

Controllable Abstractive Sentence Summarization with Guiding Entities

Changmeng Zheng^{1,2}, Yi Cai^{1,2*}, Guanjie Zhang¹, Qing Li³

¹School of Software Engineering, South China University of Technology, Guangzhou, China

²Key Laboratory of Big Data and Intelligent Robot (SCUT), Ministry of Education, China

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

sethecharm@mail.scut.edu.cn, ycai@scut.edu.cn

Abstract

Entities are the major proportion and build up the topic of text summaries. Although existing text summarization models can produce promising results of automatic metrics, for example, ROUGE, it is difficult to guarantee that an entity is contained in generated summaries. In this paper, we propose a controllable abstractive sentence summarization model which generates summaries with guiding entities. Instead of generating summaries from left to right, we start with a selected entity, generate the left part first, then the right part of a complete summary. Compared to previous entity-based text summarization models, our method can ensure that entities appear in final output summaries rather than generating the complete sentence with implicit entity and article representations. Our model can also generate more novel entities with them incorporated into outputs directly. To evaluate the informativeness of the proposed model, we develop a fine-grained informativeness metrics in the relevance, extraneousness and omission perspectives. We conduct experiments in two widely-used sentence summarization datasets and experimental results show that our model outperforms the state-of-the-art methods in both automatic evaluation scores and informativeness metrics.

1 Introduction

Automatic Text Summarization is a task of shortening and creating a concise version of a text that represents the most important or relevant information within the original text. The task can be divided into two subtasks: extractive and abstractive summarization. Recently, the research trend (Nallapati et al., 2016; Tan et al., 2017; Bansal and Passonneau, 2018; Kryściński et al., 2018) has shifted towards abstractive summarization since it can produce more natural and coherent texts than extractive methods.

Sequence-to-sequence models with attention mechanism (Chopra et al., 2016; Nallapati et al., 2016) have attracted growing attention due to their state-of-the-art performance on abstractive summarization. However, most previous seq2seq models have two obvious limitations:

The first limitation is that existing methods are difficult to identify salient entities and related events in the original articles (Cao et al., 2018). As a result, they are frequently found to be unfaithful to the input and lack of topic coherence. According to our observations, entities are important for summary generation in three points: (1) Summaries are mainly composed by entities extracted from the original texts. 77.1% and 71.5% of the noun phrases on the Gigaword and DUC-2004 datasets contain at least one entity, respectively. (2) Entities can reveal the main topics in an article (Newman et al., 2006), which can be utilized to generate the summaries of highly topic coherence. For example, the PER entity “KOBE-BRYANT” and the ORG entity “NBA” are mostly associated with topic “games” and “players”. (3) Entities (e.g. person names, locations and organizations) can provide some basic answers to the Five Ws questions (Who, What, When, Where, Why) in information gathering or problem solving. Although a summary is a syntax compression version of texts, it should not miss the key details which provide a better understanding for the original document. For example, a human-annotated summary in NYT dataset

*Corresponding author

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>.

is “Prime Min Bertie Ahern of Ireland calls for general election on May 24.” In this sentence, “Prime Min Bertie Ahern” and “Ireland” are the named entities which reveal the important person and location information in this event. Previous seq2seq models tend to generate sequence with high-frequency words like “A man calls for general election” which is less informative and cannot help people to understand this event. Considering above issues, it is necessary to incorporate entities in summaries.

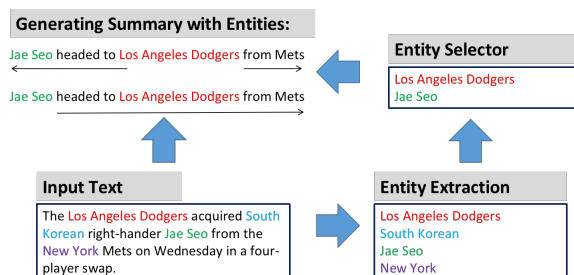


Figure 1: An overview of our model which generates summaries with guiding entities. Instead of generating summaries from left to right, our approach can control the process of generation and incorporate selected entities into summaries precisely.

The second limitation is that existing methods are limited in control of the generation process and cannot guarantee to generate summaries with important named entities. Previous entity-based summarization models (Amplayo et al., 2018; Sharma et al., 2019) extract entities and concatenate attentive entity representations with original articles to generate summaries. However, their methods generate the summaries from left to right and the decoder is initialized with implicit entity and article representations. Such implicit representations cannot guarantee entities to appear in final outputs. Including a novel entity in the summary is even harder since the entity has not been seen in training data. As a result, it is necessary to develop a controllable framework which incorporates entity information and generates summaries with selected entities.

To address above limitations, we propose a controllable neural model which generates summaries with guiding entities. As shown in Figure 1, we first extract entities in the original texts using a pretrained named entity recognition model (Devlin et al., 2019). Considering that only a few of these entities should appear in final summaries, an entity selector will identify the most important entities and send the entity representations to the summary generation phase. Previous works (Amplayo et al., 2018; Sharma et al., 2019) generates summaries from left to right, the entity information is encoded implicitly and may not appear in generated summaries. We handle this problem by combining two LSTMs to generate the left part and the right part of the sequence around the entity. Each LSTM encodes information of the other part of the sequence and then generates a summary based on the encoded article and entity representations. This allows the two LSTM decoders to connect with the selected entities fluently. As a result, we can guarantee that important entities appear in output summaries. Since the novel entities are incorporated into the generated sentences directly, our model can also generate more novel entities without going through the encoding-decoding process.

In summary, we make the following major contributions in this paper:

- We propose a controllable neural network for abstractive summarization with guiding entities. Our model can generate summaries of improved content accuracy and topic coherence by combining entity information with sentence representations. Compared to previous implicit entity-based models, our model can guarantee entities to appear in output summaries and generate more novel entities with them incorporated directly.
- To evaluate the informativeness of the proposed method, we develop a fine-grained informativeness metric which assesses the output quality in semantic vector level. The metric can evaluate not only the relevance between model outputs and manual references, but also the extra and omissive information in generated summaries. Compared to existing lexicon-based metrics, it is more human-like and helps to analyze results comprehensively.

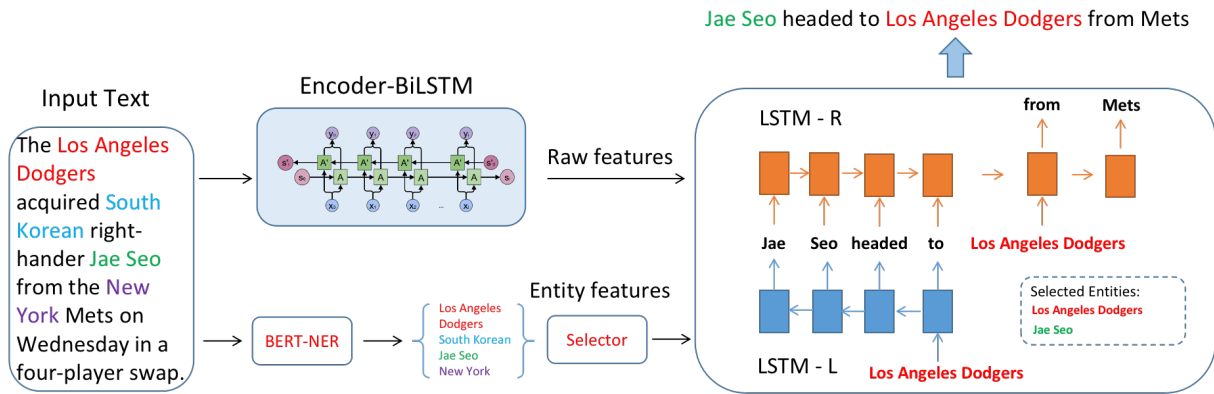


Figure 2: Our controllable neural model with guiding entities. The original article texts are encoded with a BiLSTM layer. We utilize a pretrained BERT named entity recognition tool to extract entities from input texts. The decoder consists of two LSTMs: LSTM-L and LSTM-R. Our model starts generating the left and right part of a summary with selected entities and can guarantee that entities appear in final output summaries.

- We conduct our experiments on Gigaword and DUC-2004 datasets and our model outperforms state-of-the-art methods in both ROUGE scores and informativeness metrics.

2 Related Work

Neural Abstractive Summarization. Recent years have witnessed the success in abstractive summarization using encoder-decoder framework with sequence-to-sequence models (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017; Celikyilmaz et al., 2018). The encoder which is leveraged for syntactic compression can be implemented using recurrent neural networks (Chopra et al., 2016; Tan et al., 2017; Chen and Bansal, 2018), convolutional networks (Allamanis et al., 2016; Liu et al., 2018) and transformer-based methods (Devlin et al., 2019; Song et al., 2020b). To handle the problem that many OOV words are generated by vanilla sequence-to-sequence decoder, copy mechanism is proposed to copy a word from the source text or select an unseen word from the vocabulary (See et al., 2017; Zhou et al., 2018; Wang et al., 2019). However, their methods ignore the important named entities which are major proportion and build up the topic of text summaries. As a result, their methods may miss such important information in output summaries.

Entity-based Text Summarization. Entities have been studied on the summarization tasks in the preprocessing steps to anonymize text data (Nallapati et al., 2016) or to mitigate OOV problems (Tan et al., 2017). (Amplayo et al., 2018) find it beneficial to use linked entities in decoder phase to improve the summarization performance. Their method fails in capturing the entities that do not exist in the knowledge bases. (Sharma et al., 2019) propose a two-step method which combines the salient content selector and entity-based decoder to improve the coherence for abstractive summarization. However, the above methods are based on the sequence-to-sequence framework which generates summaries from left to right and cannot guarantee that the entities appear in final output summaries. To the best of our knowledge, we are the first to use two separate LSTMs to generate summaries from guiding entities.

3 Abstractive Summarization with Guiding Entities

In this paper, we propose a neural abstractive summarization model which leverages named entity information in original articles to generate informative summaries. The architecture is illustrated in Figure 2. Given an input article, our model is able to incorporate the selected entities into generated summaries. Unlike previous works generating summaries from left to right and decoding with implicit entity representations, our model combines two LSTMs to generate the partial summaries on two sides of the guiding entities.

The overall architecture is built on the widely-used encoder-decoder framework. In encoding phase, we extract original article representations with a Bi-LSTM layer. The named entities are identified by a pretrained BERT-NER (Devlin et al., 2019) model. We select the top K important entities and transform them into word embeddings. In decoding phase, We use LSTM-L to denote the LSTM layer generating the left part of the summary and LSTM-R to generate the right part. The two LSTM layers are training simultaneously with a multitask loss, which can share the same article features and entity information. We will describe each part of our model in detail.

3.1 Input Article Encoder

For article encoder, we employ the basic bidirectional LSTM to extract sequence features following the same settings of (Sharma et al., 2019). Each token in the sentence is represented with the word embedding, denoting as x_i . The BiLSTM layer consists of a forward and backward LSTM, which outputs sequences of forward and backward hidden states: $(\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n)$ and $(\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_n)$, respectively:

$$\vec{h}_i = \overrightarrow{\text{LSTM}}(x_i, \overrightarrow{h_{i-1}}) \quad (1)$$

$$\overleftarrow{h}_i = \overleftarrow{\text{LSTM}}(x_i, \overleftarrow{h_{i-1}}) \quad (2)$$

$$\mathbf{h}_i = [\vec{h}_i; \overleftarrow{h}_i] \quad (3)$$

$$\mathbf{f} = [\mathbf{h}_n; \mathbf{h}_1] \quad (4)$$

We concatenate the forward and backward hidden states of each token as \mathbf{h}_i . The final states of the forward and backward (\mathbf{h}_n and \mathbf{h}_1) LSTM are also concatenated and to form the initial states \mathbf{f} of the two decoders.

3.2 Entity Encoder

Entities in the original texts provide informative insights for understanding text semantic and generate more concrete summaries. As the main components of noun phrases in summaries, entities can reflect the topics and also, they can be learned as a continuous vector representation. For example, we can find the ‘‘Los Angeles Dodgers’’ in Figure 1 as an American baseball team, such entity can be transformed into the same semantic as baseball team with a feature extractor trained on a large corpus.

For each input article, we first extract the entities with a pretrained BERT model (Devlin et al., 2019). Previous entity-based summarization works identify entities with a co-reference resolution system (Sharma et al., 2019) or an entity linking system (Amplayo et al., 2018), which rely on the performance of the off-the-shelf systems. Following the success of pretrained models, we choose BERT as an entity extractor, which denotes a better accuracy in identifying named entities.

For training phase, we extract entities in ground-truth summaries. According to our observations, not all the entities from original articles appear in summaries. As a result, for testing phase we simply choose the top K entities with respect to their word frequency in the current article. These entities are embedded to d -dimensional vectors $E = \{e_1, e_2, \dots, e_m\}$ where $e_i \in \mathbb{R}^d$. We use another Bi-LSTM to encode the entity embeddings and concatenate the forward and backward hidden state vectors as the final entity representations.

3.3 LSTM-L

For a given article A and the extracted entity y_k which is located in the position k of current summary, we first use the LSTM-L to generate the left-side partial summaries. At each time step t , LSTM-L predict the former word conditioned on the article representations \mathbf{f}_t , the input word x_t , and the hidden state \mathbf{h}_{t-1} .

$$p(y_t | y_{t+1}, \dots, y_k) = \text{LSTM} - L(\mathbf{f}_t, x_t, \mathbf{h}_{t-1}) \quad (5)$$

Following the work of (Sharma et al., 2019), we adopt a sequence-to-sequence network with attention mechanism here (Bahdanau et al., 2015). The article features of attention weights in each time step are

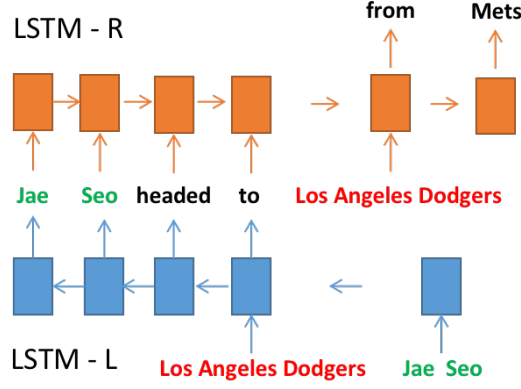


Figure 3: The enhanced abstract generator of our LSTM-L module. To make model generate different entities, we encode all possible entities to initialize the LSTM-L generator. This can also guide the LSTM-R to generate different entities.

denoted as \mathbf{f}_t . Here we apply the soft attention mechanism which denotes as ATT to capture the relevance of encoder vectors and input words.

$$\mathbf{a}_t = ATT(x_t, \mathbf{h}_{t+1}) \quad (6)$$

$$\mathbf{f}_t = \mathbf{f} \odot \mathbf{a}_t \quad (7)$$

Particularly, we observe that a summary may contain more than one entities. However, our approach only can ensure that one entity appears in the final summary. Still, we can arrange entities in different orders. For instance, in Figure 3, we can generate summaries starting with “Los Angeles Dodgers” or “Jae Seo” in our LSTM-L generator. we assume the entity sequence $S = \{e_1, \dots, e_m\}$ contains information from different entities, where m indicates the number of entities in the selected entity set. To further incorporate entity information from all selected entities, the LSTM-L now predicts the summaries according to the article representations \mathbf{f}_t , the selected entities S and the starting entity y_k .

$$p(y_t | y_{t+1}, \dots, y_k, S) = \text{LSTM-L}(\mathbf{f}_t, x_t, \mathbf{h}_{t+1}) \quad (8)$$

Similar to generate summaries from left to right, we assign that when the predicted word is the start label $\langle \text{START} \rangle$, the left side summary is completed. For training phase, we use teacher forcing algorithm (Williams and Zipser, 1995) to minimize the cross-entropy loss:

$$Loss_L = - \sum_{t=0}^{k-1} \log p(y_t^* | y_{t+1}, \dots, y_k) \quad (9)$$

The loss is calculated for the left part, where k is the position of selected entity E .

3.4 LSTM-R

After getting the predicted sequence from LSTM-L, we can utilize another LSTM to predict the rest part of a complete summary. Actually, We don’t need to generate the right-part the same as the left part since it can be processed as a normal decoder. The decoding process can be described as follows:

$$p(y_t | y_0, \dots, y_{t-1}) = \text{LSTM-R}(\mathbf{f}_t, x_t, \mathbf{h}_{t-1}) \quad (10)$$

However, since we already have the output sequence from the left part, it is not necessary to calculate the loss for the left part. Compared to a normal decoder loss function which calculates the loss from time step 0 to T (T denotes the max sequence length), LSTM-R only needs to consider the time step in and after $k + 1$. k denotes the position of the selected starting entity.

$$Loss_R = - \sum_{t=k+1}^T \log p(y_t^* | y_0, \dots, y_{t-1}) \quad (11)$$

In the inference phase, we apply both greedy sampling and beam search methods. Still, we find the beam search methods gain better results than the simple greedy sampling algorithm.

3.5 Novel Entity prediction and Joint learning

In previous entity-based methods, an entity which is unseen in training set will not appear in generated summaries in testing phase. However, our model can predict novel entities since they are extracted in advance. For entity representations in the prediction step, we randomly assign the entity features from another seen entity that is in the same entity type.

For the final loss of our model, we use the multitask loss which combines the losses of both LSTM-L and LSTM-R:

$$Loss = \alpha \sum Loss_L + (1 - \alpha) \sum Loss_R, \quad (12)$$

where α is a hyper-parameter which is assigned to control the degree of importance for each task. The two LSTMs share the same entity features and article representations.

4 Experimental Settings

4.1 Datasets and Preprocessing

To evaluate the effectiveness of our proposed method, we conducted experiments on two widely-used sentence summarization datasets: the English Gigaword corpus (Graff et al., 2003) and the DUC-2004 dataset (Over et al., 2007).

Dataset	Gigaword	DUC-2004
num(document)	4M	500
avg(article)	31.4	35.6
avg(summary)	8.2	10.4
avg-entity(article)	3.4	2.9
avg-entity(summary)	2.1	1.9

Table 1: Dataset Statistics.

Gigaword dataset leverages the first sentences of articles as inputs and the headlines as output summaries. We replicate the pre-processing steps following (Rush et al., 2015). The pre-processed dataset contains 3.8M training sentence-summary pairs and the testing number is 1951 with empty titles removed. Similar to Gigaword, the task 1 of DUC 2004 is a sentence summarization task. The dataset contains 500 documents with on average 35.6 tokens and summaries with 10.4 tokens, where each document is paired with four manual reference summaries. Due to its size, our model is trained on the Gigaword dataset and only tested on DUC-2004. We also show the statistics of named entities in Table 1. The average number of entities is 3.4 in articles of Gigaword and 2.9 in DUC-2004. The entities are extracted by a BERT NER model which is pretrained on CoNLL2003 dataset (Sang and De Meulder, 2003).

4.2 Training Details

For both datasets, we further reduce the input, output and entity vocabularies to 30k and replace the less-frequent word to “UNK”. We set the max article length to 50 and the left and right generation step to 15. The maximum length of the entity sequence is manually set to 3. For the input sequence without extracted entities, we use an explicit [MID] token to replace the guiding entities. For LSTMs, we set the hidden size to 256. We use dropout on all non-linear connections with a dropout rate of 0.2. Training is done via the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We use beam search of size 5 to generate summaries. Our model is implemented by Tensorflow framework¹.

¹Code is available at <https://github.com/thecharm/Abs-LRModel>

4.3 Baselines

For both Gigaword and DUC-2004 datasets, we compare our model with several state-of-the-art methods: **ABS+** (Rush et al., 2015) is a improved version of ABS which utilizes an attentive CNN encoder and NNLM decoder. **Luong-NMT** (Luong et al., 2015) is a two-layer LSTM encoder-decoder. **SEASS** (Zhou et al., 2017) use BiGRU encoders and GRU decoders with selective encoding. **Pointer-Generator** (See et al., 2017) is a combination of pointer network and seq2seq model. We adopt the re-implementation version of (Wang et al., 2019). **Seq2seq-Sel-MTL-ERAM** (Li et al., 2018) incorporates entailment knowledge into abstractive summarization. **Seq2Seq+E2T** (Amplayo et al., 2018) is a neural abstractive summarization model incorporating entity commonsense representations. **Concept-Pointer** (Wang et al., 2019) generates abstractive summaries with high level semantic concepts. **GenParse-Full** (Song et al., 2020a) jointly generates summaries and their syntactic dependency trees. **ControlCopying** (Song et al., 2020b) is a transformer-based framework to generate summaries with control over copying.

4.4 Evaluation Metrics

We evaluate our system using automatic metrics, human evaluation and a novel fine-grained informativeness metric. We discuss results of these evaluation metrics in section 5.

Automatic Metrics. We first evaluate our system using ROUGE-1 (unigram recall), ROUGE-2 (bigram recall) and ROUGE-L (longest common sequence)². Since the English Gigaword testing set³ contains references of different lengths while the DUC-2004 testing set⁴ fixes the summary length to 75 bytes, we use different ROUGE evaluation options following (Wang et al., 2019).

Human Evaluation. We conduct human evaluation to demonstrate that our model can generate more informative and readable summaries. We randomly choose 30 samples from the Gigaword test set and ask 3 evaluators to rate summaries generated by Seq2Seq+Att Model (our implementation), Pointer-Generator (See et al., 2017), ControlCopying (Song et al., 2020b) and our method along with human-annotators. We run the Pointer-Generator model and Seq2Seq+Att models on the Gigaword test set and the results from ControlCopying model are provided by (Song et al., 2020b).

Informativeness Metric. We develop an informativeness metric inspired by the work of (Jiang et al., 2019) in image captioning tasks and we name it as BERT-REO. BERT-REO assesses the quality of generated summaries from three perspectives: Relevance, Extraness and Omission.

Given a candidate summary C and a ground-truth reference G , we extract the context features \mathbf{c} and \mathbf{g} using a pretrained BERT model, respectively. In particular, when a output summary contains multiple sentences, we average the sentence representations as the final features.

The relevance R of candidate summaries and ground-truth references can be computed by the similarity on the euclidean metric:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2} \quad (13)$$

$$R = d(\mathbf{c}, \mathbf{g}) \quad (14)$$

We calculate the extraneous of C by performing an orthogonal projection of \mathbf{c} to \mathbf{g} . The vertical vector \mathbf{c}_\perp represent the irrelevant information.

$$\mathbf{c}_\perp = \mathbf{c} - \frac{\mathbf{c} \cdot \mathbf{g}}{\|\mathbf{g}\|^2} \mathbf{g} \quad (15)$$

$$E = d(\mathbf{c}_\perp, \mathbf{c}) \quad (16)$$

²We use the pyrouge tool (pypi.python.org/pypi/pyrouge/0.1.3).

³The ROUGE evaluation option is, -m -n 2 -s

⁴The ROUGE evaluation option is, -n 2 -m -b 75 -s

Models	Gigaword			DUC-2004		
	RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
ABS+ [†] (Rush et al., 2015)	29.76	11.88	26.96	28.18	8.46	23.81
Luong-NMT [†] (Luong et al., 2015)	33.10	14.45	30.71	28.55	8.79	24.43
SEASS [†] (Zhou et al., 2017)	36.15	17.54	33.63	29.21	9.56	25.51
Pointer-Generator [†] (See et al., 2017)	35.98	15.99	33.33	28.28	10.04	25.69
Seq2seq-Sel-MTL-ERAM [†] (Li et al., 2018)	35.33	17.27	33.19	29.33	10.24	25.24
Seq2Seq+E2T [†] (Amplayo et al., 2018)	36.15	17.54	33.63	-	-	-
Concept-Pointer [†] (Wang et al., 2019)	37.01	17.10	34.87	30.39	10.78	27.53
GenParse-Full [†] (Song et al., 2020a)	36.61	18.85	34.33	-	-	-
ControlCopying [†] * (Song et al., 2020b)	39.08	20.47	36.68	-	-	-
Our model	40.03	17.93	36.74	34.69	14.39	31.07

Table 2: ROUGE F1 scores on the Gigaword and ROUGE recall values on DUC-2004 test set. The results with [†] are taken from the corresponding papers. * indicates that the method leverages external pretrained corpus. The improvements of our model are statistically significant using a two-tailed t-test with $p < 0.01$.

The measurement of omission is similar to the extraneous, while we perform an orthogonal projection of \mathbf{g} to \mathbf{c} .

$$\mathbf{g}_{\perp} = \mathbf{g} - \frac{\mathbf{g} \cdot \mathbf{c}}{\|\mathbf{c}\|^2} \mathbf{c} \quad (17)$$

$$O = d(\mathbf{g}_{\perp}, \mathbf{g}), \quad (18)$$

where \mathbf{g}_{\perp} indicates the missing information of the candidate summary. The informativeness metric scores for each compared method are averaged along with all predicted summaries.

5 Results and Discussion

In this section, we evaluate the proposed model from both quantitative and qualitative perspectives.

5.1 Automatic Summary Evaluation

Results on Gigaword. As shown in Table 2, we reported best performance against all the strong state-of-the-art models in all metrics except for RG-2 on Gigaword. Amongst the compared methods, our model achieves the new state-of-the-art performance in RG-1 (40.03) and RG-L (36.74) scores which indicates that the incorporation of entity information contributes to a better summarization. Especially, our model outperforms the entity-based model Seq2Seq+E2T (Amplayo et al., 2018) significantly in all metrics, which demonstrates the effectiveness of our model to guarantee the named entities to appear in output summaries. Our model gains a RG-2 score of 17.93 which is higher than the Concept-Pointer but lower than the ControlCopying model which leverages the pretrained language models and external corpus. We suggest future work to explore the BERT-version of our LSTM-L and LSTM-R framework.

Results on DUC-2004. We also compare our model with several state-of-the-art methods on DUC-2004 dataset. Our model achieves better performance in all ROUGE scores and outperforms the Concept-Pointer model (Wang et al., 2019) with about 4 points. Since the DUC-2004 dataset is only used for testing, there are a large proportion of novel entities. We think the ROUGE improvements are mostly due to the leveraging of extracted guiding entities.

5.2 Human Evaluation

We perform human evaluation to analyze the informativeness, grammaticality and coherence of our model and the compared methods in Table 3. Table 3 shows that our model ranks significantly higher on informativeness, which indicates that incorporating named entities in summaries can improve the quality of output texts. Meanwhile, with the guidance of entity information, our model can generate summaries with better coherence compared to the methods without using named entities.

5.3 Informativeness Evaluation

We further evaluate our model from relevance, extraneous and omission perspectives. To the best of our knowledge, we are the first to evaluate text summaries in fine-grained semantic vector level. Different from

System	Inf.	Gram.	Coh.
HUMAN	4.58	4.42	4.53
Seq2Seq+Att	3.21	3.08	3.14
Pointer-Generator	3.63	3.98	3.85
GenParse-Full	4.03	4.22	3.83
ControlCopying	4.10	4.11	4.00
Our model	4.32	3.93	4.01

Table 3: Human Evaluation on informativeness (Inf.), grammaticality (Gram.) and coherence (Coh.). The results are statistically significant with p-value < 0.0005. Our model generates summaries with most informativeness.

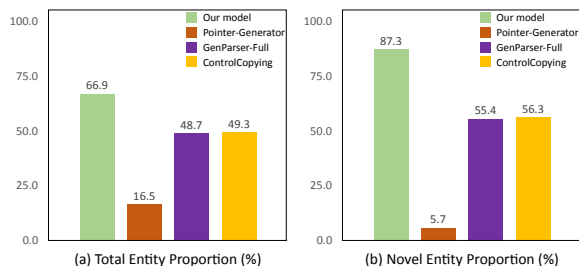


Figure 4: The total entity (a) and novel entity (b) proportion of our model outputs compared to different baselines on Gigaword dataset. Our model can generate summaries with significantly more entities than other methods.

previous lexicon-based metrics such as ROUGE, BLEU et al., semantic vector of generated sequences can reveal the implicit informativeness without taking the grammaticality into consideration.

The results of our model compared to other baselines are shown in Table 4. Our model reports competitive relevance against ControlCopying model with less extra and omissive information. This indicates that entity information may bring the topic coherence and make the generated summaries more faithful to original articles. We also reveal that the previous improvements against traditional seq2seq with attention model are significant in informativeness.

5.4 Novel Entity Prediction

In Figure 4, we show a comparison of our model against Seq2Seq+Att, Pointer-Generator (See et al., 2017) and ControlCopying (Song et al., 2020b) methods based on the proportion of predicted entities to the entities in human-annotated summaries. Compared to other methods which encode entity information and decodes from left to right, our model can generate summaries with significant more entities with the LSTM-L and LSTM-R decoders. Still, our model can output more than one entities with all the selected entities encoded.

According to our statistics, 71.5% percentages of unique entities in Gigaword test set are novel entities. Compared to existing methods, our model has the ability to generate more novel entities which are extracted in advance. We report a higher novel entity proportion against previous summarization models with copy mechanism, which reaffirms our observations on automatic summary metrics that the involved named entities improve the ROUGE scores as well as the summary quality.

5.5 Case Study

Figure 5 shows a case study comparing our model with Pointer-generator (See et al., 2017) and GenParse-Full model (Song et al., 2020a). In the four output summaries, our model captures the most important entities ‘‘Sotomayor’’ and ‘‘U.S.’’ in the human annotated reference. However, the compared methods miss the entity ‘‘U.S.’’. Moreover, our model provides a more informative summary than the simple reference with both syntactic compression and more entities contained.

System	Rel.	Ext.	Omi.
Seq2Seq+Att	11.35	15.29	15.21
Pointer-Generator	10.67	16.15	15.73
GenParse-Full	7.51	16.24	17.13
ControlCopying	7.29	15.94	17.19
Our model	7.30	18.85	17.51

Table 4: The Relevance (Rel.), Extraness (Ext.) and Omission (Omi.) scores between the output sequences of evaluated methods and human annotated summaries on the Gigaword dataset. For the relevance metric, the lower value indicates better performance.

Article: Sonia Sotomayor was sworn in Saturday as the Supreme Court’s first Hispanic justice and only third female member in the top U.S. court’s 220-year history.
Manual Reference: Sotomayor sworn in to top U.S. court.
Pointer-Generator: Sotomayor sworn in as justice.
GenParser-Full: Sotomayor sworn in as Supreme Court justice.
Our model: Sotomayor becomes first Hispanic woman in U.S. Supreme Court.

Figure 5: Output summaries of our model and the compared methods. They are manually re-cased for readability. Texts in color are the extracted named entities.

6 Conclusion

In this paper, we propose an abstractive summarization model which is more informative than previous works with the guiding entities. Our model can ensure that the important entities appear in final output texts with LSTM-L and LSTM-R decoders. Experimental results demonstrate that our model can generate entities with better fluency and coherence. We also develop a fine-grained informativeness metric to evaluate our model compared to previous state-of-the-art methods. Our model achieves better informativeness results than other methods in a comprehensive and diverse manner.

Acknowledgement

This work was supported by the Fundamental Research Funds for the Central Universities, SCUT (No.2017ZD048, D2182480), the National Key Research and Development Program of China, the Science and Technology Programs of Guangzhou (No.201704030076, 201802010027, 201902010046), National Natural Science Foundation of China (62076100) and the Hong Kong Research Grants Council (project no. C1031-18G).

References

- Miltiadis Allamanis, Hao Peng, and Charles Sutton. 2016. A convolutional attention network for extreme summarization of source code. In *International conference on machine learning*, pages 2091–2100.
- Reinald Kim Amplayo, Seonjae Lim, and Seung-won Hwang. 2018. Entity commonsense representation for neural abstractive summarization. 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 697–707.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.
- Mohit Bansal and Rebecca J Passonneau. 2018. Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Tutorial abstracts. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. 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 1662–1675.
- Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686.
- Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.
- 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.
- David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.
- Ming Jiang, Junjie Hu, Qiuyuan Huang, Lei Zhang, Jana Diesner, and Jianfeng Gao. 2019. Relevance, extraneousness, omission: A fine-grained evaluation for image captioning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1475–1480.

- Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817.
- Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441.
- Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4110–4119.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.
- David Newman, Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2006. Analyzing entities and topics in news articles using statistical topic models. In *International conference on intelligence and security informatics*, pages 93–104. Springer.
- Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.
- Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3271–3282.
- Kaiqiang Song, Logan Lebanoff, Qipeng Guo, Xipeng Qiu, Xiangyang Xue, Chen Li, Dong Yu, and Fei Liu. 2020a. Joint parsing and generation for abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Kaiqiang Song, Bingqing Wang, Zhe Feng, Liu Ren, and Fei Liu. 2020b. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181.
- Wenbo Wang, Yang Gao, He-Yan Huang, and Yuxiang Zhou. 2019. Concept pointer network for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3067–3076.
- Ronald J Williams and David Zipser. 1995. Gradient-based learning algorithms for recurrent. *Backpropagation: Theory, architectures, and applications*, 433.
- Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104.
- Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2018. Sequential copying networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.