states of all tokens and special characters in the input sequence. We use H^N as the final hidden states, i.e., the distributed representations output by the last (N-th) transformer layer. For generative SMRC, the final hidden states act as $H^N = \{h_1^N, h_2^N, ..., h_{s+t+t}^N\}$, where, s and t constrain the maximum length of H^N which are numbers of tokens in X and Y. For NER, the final hidden states act as $\check{H}^N = \{\check{h}_1^N, \check{h}_2^N, ..., \check{h}_s^N\}$.

Selective masking mechanism is required to perform during training due to the different prediction modes (decoding modes) of generative SMRC and NER. Specifically, generative SMRC serves as a generation model, and therefore needs to possess the capacity of predicting the current token in terms of preceding predictions (and tweet context). In fact, this recursive prediction mode conforms to the fundamental limitation that ground-truth answer Y is invisible in the test process. In order to simulate the recursive prediction mode, we need to impose masks on the hidden states of subsequent tokens in H^N during training. By contrast, NER serves as a sequence labeling task, which performs B/I/O tag classification for each token separately and independently. Therefore, it is unnecessary to impose masks over the hidden states $\check{H}^N$. To facilitate the representation learning of shareable encoders between the two tasks, we establish a selective masking mechanism, where Bao et al. (2020)'s pseudo-masked attention learning [13] is used. The attention score $ATTN_l$ of the transformer l is computed as follows:

$$ATTN_l = (\frac{Q_l K_l^T}{\sqrt{d_k}} + MASK)v_l \tag{2}$$

Table 2. An example of named entities for questions and tweets

Question+Tweet: *[CLS] Who have the cavs released? [SEP] The Cavs have released Edy Tavares . No surprise . He was on a non - guaranteed contract . Roster stands at 19 . - Jason Lloyd (@ Jason Lloyd NBA) [SEP]*
Named Entities: *[CLS] O O O S-ORG O [SEP] O S-ORG O O B-PER I-PER O O O O O O O O O O O O S-ORG O O O O O B-PER I-PER O O O O O O [SEP]*

$$MASK_{ij} = \begin{cases} 0, & \text{Attention is allowed} \\ -\infty, & \text{Attention is not allowed} \end{cases} \tag{3}$$

where, Q_l,K_l,V_l respectively denote the **Q**uery, **K**ey and **V**alue vectors that are obtained by linearly converting H^N or $\check{H}^N$. $MASK$ denotes the attention mask. Figure 1 shows the diagrams of masked hidden states at a certain encoding step.

Decoding of SMRC and Loss Estimation: Given the pseudo-masked final hidden states H^N, we take the hidden states H_M^N of M out of H^N by truncation, which contain latent information for predicting answers. We feed H_M^N into the linear layer G with Softmax to compute the probability distribution that every token in the vocabulary serves as an answer or part of it:

$$\begin{cases} H_M^N = [h_{s+t+1}^N, h_{s+t+2}^N, ..., h_{s+t+t}^N] \\ Y_{ge} = softmax(Linear_G(H_M^N)) \end{cases} \tag{4}$$

During training, the loss of answer prediction is estimated with the probability distribution Y_{ge}. It is the reliance for back propagation. Cross entropy f_{CE} is used to estimate the loss $\mathcal{L}_{ge}$ (where Y denotes the ground-truth answer):

$$\mathcal{L}_{ge} = f_{CE}(Y_{ge}, Y) \tag{5}$$

Decoding of NER and Loss Estimation: Given the final hidden states $\check{H}^N$, we feed them into a dropout layer for purifying their latent information. This helps to avoid overfitting. On the basis, we deliver the purified hidden states $\check{H}^N$ to the linear layer D with Softmax, so as to predict the probability distributions Y_{en} over B/I/O tags for each token. Similarly, we utilize cross entropy f_{CE} to estimate the loss $\mathcal{L}_{ne}$. All computations of NER for decoding are as follows (where Y_{en} denotes the ground-truth B/I/O tags of NER):

$$\begin{cases} \check{H}^{N'} = dropout(\check{H}^N) \\ \check{Y}_{en} = softmax(Linear_D(\check{H}^{N'})) \\ \mathcal{L}_{ne} = f_{CE}(\check{Y}_{en}, Y_{en}) \end{cases} \tag{6}$$

The learning in the channel of NER, frankly, requires the ground-truth B/I/O tags of entity mentions for supervision. However, TweetQA dataset doesn't possess annotation results of named entities. Therefore, we use the existing NER

toolkit Twitter-Stanza [14] to automatically annotate named entities of both questions and tweets. Table 2 shows the example regarding B/I/O tags of entities towards a SMRC case in TweetQA.

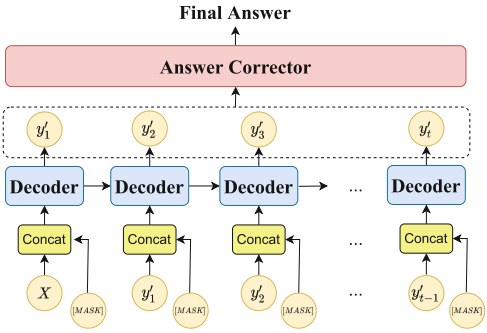

Fig. 2. The data flow produced step by step during the decoding process.

Multi-task Learning: During training, we conduct supervised learning for generative SMRC and NER tasks alternatively and iteratively in each epoch. Both the losses produced in SMRC and NER are jointly used to optimize the parameters in embedding layer, transformer encoders and predictors (i.e., generator G for SMRC while discriminator D for NER). We compute the joint loss $\mathcal{L}_{all}$ as follows (where λ denotes a trade-off coefficient):

$$\mathcal{L}_{final} = \mathcal{L}_{ge} + \lambda \mathcal{L}_{ner} \tag{7}$$

3.3 Generating Answers

Instead of extracting answers from tweets (by pointer networks), we generate answers, i.e., search the most possible tokens in the vocabulary to sequentially constitute an answer, where the greedy algorithm is used.

Specifically, conditioned on the i-th hidden state h_i^N in H_M^N (see Eq. 4), we predict the i-th token y_i' of the possible answer at the i-th time step. In order to speed up decoding, we hold up emebeddings of $\{X, y_1', ..., y_i'\}$ for each time step at run time, and concatenate them with emebedding of $\{y_i', [MASK]\}$. The resultant representation will be fed into the encoder to produce $(i+1)$-th hidden sate h_{i+1}^N in H_M^N (See Fig. 2). In this way, we iteratively predict tokens in the answer and produce the next hidden state until "*[SEP]*" is predicted.

In addition, we design an answer corrector to post-process the generated answer. It is capable of dealing with the following informal text spans. The major heuristic rules including 1) **Word Recovery** (e.g., "*did n't*"→"*didn't*") and 2) **Removing Redundant Characters** (e.g., removing "@" or "#").

4 Experimentation

4.1 Data, Evaluation and Hyperparameter Settings

- **Dataset:** We experiment on TweetQA [4]. Compared to other MRC datasets, TweetQA [4] contain a large number of unusual entities. More importantly, the answers in TweetQA [4] are free-form texts rather than the ones toughly extracted from tweets. We follow the previous work to split TweetQA. The training, validation and test sets contain 10,692, 1,086 and 1,979 instance, respecitvely.

Table 3. Performance of the state-of-the-art SMRC models and ours.

Model	BLEU-1		Meteor		Rouge-L	
	Dev	Test	Dev	Test	Dev	Test
BIDAF (Seo et al., 2016) [19]	48.3	48.7	31.6	31.4	38.9	38.6
Seq2Seq (Song et al., 2017) [18]	53.4	36.1	32.1	31.8	39.5	39.0
BERT-EX (Devlin et al., 2018) [5]	61.0	58.4	64.2	63.2	60.9	65.8
NUT-RC (Huang et al., 2020) [8]	78.2	76.1	73.3	72.1	79.6	77.9
TKR (Tian et al., 2021) [10]	68.7	69.0	64.7	65.6	70.6	71.2
EA-SMRC (Original)	**79.1**	**78.5**	**74.5**	**74.7**	**80.6**	**80.0**
EA-SMRC (Variant)	78.7	77.8	**74.5**	74.2	80.1	79.4

- **Evaluation Metrics:** For comparison, we follow the common practice to use BLEU-1 [15], Meteor [16] and Rouge-L [17] to evaluate SMRC models. The test set is not publicly available. Therefore, we submit the predicted answers to the official website of TweetQA [1] for obtaining the test performance.
- **Hyperparameter Settings:** Our source code is based on s2s-ft [12]. We use the Adam optimizer to train the MRC model. The learning rate for training is 2e-5. We set the maximum length of X to 128 and the maximum length of Y to 24. We initialize our model using the parameters of UniLM v1.2 [12], and fine-tune our model on TweetQA in 10 epochs. The batch size for training is 12. The opimal λ is set to 1. The dropout rate used for the NER task is set to 0.1.

4.2 State-of-the-art SMRC Models for Comparison

We develop two versions of SMRC models, including the aforementioned entity-aware SMRC grounded on multi-task learning (denoted as original **EA-SMRC**), as well as its variant. The variant adopts the same learning architecture, though the auxiliary task NER is implemented by Masked Language Modeling (MLM), where MLM of BERT is transferred to the learning process.

We compare our models to the state-of-the-art models including 1) **BIDAF** [4] which is an extractive MRC model based on Recurrent Neural Network

(RNN), where bi-directional attention flow is used; 2) **Seq2Seq** [18] which acts as a generative model within the RNN-based encoder-decoder framework, where copy and coverage mechanisms are leveraged; 3) **BERT-EX** [5] which is obtained by transferring the pretrained language model BERT to TweetQA and acts as an extractive MRC model; 4) **TKR** [10] which incorporates concept knowledge into the encoding process, so as to enhance the perception and representation of unusual linguistic units, where external data is applied for retrieving concept knowledge; and 5) **NUT-RC** [8] which possesses a two-channel multi-task learning architecture, where generative and extractive MRC are conducted in the two channels, and answer selection is used.

Table 4. Ablation study on TweetQA

Model	BLEU-1		Meteor		Rouge-L	
	Dev	Test	Dev	Test	Dev	Test
EA-SMRC (Original)	**79.1**	**78.5**	**74.5**	**74.7**	**80.6**	**80.0**
-NER	77.7	77.3	73.2	73.6	79.2	78.9
-NER&-CORR	75.6	76.6	71.4	72.6	77.5	78.2
-NER&-CORR&-SPLIT	74.0	75.5	69.8	71.4	75.9	77.3

4.3 Main Results

Table 3 shows the test results of our models (EA-SMRC) and the state of the art. It can be observed that both original and variant EA-SMRC models produce substantial performance gains, compared to previous work. Considering that both the models utilize entity-aware multi-task learning framework, we suggest that the proposed method is robust and capable of yielding steady improvements to some extent. Experimental results reveal the fact that original EA-SMRC achieves higher performance (BLEU-1, Meteor and Rouge-L scores) on TweetQA [4]. It is because the original version accurately introduces entity knowledge into the SMRC, while the variant one still requires to understand the semantics of context to infer the entity types, which potentially brings a certain noise due to the inadequate semantic understanding.

We concentrate on the previous work of NUT-RC [8] for advantage analysis, which used to stand on the top of leader board for a long period of time and, more importantly, it holds the same learning framework with our models (i.e., multi-task learning). From the perspective of effectiveness, our EA-SMRC models obtain better performance due to the incorporation of entity knowledge into learning. From the perspective of efficiency, frankly, EA-SMRC is relatively vest-pocket and less time-consuming because the kernel is constituted with a group of transformer encoders and two independent linear layer. By contrast, NUT-RC

possesses two groups of large transformer blocks in the learning channels, which are initialized by UniLM v1.0 [6] and BERT-Large [5].

4.4 Ablation Study

We carry out ablation experiments to verify the effects of different components in EA-SMRC. The components are progressively ablated, which include 1) "-NER" denoting the ablation of the auxiliary task NER, which boils multi-task learning down to entity-unaware single-task learning, 2) "-CORR" referring to the condition that answer correction is disable, and 3) "-SPLIT" that refers to the ablation of heuristic rules for text preprocessing.

Table 4 show the experimental results. It can be found that performance constantly degrades when the components are progressively ablated. During test, the largest performance reduction results from the ablation of NER. It proves the dominant positive effect of entity-aware multi-task learning.

Table 5. Performance obtained when different pretrained models are used

Model	Framework	BLEU-1	Meteor	Rouge-L
UniLM v1.0 [6]	FT	72.5	67.5	74.5
	MTL	**73.6**	**68.4**	**75.1**
UniLM v2.0 [13]	FT	71.0	65.7	73.1
	MTL	**72.2**	**66.9**	**74.0**
BERT [5]	FT	69.7	65.4	71.5
	MTL	**70.3**	**65.6**	**72.0**

Table 6. Performance of EA-SMRC (Variant) on TweetQA using different NER tools

NER tool	BLEU-1	Meteor	Rouge-L
CoreNLP [21]	77.6	63.7	79.5
Stanza [22]	78.2	74.0	79.6
Twitter-Stanza [14]	**79.1**	**74.5**	**80.6**

4.5 Effects of Different Pretrained Models for Transfer

We verified the performance of EA-SMRC on the validation set when different pretrained models are used for initialization. Initialization is conducted by substituting off-the-shelf parameters and embeddings of pretrained models into EA-SMRC. This enables transfer learning on TweetQA within multi-task learning

framework. We consider three pretrained models, including UniLM v1.0, UniLM v2.0 and BERT. The variant EA-SMRC is used due to its better performance on the validation set. Besides, two learning frameworks are considered, including our entity-aware multi-task learning (denoted as MTL) and entity-unaware single-task learning. The latter is equivalent to the case that pretrained models are directly transferred to TweetQA and fine-tuned there (denoted as FT).

Table 5 shows the performance of aforementioned pretrained models. It can be observed that utilizing different pretrained models will result in significantly performance. Nevertheless, all the models can achieve better performance when the MTL framework is used, compared to the FT framework. It illustrates that our entity-aware learning strategy generalizes well. Besides, it can be found that both UniLM v1.0 and UniLM v2.0 fail to produce competitive performance, compared to UniLM v1.2 in our EA-SMRC (see Table 3). It is because that UniLM1.2 doesn't apply relative position bias [20], and thus it is adaptive to the stationary position embeddings in our input layer.

4.6 Utility of NER Toolkits

We verify the utility of different NER toolkits in our method. Note that NER toolkits are used for obtain entity mentions in the training data, which support the learning of a self-contained encoder for perceiving entities. We consider three NER toolkits, including CoreNLP [21], Stanza [22] and Twitter-Stanza [14]. The former two provide a larger number of entity types (23 in CoreNLP and 18 in Stanza) and instances, compared to Twitter-Stanza. Nevertheless, the training data of Twitter-Stanza derives from the same domain with TweetQA.

Table 6 shows the experimental results. It can be observed that Twitter-Stanza yields relatively-substantial performance gains. It proves that domain relevance is more important than both data size and versatility of entity types for the adoption and utilization of NER toolkits.

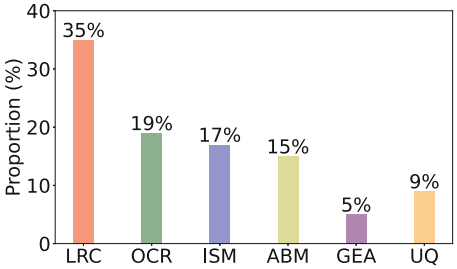

Fig. 3. The proportion of different error types.

Table 7. Examples of prediction errors produced by our SMRC models.

Type	Example
LRC	**Question:** *Who are they replying to?* **Tweet:** *This looks like blank space taken to a NEW.LEVEL. (@dunderswiftlin).* **Gold Answer:** *Gma and Taylor Swift* (**Reasons behind errors**: Be unaware of exact names of "*dunderswiftlin*" and their relationship to "*NEW.LEVEL*)
OCR	**Question:** *Who wouldn't give a long-term deal?* **Tweet:** *The Red Wings didn't believe they would get Mike Green because they wouldn't give a long-term deal.* **Gold Answer:** *The Red Wings.* (**Reasons behind errors**: Fail to correspond the co-reference "*they*" to the entity "*Red Wings*")

4.7 Error Analysis

We conduct error analysis on the predictions of our models over the validation set. The errors are caused by six classes of drawbacks, including Lack of Related Commonsense (**LRC** for short), Omission of Co-reference Resolution (**OCR**), Incorrect Segmentation of Mixed-tokens (**ISM**), Answer Boundary Misjudgement (**ABM**), Grammar Errors of the generated Answers (**GEA**), as well as Unanswerable Questions (**UQ**) caused by inexact or improper annotations. Figure 3 shows the proportions of aforementioned error types in all the misjudged answers. Table 7 gives two examples of prediction errors.

5 Conclusion

We propose an entity-aware encoding method to strengthen the current SMRC models. Multi-task learning is leveraged to enable the perception and fusion of latent information of entity mentions. Experiments on the benchmark dataset TweetQA demonstrate the effectiveness of our method. Besides of superior performance (higher BLEU-1, Meteor and Rouge-L scores), our SMRC model is vest-pocket and less time-consuming. In the future, we will enhance the entity-aware encoder from two aspects, including 1) introducing external knowledge of entities into the representation learning process, where group-based neural models will be used, and 2) conducting co-reference resolution.

Acknowledgements. This project is supported by National Key R&D Program of China (No. 2020YFB1313601) and National Natural Science Foundation of China (No.62076174 and No. 61836007).

References

1. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2383–2392 (2016)

2. Reddy, S., Chen, D., Manning, C.D.: CoQA: a conversational question answering challenge. Trans. Assoc. Comput. Linguist. **7**, 249–266 (2019)
3. Kočiskỳ, T., et al.: The narrativeqa reading comprehension challenge. Trans. Assoc. Comput. Linguist. **6**, 317–328 (2018)
4. Xiong, W., et al.: TWEETQA: a social media focused question answering dataset. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5020–5031. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1496
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423
6. Dong, L., et al.: Unified language model pre-training for natural language understanding and generation. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
7. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res.**21**(140), 1–67 (2020). http://jmlr.org/papers/v21/20-074.html
8. Huang, R., Zou, B., Hong, Y., Zhang, W., Aw, A., Zhou, G.: NUT-Rc: noisy user-generated text-oriented reading comprehension. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 2687–2698 (2020)
9. Shao, Y., Bhutani, N., Rahman, S., Hruschka, E.: Low-resource entity set expansion: a comprehensive study on user-generated text. In: Findings of the Association for Computational Linguistics: NAACL 2022. pp. 1343–1353. Association for Computational Linguistics, Seattle, United States (2022). https://aclanthology.org/2022.findings-naacl.100
10. Tian, Z., Zhang, Y., Liu, K., Zhao, J.: Topic knowledge acquisition and utilization for machine reading comprehension in social media domain. In: Proceedings of the 20th Chinese National Conference on Computational Linguistics. pp. 988–999. Chinese Information Processing Society of China, Huhhot, China (2021). https://aclanthology.org/2021.ccl-1.88
11. Xue, L., et al.: ByT5: towards a token-free future with pre-trained byte-to-byte models. Trans. Assoc. Comput. Linguist. **10**, 291–306 (2022)
12. Bao, H., Dong, L., Wang, W., Yang, N., Wei, F.: s2s-ft: fine-tuning pre-trained transformer encoders for sequence-to-sequence learning. arXiv preprint arXiv:2110.13640 (2021)
13. Bao, H., et al.: UniLMv2: pseudo-masked language models for unified language model pre-training. In: International Conference on Machine Learning, PMLR. pp. 642–652 (2020)
14. Jiang, H., Hua, Y., Beeferman, D., Roy, D.: Annotating the tweebank corpus on named entity recognition and building NLP models for social media analysis. arXiv preprint arXiv:2201.07281 (2022)
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA. pp. 311–318(2002). https://doi.org/10.3115/1073083.1073135

16. Denkowski, M., Lavie, A.: Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Edinburgh, Scotland, pp. 85–91 (2011), https://aclanthology.org/W11-2107
17. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. Association for Computational Linguistics, Barcelona, Spain. pp. 74–81 (2004). https://aclanthology.org/W04-1013
18. Song, L., Wang, Z., Hamza, W.: A unified query-based generative model for question generation and question answering. arXiv preprint arXiv:1709.01058 (2017)
19. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603 (2016)
20. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
21. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55–60 (2014)
22. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics, pp. 101–108.(2020). https://doi.org/10.18653/v1/2020.acl-demos.14