Entity Highlight Generation as Statistical and Neural Machine Translation

Jizhou Huang*†, Yaming Sun†, Wei Zhang†, Haifeng Wang† and Ting Liu*
*Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
†Baidu Inc., Beijing, China
{huangjizhou01, sunyaming, zhangwei32, wanghaifeng}@baidu.com, tliu@ir.hit.edu.cn

***Abstract*—*Entity highlight* refers to a short, concise, and characteristic description for an entity, which can be applied to various applications. In this article, we study the problem of automatically generating entity highlights from the descriptive sentences of entities. Specifically, we develop two computational approaches, one is inspired by the statistical machine translation (SMT) and another is a sequence-to-sequence learning (Seq2Seq) approach which has been successfully applied in neural machine translation (NMT) and neural summarization. In the Seq2Seq approach, we use attention mechanism, copy mechanism, and coverage mechanism. To generate entity-specific highlights, we also incorporate entity name into the Seq2Seq model to guide the decoding process. We automatically collect large-scale instances as training data without any manual annotation, and ask annotators to create a test set. We compare with several strong baseline methods, and evaluate the approaches with both automatic evaluation and manual evaluation. Experimental results show that the entity enhanced Seq2Seq model with attention, copy, and coverage mechanisms significantly outperforms all other approaches in terms of multiple evaluation metrics.**[1]

***Index Terms*—Entity Highlight Generation, Seq2Seq Model, Attention Mechanism, Copy Mechanism, Coverage Mechanism.**

I. Introduction

Entity highlight refers to a short, concise, and characteristic description for an entity. For example, *"44th U.S. President"* is a highlight for the entity *Barack Obama*[2]. Entity highlight is useful in various applications, such as Web search results enrichment with semantic information [2], entity recommendation in Web search engines [3], [4], and named entity disambiguation [5]. Figure 1 shows an example of Baidu[3] Web search engine's recommendation results for the query *Bill Clinton*. In this example which we translate from Chinese to English for the sake of understanding, an entity highlight is presented under each entity, e.g., *"44th U.S. President"* is an entity highlight of *Barack Obama*. We can see that the entity highlight increases the understandability of the corresponding recommendation by providing users a caption for it, so that users could quickly understand the key facts of the recommended entity [6].

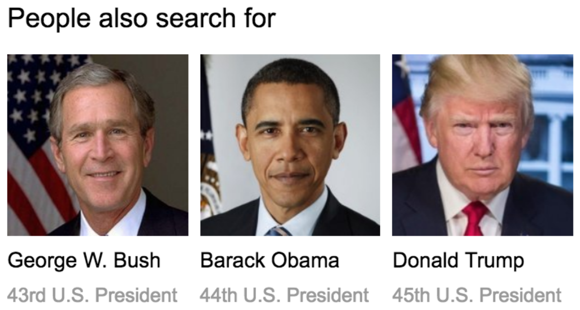


Fig. 1: An example of Baidu Web search engine's entity recommendation results captioned with entity highlights w.r.t each entity.

In this article, we study the task of entity highlight generation, which aims to generate an entity highlight from a descriptive sentence of an entity. Take *Barack Obama* as an example again. Given the sentence *"Barack Hussein Obama II is an American politician who served as the 44th President of the United States from 2009 to 2017."* from the corresponding Wikipedia article, the output entity highlight w.r.t. *Barack Obama* is a natural language expression that consists of three words, namely *"44th U.S. President"*.

The task of entity highlight generation is similar to text summarization, which aims to generate a summary that captures the core meaning of the original text. However, our task differs from text summarization in that the "highlight" in this work should be related to the entity. Entity highlight generation mainly has three challenges. The first challenge is how to effectively select the highlight information from a source sentence for generating a fluent sequence. The words from the source sentence do not have the same importance on describing the characteristic of an entity. The second challenge is how to retain the salient words in the source sentence. Salient words are those that can best capture the salient characteristics w.r.t. a given entity. Therefore, they should be retained when generating an entity highlight. As shown in aforementioned example, salient words can be high-frequency words (e.g., *"President"*) or low-frequency words (e.g., *"44th"*). The third challenge is that a highlight should be entity-related. This requires the model to use the entity to guide the generation process.

[1] This work extends our previous work [1] on generating recommendation evidence for entity recommendation system in Web search engines, in which an SMT based generation model is developed. This work significantly extends our previous work in that a neural sequence-to-sequence approach is developed, which effectively handles the unknown or rare word problem. Moreover, the entity information is not used in the SMT based approach, which is naturally incorporated in the neural network approach to generate entity-specific results.

[2] https://en.wikipedia.org/wiki/Barack_Obama

[3] https://www.baidu.com/

In this article, we develop two computational approaches to study the task of entity highlight generation, one is inspired by the statistical machine translation (SMT) and another is based on sequence-to-sequence learning (Seq2Seq). The use of Seq2Seq learning is motivated by the recent remarkable success of applying Seq2Seq in neural machine translation (NMT) [7]–[12] and text summarization [13]–[17]. Specifically, the Seq2Seq model for entity highlight generation includes an encoder that obtains the hidden states of a source sentence, and a decoder that generates an entity highlight from the hidden states of the encoder. To address the aforementioned challenges of entity highlight generation, we use attention mechanism [8], [18], [19] to selectively focus on the characteristic fragments of the source text, and use copy mechanism [17], [20], [21] to selectively replicate appropriate segments from the source text. We also use coverage mechanism [17] to avoid generating repetitive words. Furthermore, we regard entity name as side information and integrate it into the Seq2Seq model so as to generate entity-specific highlights.

It is widely known that the performance of a Seq2Seq model highly depends on the amount and the quality of training data. As there is no large-scale and publicly available data set, we construct a data set by leveraging an online encyclopedia. We collect about 0.7M instances as the training data without using any manual annotation, and build a test set consisting of 1,000 instances with human labeling. We conduct comprehensive experimental study by comparing with multiple strong baseline methods and doing thoughtful model analysis. We evaluate the experimental results with BLEU, ROUGE, and manual evaluation. Results reveal that the NMT based approach consistently performs better than the SMT based approach. Seq2Seq model with attention, copy, and coverage mechanisms has the ability to generate meaningful entity highlights through learning from massive training data. Based on ablation test and case study, it is shown that both copy mechanism and coverage mechanism can help to improve the performance in terms of multiple evaluation metrics. Furthermore, the incorporation of entity name information can significantly improve the performance of the model.

The remainder of this article is organized as follows. We introduce the related work in Section II. We then present the SMT based approach in Section III, followed by the NMT based approach in Section IV. Experimental setup and result analysis are reported in Section V and Section VI, respectively. Finally, we conclude the article in Section VII.

II. Related Work

Our entity highlight generation task is closely related to the task of text summarization, which aims to condense documents or sentences into a shorter version and preserve important contents. Text summarization approaches can be divided into two groups, namely extractive summarization and abstractive summarization. Extractive summarization extracts salient sentences or phrases from the original text and concatenates them to form a summary [22]–[26]. Abstractive summarization produces a summary consisting of aspects that may not appear as part of the original text [27], [28]. Abstractive summarization pays more attention on text generation, and requires a deeper linguistic (semantic) analysis of texts [29], [30]. In recent years, some studies use Seq2Seq learning on abstractive sentence summarization. Rush *et al.* [13] propose a fully data-driven approach and combine a neural language model with a contextual input encoder for abstractive sentence summarization, and their method shows significant improvements on the DUC-2004 shared task compared with several strong baselines. Nema *et al.* [15] study the query-based abstractive summarization task and propose a diversity driven attention model under the encode-attend-decode paradigm. Compared with text summarization, the entity highlight generation task takes entity as part of input and aims at generating a short and concise description w.r.t. the given entity.

Our work is also related to the task of sentence compression. The methods proposed in [31]–[34] compress an original sentence by deleting words or constituents. However, these methods are restricted to word deletion, and therefore are not readily applicable to the more complex entity highlight generation task.

The use of Seq2Seq learning in this article is motivated by the recent remarkable success of applying Seq2Seq in various NLP tasks. Seq2Seq learning with neural networks is first proposed by Sutskever *et al.* [7] for machine translation. To further improve the performance, multiple mechanisms can be incorporated into the basic Seq2Seq model. During the decoding process, to efficiently use the encoder information, attention mechanism [8] which allows the model to soft search for parts of a source text can be used. The copy mechanism, which uses the information from context based on attention mechanism, can relieve the rare word problem [17], [20], [21], [35]–[37]. The coverage mechanism is proposed to alleviate the over-translation and under-translation problems in NMT models [9], [10] and the repetition problem in neural abstractive summarization [17]. We follow the work [8], [17], [21] and incorporate the attention, copy, and coverage mechanisms into the Seq2Seq model. Different from previous studies that have applied attention, copy, and coverage mechanisms, our model further incorporates the entity name information into the decoding process to help to generate entity-specific results.

III. Entity Highlight Generation as Statistical Machine Translation

Figure 2 shows an overview of the SMT based method. This method contains two components, i.e., sentence preprocessing and entity highlight generation. Sentence preprocessing mainly includes Chinese word segmentation [38], POS tagging [39], and dependency parsing [40] for the input sentences, as POS tags and dependency information are necessary for the following stages. Entity highlight generation is designed to generate entity highlights from the input sentences with a statistical machine translation model. The entity highlight generation model needs three data sources. First, $sen\text{-}eh$ parallel corpus is used to train the "translation" model and language model (described in Section III-A and III-B). Furthermore, headlines of news articles and manually-labeled entity highlights are used to train an attractiveness model (described in Section III-D) to increase the attractiveness of generated entity highlights.

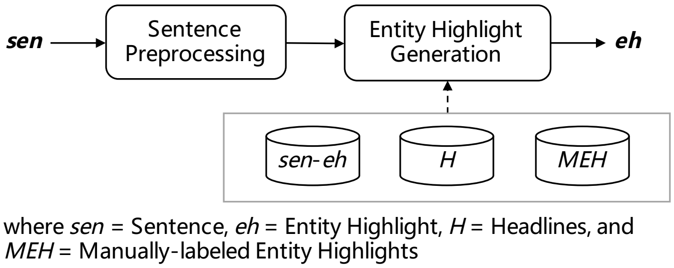

sen
Sentence Preprocessing
Entity Highlight Generation
eh
sen-eh
H
MEH
where sen = Sentence, eh = Entity Highlight, H = Headlines, and MEH = Manually-labeled Entity Highlights

Fig. 2: Overview of the SMT based method.

The entity highlight generation model contains four sub-models: a translation model, a language model, a length model, and an attractiveness model, which control the adequacy, fluency, length, and attractiveness of the entity highlights, respectively.[4]

A. Translation Model (M1)

Entity highlight generation is a decoding process. Similar to [41], the input sentence sen is first segmented into a sequence of units $s\bar{e}n_1^l$, which are then "translated" to a sequence of units $e\bar{h}_1^l$. Let $(s\bar{e}n_i, e\bar{h}_i)$ be a pair of translation units, their translation likelihood is computed using a score function $\phi_{tm}(s\bar{e}n_i, e\bar{h}_i)$. Thus the translation score between sen and eh is decomposed into:

$$p_{tm}(s\bar{e}n_1^l, e\bar{h}_1^l) = \prod_{i=1}^{l} \phi_{tm}(s\bar{e}n_i, e\bar{h}_i)^{\lambda_{tm}}, \tag{1}$$

where λ_{tm} is the weight for the translation model. Actually, it is defined similarly to the translation model in SMT [42].

B. Language Model (M2)

We use a tri-gram language model in this work. The language model based score for the entity highlight eh is computed as:

$$p_{lm}(eh) = \prod_{j=1}^{J} p(w_j|w_{j-2}w_{j-1})^{\lambda_{lm}}, \tag{2}$$

where J is the number of words of eh, w_j is the j-th word of eh, and λ_{lm} is the weight for the language model.

C. Length Model (M3)

We use a length-penalty function to generate short entity highlights whenever possible. The length score for the entity highlight eh is computed as:

$$p_{lf}(eh) = \begin{cases} N & \text{if } N \leq 10 \\ \frac{1}{N-10} & \text{if } N > 10 \end{cases}, \tag{3}$$

where N is the number of Chinese characters of eh.

D. Attractiveness Model (M4)

The attractiveness model is introduced to facilitate the generation of informative and attractive entity highlights, which can better attract users to browse and click the recommended entity. After analyzing a set of manually labeled entity highlights, we found that the attractiveness of eh depends on three aspects: the vocabulary used, the language style, and sentence structure. We use two sub-models to capture these aspects. The first one is a special language model trained on headlines of news articles, which tries to generate catchy and interesting entity highlights with similar vocabularies and styles to headlines. The motivation is that news editors usually try their best to use the attractive expressions to write the headlines. The second one is a sentence structure model trained on human annotated entity highlights, which tries to generate entity highlights with popular syntax styles that users might prefer. Hence the attractiveness model is decomposed into:

$$p_{am}(eh) = p_{hl}(eh)^{\lambda_{hl}} \cdot p_{ss}(eh)^{\lambda_{ss}}, \tag{4}$$

where $p_{hl}(eh)$ is the headline language model and $p_{ss}(eh)$ is the sentence structure model. $p_{hl}(eh)$ is similar to $p_{lm}(eh)$, but trained on headlines. $p_{ss}(eh)$ is computed as:

$$p_{ss}(eh) = \max(K(T_{eh}, T_{t_i})), \tag{5}$$

where T_x is the dependency tree of sentence x, t_i is the human annotated entity highlights, and $K(\cdot,\cdot)$ is the dependency tree kernels described in [43], which measures the structure similarity between sentences.

Finally, we combine the four sub-models based on a log-linear framework and get the entity highlight generation model:

$$\begin{aligned} &p(eh|sen) = \lambda_{tm}\Sigma_{i=1}^{l} \log \phi_{tm}(s\bar{e}n_i, e\bar{h}_i) \\ &+ \lambda_{lm}\Sigma_{j=1}^{J} \log p(w_j|w_{j-2}w_{j-1}) + \lambda_{lf} \log p_{lf}(eh) \\ &+ \lambda_{hl}\Sigma_{l=1}^{L} \log p(w_l|w_{l-2}w_{l-1}) + \lambda_{ss} \log p_{ss}(eh). \end{aligned} \tag{6}$$

IV. Entity Highlight Generation as Neural Machine Translation

We build a Seq2Seq neural network for entity highlight generation. To make better use of the encoder information, we utilize the attention mechanism during decoding. To retain the salient words in the source text, we incorporate the copy mechanism in the decoding process. To alleviate the repetition problem in the decoding process, we also incorporate the coverage mechanism into the Seq2Seq model. Furthermore, we regard entity name as side information and integrate it into the Seq2Seq model so as to generate entity-specific highlights. In the following we respectively introduce the basic Seq2Seq model, attention mechanism, copy mechanism, coverage mechanism, and the entity enhanced Seq2Seq models.

A. The Basic Seq2Seq Model

A basic Seq2Seq model [7] is composed of an encoder and a decoder. The encoder takes the sequence $(x_1, x_2, ..., x_M)$ as input, and obtains a sequence of hidden vectors (h_1, h_2, ..., h_M). The decoder takes h_M as the initial hidden state,

[4] The entity highlight generation model applies monotone decoding, which does not contain a reordering sub-model that is often used in SMT.

and takes a special symbol "GO" as the initial input. In both encoder and decoder, we utilize Long Short-Term Memory (LSTM) [44] as the basic computational unit, which has been proven effective in many tasks [45], [46]. The calculation of LSTM based RNN is briefly described as follows. The approach calculates the hidden vector h_t based on the current word vector d_t and the output vector h_{t-1} in the last time step,

$$f_t = \sigma(W_f \cdot [h_{t-1}, d_t] + b_f), \tag{7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, d_t] + b_i), \tag{8}$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, d_t] + b_C), \tag{9}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t, \tag{10}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, d_t] + b_o), \tag{11}$$

$$h_t = o_t * \tanh(C_t), \tag{12}$$

where f_t, i_t, and o_t are forget, input, and output gates, σ is sigmoid function.

To better encode the input sequence, we also apply a bidirectional RNN [8], which is composed of a forward encoder RNN and a backward encoder RNN to encode the input sequence from both directions.

The training objective is to maximize the conditional probability $p(y_{o1}, y_{o2}, ..., y_{oM'} | x_1, x_2, ..., x_M)$ given the input sequence $(x_1, x_2, ..., x_M)$ and its target output sequence $(y_{o1}, y_{o2}, ..., y_{oM'})$. To improve the efficiency when using a large word vocabulary, we adopt the sample softmax proposed in [47].

B. Attention Mechanism

In the basic Seq2Seq model, the fixed-length vector computed by the encoder acts as the source for the decoder to generate the output sequence. However, when the input sequence is long, it is difficult for the encoder to compress all the necessary information into a fixed-length vector. Moreover, the words from the source sentence do not have the same importance on describing the characteristic of an entity. Therefore, we utilize the attention mechanism [8], [18], [19] to allow the decoder to automatically soft search for parts of the input sequence.

For the RNN decoder, at each time step t, the hidden state s_t is computed as follows:

$$s_t = f(s_{t-1}, y_{t-1}, c_{t-1}), \tag{13}$$

where s_{t-1} is the previous hidden state, y_{t-1} is the current input which can be the predicted word of time step $t-1$ (in the predicting process) or the target word of time step $t-1$ (in the training process), and c_{t-1} is the previous attention context vector. The attention context vector c_t is computed as a weighted sum of the hidden states $(h_1, h_2, ..., h_M)$ obtained by the encoder:

$$c_t = \sum_{j=1}^{M} \partial_{tj} h_j, \tag{14}$$

where weight ∂_{tj} is a score indicating how much attention should be put on the j-th encoder hidden state h_j at each output time step t, and it is computed according to the work [19] as follows:

$$\partial_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{M} \exp(e_{tk})}, \tag{15}$$

$$e_{tj} = v^T \tanh(W h_j + U s_t + b_{attn}), \tag{16}$$

where vector v and matrices W, U are learnable parameters of the model.

C. Copy Mechanism

In the basic Seq2Seq model, each output word is predicted within the scope of a predefined word vocabulary. As a result, rare words cannot be generated by the decoder. However, low-frequency words such as named entities and numbers are important clues to highlight an entity. Therefore, we utilize the copy mechanism [17], [20], [21] to relieve the rare word problem and retain the important information in the source text. At each decoding time step, the attention mechanism assigns source words importance scores, which can be seen as an attention distribution. Thus, the out-of-vocabulary (OOV) words can be replicated according to the attention distribution. We need a switch to determine at each decoding time step, a word should be generated from the predefined word vocabulary or replicated from the source sentence according to the attention distribution. In the following we introduce two types of copy mechanism, namely hard switch copy mechanism and soft switch copy mechanism.

1) Hard Switch Copy Mechanism: The hard switch copy mechanism includes three important neural output layers, namely generate softmax, copy softmax, and gate softmax. The generate softmax produces a vocabulary-size probability distribution to determine which word should be generated. The copy softmax outputs the probability distribution of the words in the source sentence, which can be used to select a source word to copy. The gate softmax generates a binary output, which is used as a hard switch to determine at each decoding time step, a word should be generated from the predefined word vocabulary according to the generate softmax or replicated from the source sentence according to the copy softmax.

Specifically, at each decoding time step t, the generate softmax takes the output vector h_t and attention context c_t as input, and predicts the probability distribution of the predefined vocabulary as follows:

$$P_{generate} = softmax(linear([h_t; c_t])). \tag{17}$$

The copy softmax outputs the probability distribution of the source words according to the attention weights of the hidden states obtained by the encoder, as described in Equation (15).

The gate softmax takes the hidden state s_t as input, and generates a scalar binary output y_{gt} as follows:

$$y_{gt} = \arg\max \ softmax(linear(s_t)). \tag{18}$$

If y_{gt} equals to 1, the output word y_{ot} will be replicated from the source sentence according to the probability distribution of the copy softmax; and if y_{gt} equals to 0, y_{ot}

will be generated from the predefined vocabulary according to $P_{generate}$ produced by the generate softmax.

For an input source sequence $(x_1, x_2, ..., x_M)$ and its target output sequence $(y_{o1}, y_{o2}, ..., y_{oM'})$, the target gate sequence $(y_{g1}, y_{g2}, ..., y_{gM'})$ of the gate softmax is obtained as follows:

$$y_{gt} = \begin{cases} 1 & if\ y_{ot}\ exists\ in\ input\ sequence \\ 0 & otherwise \end{cases} . \quad (19)$$

If y_{gt} equals to 1, the target output of the copy softmax will be the location of the first occurrence of y_{ot} in the source sequence; and if y_{gt} equals to 0, the target output of the generate softmax will be the index of y_{ot} in the predefined vocabulary. The Seq2Seq model with copy mechanism is trained to maximize the conditional probability $p(y_{o1}, ..., y_{oM'}; y_{g1}, ..., y_{gM'} | x_1, ..., x_M)$.

2) Soft Switch Copy Mechanism: In this article, we adopt the soft switch copy mechanism proposed in [17], which also includes three neural output layers, namely generate softmax, copy softmax, and gate. The generate softmax produces the probability distribution of the predefined vocabulary. The copy softmax outputs the probability distribution of the source words. The gate generates a probability $p_{gate} \in [0, 1]$ that is used as a soft switch to combine the probability distribution of the predefined vocabulary and probability distribution of the source words to form a final distribution that is used to predict words.

Specifically, at each decoding time step t, the generate softmax takes the output vector h_t and attention context c_t as input, and predicts the probability distribution of the predefined vocabulary according to Equation (17).

The copy softmax outputs the probability distribution of the source words according to the attention weights of the hidden states obtained by the encoder, as described in Equation (15).

The gate generates the switch probability p_{gate} as follows:

$$p_{gate} = \sigma(w_c^T c_t + w_s^T s_t + w_y^T v_t^y + b), \quad (20)$$

where σ is the sigmoid function, w_c, w_s, w_y and scalar b are learnable parameters, v_t^y is the embedding vector of the input word of decoder, and c_t is the attention context vector.

For each source sentence, an extended vocabulary is defined which is the union of the predefined vocabulary and all words appearing in the source sentence. The prediction probability of each word w in the extended vocabulary is calculated as follows:

$$P(w) = p_{gate} P_{generate}(w) + (1 - p_{gate}) \sum_{j: w_j = w} \partial_{tj}. \quad (21)$$

If the word w is an OOV word, $P_{generate}(w)$ is zero; and if w does not appear in the source sentence, $\sum_{j: w_j = w} \partial_{tj}$ is zero.

D. Coverage Mechanism

The coverage mechanism is originally proposed to address issues of repeating and dropping translations in NMT models [9], [10]. To address the repetition problem in our task, we adopt the coverage mechanism used in [17], which is tailored for neural abstractive summarization.

Specifically, a coverage vector cov_t is maintained, which is the sum of attention distributions over all previous decoding time steps:

$$cov_t = \sum_{t'=0}^{t-1} \partial_{t'}. \quad (22)$$

The coverage vector cov_t records the degree of coverage that each source word receives from the attention mechanism. To make the coverage vector have an effect on the attention mechanism's current decision, the coverage vector is applied into the attention mechanism as an extra input. The Equation (16) in attention mechanism is changed to:

$$e_{tj} = v^T \tanh(W h_j + U s_t + w_{cov} cov_{tj} + b_{attn}), \quad (23)$$

where w_{cov} is a learnable parameter vector of the same length as v.

A coverage loss is defined as follows, which aims to penalize repeatedly attending to the same locations in the source sentence:

$$covloss_t = \sum_{j=1}^{M} \min(\partial_{tj}, cov_{tj}). \quad (24)$$

Finally, the coverage loss is reweighted by a hyperparameter λ, and added to the primary loss function.[5]

E. Entity Enhanced Seq2Seq Model

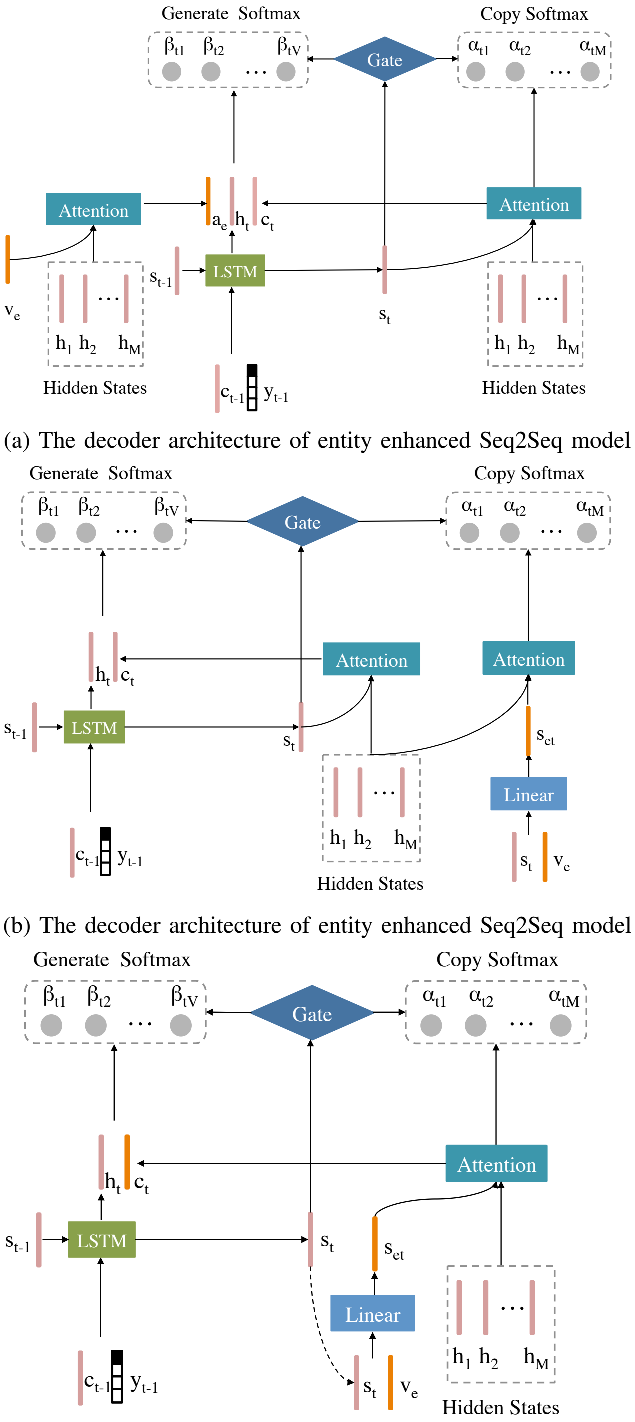
For our human beings, given a source text and the corresponding entity name, to compose an appropriate entity highlight, we generally select the salient words not only by consulting the meaning of the source text, but also by referring to the entity that the source text describes. Therefore, we use the entity name as side information to guide the decoding process to generate better entity highlights.

To represent the semantic information of entity name, we encode it into a vector v_e by applying an RNN on the words in it.[6] We try three different strategies to make use of v_e in the decoder that incorporates both attention mechanism and copy mechanism, namely applying v_e in the generation module, applying v_e in the copy module, and applying v_e in both the generation module and copy module. We implement three models named entity enhanced Seq2Seq model 1, model 2, and model 3 w.r.t. the preceding three strategies.

1) Entity Enhanced Seq2Seq Model 1: Figure 3a illustrates the decoder architecture of entity enhanced Seq2Seq model 1. We use the entity vector v_e to attend to the encoder's hidden states and obtain an entity attention vector a_e through the attention mechanism described in Section IV-B to capture the context information related to the entity. To make the generation process be constrained to the entity, we concatenate the entity attention vector a_e, the current output vector h_t of

[5] Following [17], to obtain the final coverage model, we first train the base model with abundant batches, then we add the coverage mechanism and train the model for a further 3,000 batches. The hyperparamter λ of coverage mechanism is set to 1.

[6] We also tried to encode entity name by averaging the vector representations of the words in it. Experimental results show that RNN is a more effective way to encode entity name in our task.



(a) The decoder architecture of entity enhanced Seq2Seq model 1.

(b) The decoder architecture of entity enhanced Seq2Seq model 2.

(c) The decoder architecture of entity enhanced Seq2Seq model 3.

Fig. 3: Illustrations of three entity enhanced Seq2Seq models.

the RNN cell, and the attention context vector c_t and feed them into the generate softmax.

For both the hard switch copy mechanism and the soft switch copy mechanism, the generate softmax produces the probability distribution of the predefined vocabulary as follows:

$$P_{generate} = softmax(linear([a_e; h_t; c_t])). \tag{25}$$

The copy softmax outputs the probability of each source word according to Equation (15).

2) Entity Enhanced Seq2Seq Model 2: Figure 3b illustrates the decoder architecture of entity enhanced Seq2Seq model 2. In this model, we use the entity vector v_e to guide the copy mechanism to replicate the source words that are more related to the entity. To this end, we apply two different attention layers in this model. In the first attention layer, we use the hidden state s_t to attend to the hidden states of the encoder to obtain the attention context vector c_t, which is used as part of the input of the generate softmax to produce the probability distribution of the predefined vocabulary as described in Equation (17). In the second attention layer, we concatenate the entity vector v_e with hidden state s_t, and obtain a new vector s_{et} through a linear layer:

$$s_{et} = linear([v_e; s_t]). \tag{26}$$

Then we use s_{et} to attend to the hidden states of encoder, and obtain the attention weights of source words according to Equation (15), which are used as the output probability distribution of the copy softmax.

3) Entity Enhanced Seq2Seq Model 3: Figure 3c illustrates the decoder architecture of entity enhanced Seq2Seq model 3. In this model, we use the entity vector v_e to guide both the generation module and the copy module, so that the model can generate or replicate the words that are more related to the entity. To this end, we concatenate the entity vector v_e and the current hidden state s_t, and obtain a new state s_{et} according to Equation (26).

We use the new state s_{et} to attend to the encoder's hidden states, and obtain the context attention vector and attention weights of source words. The obtained context attention vector c_t is used as part of the input of generate softmax as described in Equation (17). The obtained attention weights of source words are treated as the output probability distribution of copy softmax according to Equation (15).

The copy mechanism applied in the preceding three entity enhanced Seq2Seq models can be the hard switch copy mechanism or the soft switch copy mechanism. If the applied copy mechanism is the hard switch copy mechanism, the gate outputs a binary scalar output according to Equation (18), and at each decoding time step, the word is generated according to the vocabulary distribution produced by generate softmax or copied from the source sentence according to the source words distribution produced by copy softmax. If the applied copy mechanism is the soft switch copy mechanism, the gate outputs a probability according to Equation (20), and at each decoding time step, the word is predicted according to the combined probability distribution over the extended vocabulary according to Equation (21). The coverage mechanism can also be applied to the three preceding entity enhanced Seq2Seq models, as described in Equation (23) and Equation (24).

V. Experimental Setup

This section describes the data sets, compared methods, and evaluation metrics in our experiments.

A. Experimental Data

Since there is no publicly available data set, we build the training data and test set based on the online encyclopedia Baidu Baike.[7] Manually labeling instances is expensive and limited in quantity, so we collect the training data automatically as follows. First, given an entity e and its Baike article, the infobox tags are extracted as candidate entity highlights. Second, for e, we rank all the sentences in its abstract text according to the TF*IDF scores of them, and retain top 10 sentences. Third, for each retained sentence sen_i, we compute its relevance score with each candidate entity highlight eh_j by computing the word overlap rate $\frac{l(eh_j,sen_i)}{w(eh_j)}$, where $w(eh_j)$ is the number of words in eh_j, and $l(eh_j, sen_i)$ is the number of overlapping words between eh_j and sen_i. Finally, we rank all the candidate entity highlights according to their relevance scores with sen_i, if the top ranked eh_k has a relevance score with sen_i larger than the threshold 0.5, we treat the triple e-sen_i-eh_k as a training instance. Figure 4 shows an example of training instances. The input includes an entity name and one source sentence that describes the entity, the output is an entity highlight.

Entity name: Barack Obama
Source sentence: Barack Hussein Obama II is an American politician who served as the 44th President of the United States from 2009 to 2017.
Entity highlight: 44th U.S. President

Fig. 4: An example of training instances.

We take entities that belong to "People", "Animal", and "Plant" categories as a case study, and collect 799,080 instances as training set (denoted by $\mathcal{D}$). The average number of entity highlights per entity is 1.5 in the training set. The average numbers of words of entity highlights and source sentences are 3.7 and 25.2, respectively. A preliminary evaluation based on 100 randomly selected instances from $\mathcal{D}$ shows that the accuracy of alignment is 0.91, which is promising for our task. In the same way, we automatically build a development set $\mathcal{D}_v$ consisting of 1,000 instances that are not included in $\mathcal{D}$.

To train the attractiveness model described in Section III-D, we need data of headlines and human annotated entity highlights. We first extracted all headlines from three major Chinese news Websites[8], then we ranked the headlines by the click count in the query logs, and finally top ranked 10 million headlines were remained. To guide the model to generate entity highlights with similar structures to human composed entity highlights, we need a set of human annotated entity highlights of high-quality. We used crowdsourcing method [48] to collect this set. We asked annotators to compose entity highlights for each sentence, then asked 5 different annotators to vote each entity highlight with two options: acceptable or not, finally an entity highlight voted by more than 4 annotators out of 5 as acceptable were kept. A total of 104,775 excellent entity highlights were obtained.

To construct the test set $\mathcal{T}$, we randomly sample 1,000 entities that are not included in the training set and development set from the three categories mentioned above. For each entity e, we first rank all sentences in its Baike article according to their TF*IDF scores, and retain top 10 ranked sentences. Then, we ask an annotator to select one sentence that is considered as the best one to describe entity e from the 10 sentences. After that, we ask other two annotators to compose three entity highlights for e separately. Each entity highlight eh is composed based on two criteria: 1) eh should be a short, concise, and characteristic description of e; and 2) eh can be composed in one of two ways: by extraction or by abstraction. The above criteria are used to ensure that the reference entity highlights have diverse styles with high quality. Third, we invite other three judges to assess the quality of the entity highlights for each entity as per a three-level graded quality scale: perfect, good, and bad. Finally, the test set $\mathcal{T}$ is obtained, which contains 1,000 instances of e-sen-$ehlist$. In our experiments, for each sentence sen, the top-ranked 3 reference entity highlights ($ehlist$) are used for evaluation. The main reason is that, multiple entity highlights could be held by a sentence, therefore, the more reference entity highlights we have for a test sentence, the more reasonable the evaluation result is.

B. SMT Model

Our SMT based approach is abbreviated as **SMT**, which uses a statistical machine translation model to generate entity highlights from the input sentences by removing unimportant words and replacing certain phrases with other more concise and attractive phrases. The details of our SMT model are described in Section III.

C. Seq2Seq Models

Variations of Seq2Seq models with different combinations of attention mechanism, copy mechanism (including hard switch copy mechanism and soft switch copy mechanism), and coverage mechanism are implemented for comparison. All variations of Seq2Seq models are listed as follows.

- **S2S** This model is a basic Seq2Seq model, and utilizes the unidirectional RNN as encoder.
- **S2S+Att** This model utilizes the unidirectional RNN as encoder, and incorporates attention mechanism (denoted by Att) in the decoding process.
- **S2S+Att+HCopy** This model utilizes unidirectional RNN as encoder, and incorporates both attention and copy mechanisms when decoding a sequence. Specifically, the hard switch copy mechanism (denoted by HCopy) is applied in this model.
- **BiS2S+Att+HCopy** This model utilizes bidirectional RNN as encoder, and incorporates attention mechanism and hard switch copy mechanism when decoding a sequence.

[7] In this article, we take the Chinese language as an example and construct training data and test data based on Baidu Baike (https://baike.baidu.com/), which is the largest Chinese encyclopedia in the world. For more languages, the online encyclopedia Wikipedia could be used for the construction of training data and test data.

[8] (1) news.qq.com, (2) news.sina.com.cn, and (3) news.sohu.com

- **BiS2S+Att+SCopy** This model utilizes bidirectional RNN as encoder, and incorporates both attention and copy mechanisms in the decoding process. Specifically, the soft switch copy mechanism (denoted by SCopy) is applied in this model. This model is equivalent to the "pointer-generator" model proposed in [17].
- **BiS2S+Att+SCopy+Cov** This model further incorporates coverage mechanism into BiS2S+Att+SCopy. It is equivalent to the "pointer-generator+coverage" model proposed in [17].

D. Entity Enhanced Seq2Seq Models

Variations of entity enhanced Seq2Seq models with different combinations of attention mechanism, copy mechanism (including hard switch copy mechanism and soft switch copy mechanism), and three different entity enhanced strategies (denoted by E1, E2, and E3) as described in Section IV-E are implemented for comparison. All variations of entity enhanced Seq2Seq models are listed as follows.

- **BiS2S+Att+HCopy+E1** This model is an entity enhanced Seq2Seq model 1 based on BiS2S+Att+HCopy.
- **BiS2S+Att+HCopy+E2** This model is an entity enhanced Seq2Seq model 2 based on BiS2S+Att+HCopy.
- **BiS2S+Att+HCopy+E3** This model is an entity enhanced Seq2Seq model 3 based on BiS2S+Att+HCopy.
- **BiS2S+Att+SCopy+E1** This model is an entity enhanced Seq2Seq model 1 based on BiS2S+Att+SCopy.
- **BiS2S+Att+SCopy+E2** This model is an entity enhanced Seq2Seq model 2 based on BiS2S+Att+SCopy.
- **BiS2S+Att+SCopy+E3** This model is an entity enhanced Seq2Seq model 3 based on BiS2S+Att+SCopy.
- **BiS2S+Att+SCopy+Cov+E1** This model is an entity enhanced Seq2Seq model 1 based on the coverage model BiS2S+Att+SCopy+Cov.
- **BiS2S+Att+SCopy+Cov+E2** This model is an entity enhanced Seq2Seq model 2 based on the coverage model BiS2S+Att+SCopy+Cov.
- **BiS2S+Att+SCopy+Cov+E3** This model is an entity enhanced Seq2Seq model 3 based on the coverage model BiS2S+Att+SCopy+Cov.

E. Other Methods

Entity highlight generation can also be modeled as abstractive sentence summarization or sequence labeling. The following two strong methods are selected for comparison.

- **AS** This method is proposed by Rush *et al.* [13], which uses a neural attention model to generate a shorter version of a given sentence while attempting to preserve its meaning for the task of abstractive sentence summarization.
- **LSTM-CRF** This method is the LSTM-CRF model, which is an effective model for sequence tagging [46], [49]. The LSTM-CRF model encodes the source sentence with bidirectional LSTMs, and produces the tag sequence with a CRF layer. This method can be seen as a kind of extractive summarization method for entity highlight generation. To construct the training data for the LSTM-CRF model, we first identify the text span in sen which has the maximum overlap with eh for each (sen, eh) pair, then we treat the identified span as the chunk to detect and use the BIO (Beginning, Inside and Outside of a chunk) tagging scheme [50] to label each word.

F. Evaluation Metrics

We evaluate the methods with both automatic and manual evaluation metrics. For automatic evaluation, BLEU[9] [51] and ROUGE [52] are employed, which have been widely used for automatic evaluation in machine translation (MT) and text summarization. BLEU measures the similarity between a translation and the human references. ROUGE measures the similarity between a generated summary and the ideal summaries created by humans. The manual evaluation is similar to the human evaluation for MT [53]. We evaluate an entity highlight eh generated by a model manually based on fluency and usability, each of which has three scales including bad, good, and perfect. Here is a brief description of the manual evaluation criteria.

Fluency	Bad: eh is incomprehensible.
	Good: eh is comprehensible.
	Perfect: eh is a flawless expression.
Usability	Bad: eh is not a description of e.
	Good: eh is a description of e.
	Perfect: eh is a characteristic description of e.

For manual evaluation, we ask two raters to evaluate the entity highlights generated by each model over the entire test set $\mathcal{T}$ based on the above criteria. The final manual evaluation results are obtained by averaging the results of the two raters. To assess the agreement between the two raters, we compute the kappa [54] statistic between their evaluation results. Kappa is defined as $K = \frac{P(A)-P(E)}{1-P(E)}$, where $P(A)$ is the proportion of times that the labels agree, and $P(E)$ is the proportion of times that they may agree by chance. We define $P(E)$=1/3, as the labeling is based on three point scales. The kappa statistics for fluency and usability are 0.6727 and 0.7958, respectively, which indicates a substantial agreement (K: 0.61-0.8) according to [55].

G. Parameter Settings

Given the training set $\mathcal{D}$ and the development set $\mathcal{D}_v$, $\mathcal{D}$ was used to train all the models, and $\mathcal{D}_v$ was used to tune the parameters for all the models. All models with the parameter settings that best performed on $\mathcal{D}_v$ were selected for use. For SMT model, the parameters $\lambda_{tm}, \lambda_{lm}, \lambda_{lf}, \lambda_{hl}$, and λ_{ss} were estimated by adopting the approach of minimum error rate training (MERT) that is popular in SMT [56]. For Seq2Seq models, all parameters were updated using stochastic gradient descent during training. The hyper-parameters of the Seq2Seq models are as follows: word vocabulary size is 200,000, the number of hidden units is 256, batch size is 128, and dimension of word embedding is 128.

[9] We use multi-bleu.perl of the open source toolkit MOSES downloaded from https://github.com/moses-smt/mosesdecoder with the default parameters to compute the BLEU scores.

TABLE I: Performance of all methods. RG refers to ROUGE. Boldface indicates the best score w.r.t. each metric.

Method	Automatic Evaluation				Manual Evaluation					
	BLEU	ROUGE			Fluency (%)			Usability (%)		
		RG-1	RG-2	RG-L	Bad	Good	Perfect	Bad	Good	Perfect
AS (Rush et al. [13])	6.80	11.73	3.44	10.98	57.30	25.20	17.50	90.10	9.10	0.80
LSTM-CRF (Ma et al. [46])	2.49	6.30	2.89	6.26	62.10	19.55	18.35	78.05	19.10	2.85
SMT (Huang et al. [1])	17.02	9.97	5.21	9.79	73.20	15.10	11.70	83.85	13.55	2.60
S2S	1.63	6.53	0.73	6.33	79.20	10.95	9.85	93.45	6.10	0.45
S2S+Att	12.31	10.95	4.20	10.75	63.60	16.85	19.55	82.35	15.35	2.30
S2S+Att+HCopy	24.46	17.16	8.41	16.69	44.00	24.20	31.80	63.55	30.60	5.85
BiS2S+Att+HCopy	25.73	16.56	8.88	16.22	39.15	25.15	35.70	58.05	35.00	6.95
BiS2S+Att+SCopy (See et al. [17])	31.79	16.48	10.01	16.36	37.10	24.65	38.25	55.10	36.65	8.25
BiS2S+Att+SCopy+Cov (See et al. [17])	34.43	17.82	10.60	17.64	33.60	25.95	40.45	52.95	37.90	9.15
BiS2S+Att+HCopy+E1	26.73	16.49	9.07	16.25	38.15	26.35	35.50	57.35	35.15	7.50
BiS2S+Att+HCopy+E2	25.11	16.62	8.42	16.21	38.65	25.95	35.40	57.90	35.25	6.85
BiS2S+Att+HCopy+E3	28.50	17.21	9.15	16.88	37.80	26.20	36.00	57.10	35.35	7.55
BiS2S+Att+SCopy+E1	32.01	17.02	10.06	16.84	33.50	26.65	39.85	52.60	**39.40**	8.00
BiS2S+Att+SCopy+E2	31.51	17.43	10.23	17.19	34.10	25.10	40.80	53.00	38.15	8.85
BiS2S+Att+SCopy+E3	32.70	17.50	10.43	17.28	33.15	25.95	40.90	52.10	38.90	9.00
BiS2S+Att+SCopy+Cov+E1	33.02	17.20	10.62	17.00	29.55	29.25	41.20	52.10	**39.40**	8.50
BiS2S+Att+SCopy+Cov+E2	32.43	17.32	10.37	17.09	34.05	26.30	39.65	53.00	37.85	9.15
BiS2S+Att+SCopy+Cov+E3	**36.28**	**18.34**	**11.34**	**18.12**	**28.75**	**29.85**	**41.40**	**50.25**	38.95	**10.80**

VI. Results and Analysis

We report empirical results, model comparisons, and model analysis in this section.

A. Model Comparisons

In this subsection, we compare the performance of different methods including AS, LSTM-CRF, SMT, and Seq2Seq models including entity enhanced models. Evaluation results are shown in Table I.

AS achieves higher BLEU and ROUGE scores than LSTM-CRF. The entity highlights tagged by LSTM-CRF contain a lot of empty results. Therefore, LSTM-CRF is not suitable for our task, even though it is an effective method in sequential labeling tasks such as named entity recognition. SMT outperforms AS and LSTM-CRF in terms of BLEU. Compared with AS, LSTM-CRF, and SMT, all other Seq2Seq models except S2S and S2S+Att achieve higher performance in terms of multiple evaluation metrics. Among the Seq2Seq models, BiS2S+Att+SCopy+Cov achieves the highest BLEU and ROUGE scores. Among the entity enhanced Seq2Seq models, BiS2S+Att+SCopy+Cov+E3 achieves the highest BLEU and ROUGE scores. The entity enhanced model BiS2S+Att+SCopy+Cov+E3 also significantly outperforms all other models by a large margin in terms of multiple evaluation metrics. The results verify the effectiveness of applying Seq2Seq models in our task.

Compared with AS and SMT, Seq2Seq models including entity enhanced models that incorporate the attention mechanism and copy mechanism have the advantage that they can replicate OOV words from the source text. LSTM-CRF has a limitation that it cannot generate the word that does not occur in the source text. The manual evaluation results show that the entity highlights generated by SMT have poor quality especially in terms of usability. The main reason is that, SMT is mainly based on word alignment, therefore, it cannot effectively utilize the global information of the source text. By contrast, Seq2Seq models can attend to all the source information using the attention mechanism during the decoding process. In addition, SMT cannot handle the OOV words in the source text.

B. Model Analysis

In this subsection, we give the detailed analysis of different variations of the Seq2Seq models including entity enhanced models. Table I shows the results of each Seq2Seq model.

First, we investigate whether attention mechanism and copy mechanism are helpful to our task. We take the hard switch copy mechanism as an example, and compare models S2S, S2S+Att, and S2S+Att+HCopy. S2S+Att achieves significant performance gain over S2S, which demonstrates that attention mechanism can help to generate better entity highlights by selectively focusing on characteristic fragments of the source text during decoding. S2S+Att+HCopy significantly outperforms S2S+Att, which demonstrates the effectiveness of copy mechanism.

Second, we compare the performance of unidirectional RNN encoder and bidirectional RNN encoder. BiS2S+Att+HCopy outperforms S2S+Att+HCopy in terms of BLEU and manual evaluation metrics, which indicates the superiority of using bidirectional RNN encoder over unidirectional RNN encoder in the Seq2Seq model.

Third, we evaluate the performance of hard switch copy mechanism and soft switch copy mechanism by comparing BiS2S+Att+SCopy and BiS2S+Att+HCopy. The results show that soft switch copy mechanism is more effective than hard switch copy mechanism in our task.

Fourth, we investigate whether coverage mechanism is helpful to our task by comparing the two models BiS2S+Att+SCopy and BiS2S+Att+SCopy+Cov. The results show that BiS2S+Att+SCopy+Cov significantly outperforms BiS2S+Att+SCopy, which demonstrates the effectiveness of coverage mechanism.

Finally, we evaluate whether the entity name information can facilitate the generation of higher-quality entity highlights.

Compared with their corresponding base models[10], there are no significant gains achieved by entity enhanced Seq2Seq model 1 and model 2. This shows that the ways used to incorporate entity name in entity enhanced Seq2Seq model 1 and model 2 cannot effectively contribute to the entity highlight generation. By contrast, BiS2S+Att+SCopy+Cov+E3 significantly outperforms all other models in terms of multiple evaluation metrics, which indicates the effectiveness of entity enhanced Seq2Seq model 3. The improvement achieved by entity enhanced Seq2Seq model 3 demonstrates that utilizing the entity name information in both the word generation module and the copy module can significantly contribute to each decoding step and can help to generate better entity highlights w.r.t. an entity.

C. Case Study

In this subsection, we conduct a case study to analyze the advantages and disadvantages of the Seq2Seq models. First, we analyze the ability to produce salient words of different models. All words except stop words[11] in the reference entity highlights are considered as salient words.

The $recall$ of salient words is computed as follows:

$$recall = \frac{1}{\mid \mathcal{T} \mid}\sum_{t\in|\mathcal{T}|}\frac{1}{3}\sum_{i\in\{1,2,3\}}\frac{cw_t^i}{vw_t^i}, \qquad (27)$$

where $\mathcal{T}$ is the test set, t is the index of a test instance, vw_t^i is the total number of salient words in the i-th reference of the t-th test instance, cw_t^i is the number of overlapped salient words between a generated entity highlight and the i-th reference of the t-th test instance.

To measure the ability of different models to produce OOV words, we compute $recall_{oov}$ that only considers OOV words as follows:

$$recall_{oov} = \frac{1}{\mid \mathcal{T}_o \mid}\sum_{t\in|\mathcal{T}_o|}\frac{1}{3}\sum_{i\in\{1,2,3\}}\frac{co_t^i}{vo_t^i}, \qquad (28)$$

where $\mathcal{T}_o$ is the subset of test set $\mathcal{T}$, and each instance in $\mathcal{T}_o$ has one or more references containing OOV words. co_t^i is the number of overlapped OOV words between a generated entity highlight and the i-th reference of the t-th test instance, and vo_t^i is the number of OOV words in the i-th reference of the t-th test instance.

Table II shows the $recall$ and $recall_{oov}$ of different models. Among the compared methods, AS, SMT, S2S, and S2S+Att cannot produce OOV words, so we have not computed $recall_{oov}$ for them. The LSTM-CRF model generates tags for a source sentence, so that any words in the source sentence might be retained. Therefore, we do not compute $recall_{oov}$ for LSTM-CRF too. From the results, we make the following observations: 1) all Seq2Seq models that incorporate attention, copy, and coverage mechanisms produce more salient words than AS, LSTM-CRF, SMT, S2S, and S2S+Att, which shows that these mechanisms can help the Seq2Seq models to better identify the characteristics of the entities from the source sentences; 2) all Seq2Seq models that are enhanced by E3 outperform their corresponding base models in terms of $recall$ (i.e., BiS2S+Att+HCopy+E3 vs. BiS2S+Att+HCopy, BiS2S+Att+SCopy+E3 vs. BiS2S+Att+SCopy, and BiS2S+Att+SCopy+Cov+E3 vs. BiS2S+Att+SCopy+Cov), which verifies that incorporating entity name information with E3 can also help the Seq2Seq models to produce more salient words; 3) BiS2S+Att+SCopy+Cov+E3 achieves the highest $recall$ score, which demonstrates that the proposed model can generate better entity highlights with more salient words; and 4) empirically hard switch copy mechanism performs better than soft switch copy mechanism in terms of $recall_{oov}$.

TABLE II: Recall of salient words of different models.

Method	$recall$	$recall_{oov}$
AS (Rush et al. [13])	12.25%	-
LSTM-CRF (Ma et al. [46])	6.15%	-
SMT (Huang et al. [1])	10.44%	-
S2S	5.87%	-
S2S+Att	10.78%	-
S2S+Att+HCopy	18.46%	5.04%
BiS2S+Att+HCopy	17.71%	**8.32**%
BiS2S+Att+SCopy (See et al. [17])	17.23%	5.24%
BiS2S+Att+SCopy+Cov (See et al. [17])	18.43%	6.59%
BiS2S+Att+HCopy+E1	17.71%	6.87%
BiS2S+Att+HCopy+E2	17.38%	8.04%
BiS2S+Att+HCopy+E3	18.51%	6.94%
BiS2S+Att+SCopy+E1	17.64%	4.74%
BiS2S+Att+SCopy+E2	18.16%	5.69%
BiS2S+Att+SCopy+E3	18.26%	5.07%
BiS2S+Att+SCopy+Cov+E1	17.49%	6.43%
BiS2S+Att+SCopy+Cov+E2	18.07%	6.12%
BiS2S+Att+SCopy+Cov+E3	**18.98**%	5.86%

We show several representative examples of the compared models, and give three examples in Figure 5. In example a, the word "*45*" is a rare word and represented with "*00*" in the word vocabulary of our Seq2Seq models. The models AS, SMT, S2S, and S2S+Att all fail to generate the salient word "*45*", due to the lack of the ability to handle OOV words. All other Seq2Seq models except S2S and S2S+Att produce "*45*" successfully. Although LSTM-CRF does not have the problem of OOV word generation, it cannot produce words that do not appear in the source sentence as shown in example b. It is also shown in example b that both BiS2S+Att+SCopy+Cov and BiS2S+Att+SCopy+Cov+E3 generate better entity highlights in comparison with other models. Their generated entity highlights contain the words that are unseen in the source sentence, which make the entity highlights more fluent and diverse in style. Example c shows that entity enhanced Seq2Seq model 3 can generate better entity highlights in comparison with its base model, which demonstrates the effectiveness of the incorporation of entity name information into the Seq2Seq model.

To analyze the disadvantages of our Seq2Seq models, we take the best performing model BiS2S+Att+SCopy+Cov+E3 as an example. Specifically, we analyze the cases with bad scores in terms of fluency and usability, and find two kinds of typical errors: 1) 22.26% of the bad cases contain irrelevant

[10] For example, BiS2S+Att+SCopy is the base model of BiS2S+Att+SCopy+E1, BiS2S+Att+SCopy+E2, and BiS2S+Att+SCopy+E3.

[11] The stop word list contains 1,215 Chinese words including punctuations and words that appear frequently in documents while carry no significant information.

Example a - Entity name: 唐纳德·特朗普 (Donald Trump)

Source sentence: 2016 年 11 月 9 日 , 美国大选 计 票 结果显示 : 共和党 候选人 唐纳德 · 特朗普 已 获得 了 276 张 选举人 票 , 超过 270 张 选举人 票 的 获胜 标准 , 当选 美国 第 45 任 总统 (In November 9, 2016, the U.S. presidential election results showed: Republican candidate Donald Trump won 276 electoral votes, more than the 270 electoral votes required to win, was elected the 45th U.S. President.)

Golden: 美国 第 <u>45</u> 任 总统 (45th U.S. President)

AS: 第 2 任 总统 候选人(candidate for the second President)

LSTM-CRF: 美国 第 <u>45</u> 任 总统 (45th U.S. President)

SMT: 曾 任 总统 (former President)

S2S: 美国 第 00 任 总统 (00th U.S. President)

S2S+Att: 第 0 任 美国 第 00 届 总统 ♮ (0th 00th U.S. President)

Other Seq2Seq models: 美国 第 <u>45</u> 任 总统 (45th U.S. President)

Example b - Entity name: 界王 (Kaiō)

Source sentence: 日本 著名 漫画 《七龙珠》 登场 角色 ， 是 负责管理 银河 的 神 ， 一共 有 五 个 ， 分别 是 东南西北 四 个 界王 和 大 界王 。 在 神界 地位 在 阎王 之 上 ， 仅次于 界王 神 。 (Kaiō (Lord of the Worlds) are the debut character in the famous Japanese manga “Dragon Ball", who are the gods in charge of galaxy. There are five Kaiō, namely East Kaiō, West Kaiō, South Kaiō, North Kaiō, and Dai Kaiō. The level of Kaiō in the deities world is higher than the King of Hell, and merely lower than Kaiō-shin (Lord of Lords).)

Golden: 日本 著名 漫画 《七龙珠》 登场 角色 (the debut character of the famous Japanese manga “Dragon Ball")

AS: 登场 作品 漫画 中 人物 ♮ (debut works: the character in manga)

LSTM-CRF: 《七龙珠》 (“Dragon Ball")

SMT: 漫画 《七龙珠》 登场 角色 (debut character of the manga “Dragon Ball")

S2S: 登场 作品 《死神》 (debut works: “Azrael")

BiS2S+Att+SCopy+Cov: 登场 作品 《七龙珠》 (debut works: “Dragon Ball")

BiS2S+Att+SCopy+Cov+E3: 登场 作品 《七龙珠》 (debut works: “Dragon Ball")

Example c - Entity name: 孙妍在 (Sun Yanzai)

Source sentence: 2016 年 8 月 21 日 凌晨 ， 在 巴西 里约 奥林 匹克 体育场 举行 的 艺术体操 个人 全能 决赛 中 ， 韩国 艺体 精灵 孙妍在 以 总分 72.898 分 排名 第 4 ， 无缘 奖牌 (In the early morning of August 21, 2016, in the artistic individual all-around finals held at the Olympic Stadium in Rio de Janeiro, the Korean artistic genie Sun Yanzai was ranked fourth with a total score of 72.898 points, and missed medal.)

Golden: 韩国 艺体 精灵 (Korean artistic genie)

AS: 运动 项目 艺术体操 个人 全能 (sports event: artistic individual all-around)

LSTM-CRF:艺术体操 (eurythmics)

SMT: 艺术体操 个人 全能 第 72.898 ♮ (eurythmics individual all-around rank 72.898)

S2S: 韩国 女子 田径 运动员 (Korean women’s track and field athlete)

BiS2S+Att+SCopy+Cov: 运动 项目 艺术体操 (sports event: eurythmics)

BiS2S+Att+SCopy+Cov+E3: 韩国 艺体 精灵 (Korean artistic genie)

Fig. 5: Examples of generated entity highlights. Word segmentation is applied, where OOV words are *underlined*, words in *red rectangles* are unseen in the source sentence, and each salient word is marked with a distinctive color. The ones with errors in language are indicated by ♮.

***Example d* - Entity name:** 死神 (Azrael)

Source sentence: ” 死神 ” 能 用 地狱火 霰弹枪 造成 巨大 伤害 ， 能够 用 幽灵 形态 躲避 伤害 ， 还能 用 暗影步 在 各 个 地点 来 回 穿梭 ， 这些 能力 让 他 足以 成为 最 致命 的 杀手 (“Azrael” can cause great damage to the shotgun by Hellfire shotgun. It can use ghost form to avoid injury, and can also shuttle back and forth in every place with shadow step, which makes him the most deadly killer.)

Golden: 能够 用 幽灵 形态 躲避 伤害 的 最 致命杀手 (the most deadly killer who can use ghost form to avoid injury)

BiS2S+Att+SCopy+Cov+E3: 代表作品 躲避 伤害 (Representative work: avoid injury)

***Example e* - Entity name:** 迪克西·迪恩 (Dixie Dean)

Source sentence: 2001 年 ， 利物浦 当地 的 雕塑家 汤姆 墨菲 把 一 尊 迪恩 的 塑像 在 古迪逊 球场 外 立 起 ， 其 下 刻 着 “ 球员 、 绅士 、 埃弗顿 人 ” 以 纪念 这位 俱乐部 历史 上 这位 传奇 射手 (In 2001, Tom Murphy, a local sculptor in Liverpool, set up a statue of Dean outside the ancient didison stadium, engraved with “players, gentlemen, Everton people” to commemorate the legendary shooter in the history of the club.)

Golden: 利物浦 俱乐部 历史 上 的 传奇 射手 (a legend shooter in the history of club in Liverpool)

BiS2S+Att+SCopy+Cov+E3: 俱乐部 雕塑家 ♮ (club sculptor)

Fig. 6: Error analysis for the BiS2S+Att+SCopy+Cov+E3 model. In the generated *entity highlights*, the words irrelevant to the meaning of source sentence are marked in *red*.

words w.r.t. the source text and the given entity; and 2) 27.21% of the bad cases contain words that are unrelated with each other in the source text. Two representative examples of such bad cases are shown in Figure 6, which cannot be handled well by the current decoder. In example d, BiS2S+Att+SCopy+Cov+E3 generates irrelevant words w.r.t. the source text and the given entity. In example e, BiS2S+Att+SCopy+Cov+E3 generates words that are unrelated with each other, and generates a wrong entity highlight which distorts the original meaning of the source text.

VII. Conclusions and Future Work

In this article, we study the problem of automatically generating entity highlights. We develop both SMT based and NMT based approaches, and conduct extensive experiments to verify the effectiveness of the approaches. We collect training data in an automatic way, and evaluate our approach with both automatic and manual evaluation metrics. Experiments show that the NMT based approaches with different combinations of attention, copy, and coverage mechanisms perform better than the SMT based approach. Results also demonstrate that the performance of the Seq2Seq models can be further improved by incorporating entity name information into them.

In the future, we plan to improve this work from two directions. On one hand, the sequence-to-sequence learning based approach could be further improved from both entity and sentence/passage parts, including richer entity semantics such as the entity category in the encyclopedia and richer passage information such as the syntactic structure. On the other hand, in the real scenario when a user issues a query, it would be interesting if the entity highlights of the recommended entities are also related to the query. The entity highlight generation approach proposed in this work does not consider the query information.

ACKNOWLEDGMENT

This research is supported by the National Basic Research Program of China (973 program No. 2014CB340505). The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] J. Huang, S. Zhao, S. Ding, H. Wu, M. Sun, and H. Wang, "Generating recommendation evidence using translation model," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016, pp. 2810–2816.

[2] A. Singhal, "Introducing the knowledge graph: things, not strings," Official Blog of Google: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html, 2012.

[3] X. Yu, H. Ma, B.-J. P. Hsu, and J. Han, "On building entity recommender systems using user click log and freebase knowledge," in *Proceedings of the 7th ACM international conference on Web search and data mining*, 2014, pp. 263–272.

[4] J. Huang, S. Ding, H. Wang, and T. Liu, "Learning to recommend related entities with serendipity for web search users," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 17, no. 3, pp. 25:1–25:22, Apr. 2018.

[5] Y. Li, C. Wang, F. Han, J. Han, J. Han, and X. Yan, "Mining evidences for named entity disambiguation," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1070–1078.

[6] J. Huang, W. Zhang, S. Zhao, S. Ding, and H. Wang, "Learning to explain entity relationships by pairwise ranking with convolutional neural networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2017, pp. 4018–4025.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *arXiv preprint arXiv*, 2014.

[9] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[10] H. Mi, B. Sankaran, Z. Wang, and A. Ittycheriah, "Coverage embedding models for neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016, pp. 955–960.

[11] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, 2017, pp. 67–72.

[12] J. Zhang, M. Wang, Q. Liu, and J. Zhou, "Incorporating word reordering knowledge into attention-based neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 1524–1534.

[13] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 379–389.

[14] S. Chopra, M. Auli, and A. M. Rush, "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*, 2016, pp. 93–98.

[15] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran, "Diversity driven attention model for query-based abstractive summarization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1063–1072.

[16] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective encoding for abstractive sentence summarization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 1095–1104.

[17] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 1073–1083.

[18] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[19] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Advances in Neural Information Processing Systems*, 2015, pp. 2773–2781.

[20] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[21] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 140–149.

[22] R. Mihalcea, "Language independent extractive summarization," in *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, 2005, pp. 49–52.

[23] D. Parveen, H.-M. Ramsl, and M. Strube, "Topical coherence for graph-based extractive summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1949–1954.

[24] K. Woodsend and M. Lapata, "Automatic generation of story highlights," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 565–574.

[25] H. Kobayashi, M. Noguchi, and T. Yatsuka, "Summarization based on embedding distributions," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1984–1989.

[26] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 484–494.

[27] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Computational linguistics*, vol. 28, no. 4, pp. 399–408, 2002.

[28] A. Khan and N. Salim, "A review on abstractive summarization methods," *Journal of Theoretical and Applied Information Technology*, vol. 59, no. 1, pp. 64–72, 2014.

[29] D. Das and A. F. Martins, "A survey on automatic text summarization," *Literature Survey for the Language and Statistics II course at CMU*, vol. 4, pp. 192–195, 2007.

[30] P.-E. Genest and G. Lapalme, "Text generation for abstractive summarization," in *TAC*, 2010.

[31] J. Turner and E. Charniak, "Supervised and unsupervised learning for sentence compression," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 290–297.

[32] M. Galley and K. McKeown, "Lexicalized Markov grammars for sentence compression," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007, pp. 180–187.

[33] T. Nomoto, "Discriminative sentence compression with conditional random fields," *Information processing & management*, vol. 43, no. 6, pp. 1571–1587, 2007.

[34] W. Che, Y. Zhao, H. Guo, Z. Su, and T. Liu, "Sentence compression for aspect-based sentiment analysis," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 12, pp. 2111–2124, 2015.

[35] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *Proceedings of the 53rd Annual Meeting of the Association for*

Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015, pp. 11–19.

[36] J. Gu, Z. Lu, H. Li, and V. O. Li, “Incorporating copying mechanism in sequence-to-sequence learning,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 1631–1640,.

[37] Z. Cao, C. Luo, W. Li, and S. Li, “Joint copying and restricted generation for paraphrase,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 1319–1323.

[38] F. Peng, F. Feng, and A. Mccallum, “Chinese segmentation and new word detection using conditional random fields,” in *Proceedings of the 20th International Conference on Computational Linguistics*, 2004, pp. 562–568.

[39] J. Gimenez and L. Marquez, “SVMTool: A general POS tagger generator based on Support Vector Machines,” in *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004.

[40] R. McDonald, K. Lerman, and F. Pereira, “Multilingual dependency analysis with a two-stage discriminative parser,” in *CoNLL*, 2006, pp. 216–220.

[41] S. Zhao, X. Lan, T. Liu, and S. Li, “Application-driven statistical paraphrase generation,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, 2009, pp. 834–842.

[42] P. Koehn, F. J. Och, and D. Marcu, “Statistical phrase-based translation,” in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2003, pp. 48–54.

[43] A. Culotta and J. Sorensen, “Dependency tree kernels for relation extraction,” in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 2004, pp. 423–429.

[44] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals, “Sentence compression by deletion with LSTMs,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 360–368.

[46] X. Ma and E. H. Hovy, “End-to-end sequence labeling via bi-directional lstm-cnns-crf,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[47] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, “On using very large target vocabulary for neural machine translation,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 1–10.

[48] P.-Y. Hsueh, P. Melville, and V. Sindhwani, “Data quality from crowdsourcing: A study of annotation selection criteria,” in *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, 2009, pp. 27–35.

[49] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 2016, pp. 260–270.

[50] L. A. Ramshaw and M. Marcus, “Text chunking using transformation-based learning,” in *Third Workshop on Very Large Corpora, VLC@ACL 1995, Cambridge, Massachusetts, USA, June 30, 1995*, 1995.

[51] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 311–318.

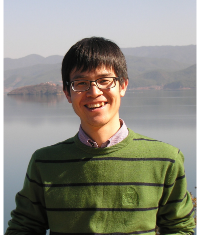
[52] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text summarization branches out: Proceedings of the ACL-04 workshop*, 2004.

[53] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder, “(meta-) evaluation of machine translation,” in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, pp. 136–158.

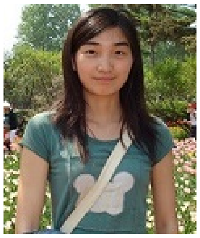
[54] J. Carletta, “Assessing agreement on classification tasks: The kappa statistic,” *Computational Linguistics*, pp. 249–254, 1996.

[55] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33, no. 1, p. 159, March 1977.

[56] F. J. Och, “Minimum error rate training in statistical machine translation,” in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 160–167.



Jizhou Huang received the Master’s Degree in Dec. 2006 from Chongqing University, Chongqing, China. Since 2014, he has been a Ph.D. candidate at the Department of Computer Science, Harbin Institute of Technology, Harbin, China. He is currently a principal scientist at Baidu Inc. His current research interests include natural language processing, information retrieval, and recommendation system.



Yaming Sun received her Ph.D. Degree in Jan. 2018 from the Department of Computer Science, Harbin Institute of Technology, Harbin, China. She is currently an engineer at Baidu Inc. Her current research interests include natural language processing, entity disambiguation, and representation learning.

Wei Zhang has a B.S. and M.S. from the college of Computer Science and Technology, Zhejiang University, Hangzhou, China. He is currently a senior engineer at Baidu Inc. His research interests include recommendation system, information retrieval, and natural language processing.

Haifeng Wang Ph.D., Vice President of Baidu, head of Baidu’s Artificial Intelligence Group (AIG) and Baidu Research, head of China’s National Engineering Laboratory of Deep Learning Technology and Application. Dr. Wang was the president of the Association for Computational Linguistics (ACL) in 2013, and is an ACL fellow. He has served as program chair, workshop chair, tutorial chair, area chair, industry chair, and sponsorship chair for several top conferences including SIGIR, ACL, IJCAI, KDD, COLING, IJCNLP, etc., as well as associate editor, guest editor and reviewers for some academic journals. Dr. Wang was honored the Second Prize of China’s National Science and Technology Progress Award in 2015, and National Scientific Innovation and Advancement Award in 2017.

Ting Liu received his Ph.D. Degree in 1998 from the Department of Computer Science, Harbin Institute of Technology, Harbin, China. He is a full professor of Department of Computer Science, and the director of Research Center for Social Computing and Information Retrieval (HIT-SCIR) from Harbin Institute of Technology. His research interests include information retrieval, natural language processing, and social media analysis.