Exploring Human-Like Reading Strategy for Abstractive Text Summarization

Min Yang,[1] **Qiang Qu,**[1*] **Wenting Tu,**[2] **Ying Shen,**[3] **Zhou Zhao,**[4] **Xiaojun Chen**[5]
[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2]Shanghai University of Finance and Economics
[3]Peking University Shenzhen Graduate School, [4] Zhejiang University, [5]Shenzhen University
{min.yang, qiang}@siat.ac.cn, tu.wenting@mail.shufe.edu.cn
shenying@pkusz.edu.cn, zhaozhou@zju.edu.cn, xjchen@szu.edu.cn

Abstract

The recent artificial intelligence studies have witnessed great interest in abstractive text summarization. Although remarkable progress has been made by deep neural network based methods, generating plausible and high-quality abstractive summaries remains a challenging task. The human-like reading strategy is rarely explored in abstractive text summarization, which however is able to improve the effectiveness of the summarization by considering the process of reading comprehension and logical thinking. Motivated by the human-like reading strategy that follows a hierarchical routine, we propose a novel Hybrid learning model for Abstractive Text Summarization (HATS). The model consists of three major components, a knowledge-based attention network, a multi-task encoder-decoder network, and a generative adversarial network, which are consistent with the different stages of the human-like reading strategy. To verify the effectiveness of HATS, we conduct extensive experiments on two real-life datasets, CNN/Daily Mail and Gigaword datasets. The experimental results demonstrate that HATS achieves impressive results on both datasets.

Introduction

Abstractive text summarization aims to generate condensed and concise summaries that retain the salient information and overall meaning of the source articles. As opposed to the extractive text summarization, which extracts the best summarizing components from the input documents, abstractive summaries potentially contain new phrases that do not appear in the source articles. Abstractive text summarization has attracted increasing attention recently due to its broad applications in natural language processing (NLP).

Recent advances in the deep learning based approaches (i.e., the sequence to sequence framework) (Rush, Chopra, and Weston 2015; Nallapati et al. 2016) have taken the state-of-the-art of abstractive text summarization to a new level. The general idea behind these methods is to encode the input documents as vector representations with a long short-term memory (LSTM), and then use another LSTM as the decoder to generate the corresponding summaries. The sequence to sequence (seq2seq) framework has become the mainstream due to their capability of capturing the semantic and syntactic relations between raw documents and their summaries in a scalable and end-to-end way.

Although great efforts have been devoted to the abstractive text summarization, generating plausible and high-quality abstractive summaries remains a challenging task in practice, because computers lack human knowledge as well as language capability to understand the entire text and then write a summary highlighting its main points. Despite the significance of human reading ability, to date, no attempt has been devoted to exploring the human-like reading strategy in abstractive text summarization (i.e., how humans summarize an article).

When humans read, comprehend and summarize a piece of text, their exploration of the reading process organizes itself most naturally into an examination of three phases: general understanding of the document, task-specific reading comprehension, and polishing process (Avery and Graves 1997; Toprak and Almacıoğlu 2009). Humans first set the purpose of reading and pre-view the text quickly with prior (background) knowledge to get a general understanding of the document. As revealed by previous work (Tarchi and others 2017), prior knowledge[1] facilitates and enhances human reading, which is expected to have a large influence on the reading process as it helps the reader to construct a coherent mental representation of the document and gain an overview of the content in the document.

Reading comprehension is a central component of skilled reading, which is essential to ensure the good understanding of a document. To construct the meaning of a text, readers have to go beyond literal information through the generation of inferences. Indeed, inferences are what make readers move from a mere interpretation of individual sentences to a global meaning that integrates multiple sentences (Tarchi and others 2017). After reading the text thoroughly, readers are required to find the task-specific information and make many different types of inferences, such as pointing out the category and retaining the salient information of the input documents.

Similar to human cognitive process for writing a high-quality summary, the readers will evaluate the generated

*Qiang Qu is corresponding author.

[1]Here, prior knowledge is defined as the reader's actual knowledge, available before a certain learning task.

summary and then polish the summary when necessary based on the evaluation signal. This polishing process creates opportunities for deeper understanding as well as error correction and ensures that the summarization goal has been met. For example, in practice, the generated summary usually needs to be edited for accuracy and fluency by adding further information and rephrasing the generated words when necessary. Overall, if one desires to create a machine intelligence possessing such a reading comprehension skill of humans for abstractive text summarization, exploring these hierarchical stages of the human-like reading strategy is quite necessary.

In this paper, we propose a novel Hybrid learning model for Abstractive Text Summarization (HATS), which mimics the process of how humans write a summary for a piece of text. Similar to previous state-of-the-art methods (Paulus, Xiong, and Socher 2017), the sequence to sequence framework is used as the backbone of our summarization system. HATS additionally consists of three components corresponding to the hierarchical stages of the human-like reading strategy. **First**, we design a knowledge-attention network to get the general understanding of the document, which leverages the commonsense knowledge from the knowledge base (KB) as prior knowledge to distinguish the important information from the input text and determine the focus of the summary. It is intuitive that the importance of each context word in the document is significantly influenced by the entity mentions in KB. **Second**, to enhance the process of reading comprehension and deeply understand a text, a multi-task learning is proposed to jointly train the task of abstractive summarization and two other related tasks: text categorization and syntax annotation. Specifically, text categorization improves the quality of locating salient information of the text and syntax annotation exploits word-level syntax to generate high-quality summaries from the language modeling perspective. **Third**, we employ a generative adversarial network (GAN) to further refine the performance of abstractive text summarization by using a discriminative model to guide the training of the generative model in an adversarial process. This adversarial process can eventually adjust the generative model to generate human-like and high-quality abstractive summaries.

We summarize our main contributions as follows:

- To the best of our knowledge, this is the first work exploring human-like reading strategy for abstractive text summarization.
- We leverage the commonsense knowledge from the knowledge base as prior knowledge to capture the important information of the input document, obtaining the general understanding of the document.
- A multi-task learning system is employed to jointly train the abstractive summarization task and two other related tasks: text categorization and syntax annotation. Specifically, text categorization helps to learn a category-specific text encoder and improve the quality of locating salient information of the text. The syntax annotation helps to generate high-quality summaries from the language modeling perspective, and thus alleviates the issues of incomplete sentences and duplicated words.
- A generative adversarial network is employed to further refine the summarization performance and generate more plausible, high-quality and human-like abstractive summaries.
- Extensive experiments are conducted to show that HATS achieves substantial improvements over the compared methods on the widely used CNN/Daily Mail and Gigaword datasets.

Related Work

Abstractive Text Summarization There has been increasing interest in generalizing the neural language model to the field of abstractive summarization based on the sequence-to-sequence model (Rush, Chopra, and Weston 2015; Nallapati et al. 2016; See, Liu, and Manning 2017). For example, Rush, Chopra, and Weston (2015) were the first to apply the attention-based encoder-decoder model to abstractive text summarization, achieving state-of-the-art performance on two sentence-level summarization datasets. Nallapati et al. (2016) proposed the attention encoder-decoder RNN that captured the hierarchical document structure and identified the key sentences and keywords in the document. See, Liu, and Manning (2017) proposed a hybrid pointer-generator network that allowed both copying words from the source text via pointing and generating words from a fixed vocabulary. Cao et al. (2018a) used existing summaries as soft templates to guide the seq2seq model. Cao et al. (2018b) exploited open information extraction and dependency parse methods to extract actual fact descriptions from the source text, and then forced the generation of summaries conditioned on both the source text and the extracted fact descriptions.

Several recent studies attempted to integrate the encoder-decoder RNN and reinforcement learning paradigms for abstractive summarization, taking advantages of both (Paulus, Xiong, and Socher 2017; Liu et al. 2018). For example, Paulus, Xiong, and Socher (2017) combined the maximum-likelihood cross-entropy loss with rewards from policy gradient reinforcement learning to reduce exposure bias. Liu et al. (2018) proposed an adversarial process for abstractive text summarization, in which the generator is built as an agent of reinforcement learning.

Human-like Reading Strategy in NLP In parallel, attempts have also been made to study how people read the natural language. Masson (1983) explored how people answer questions by first skimming the document, capturing relevant information, and carefully reading these content to get the answer. Furbach, Schon, and Stolzenburg (2014) briefly introduced the principles of cognitive computing and revealed that natural language question answering is an example of this new computing paradigm. Li, Li, and Wu (2018) presented a human-like reading strategy for document-based question answering. Based on the reading strategy, they made a good combination of general understanding of both document and question. To date, no work explores the human-like reading strategy for abstractive text summarization. Our work takes the lead in this topic.

Our Methodology

Problem Definition

Assume that each input article $X = \{x_1, x_2, ..., x_n\}$ has a corresponding reference summary $Y = \{y_1, y_2, ..., y_k\}$ and a category label C, where n and k denote the length of the input document and reference summary, respectively. Given an input article X, the abstractive summarization task tries to generate a summary $\hat{Y} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_m}\}$, where m denotes the length of the generated summary. For the text categorization task, given an input article X, our objective is to predict the category label $\hat{C}$ for the input article. For syntax annotation, we have $Z = \{z_1, z_2, ..., z_m\}$ denoting the Combinatory Category Grammar (CCG) (Steedman and Baldridge 2011) supertag sequence for the corresponding summary Y of source text X. CCG uses a set of lexical categories to represent constituents, which provides a connection between syntax and semantics of natural language. CCG supertag annotation (Clark 2002) is a task to assign lexical categories to each word in a piece of text.

Architecture of Our Approach

As discussed in Section 1, there are three phases (general understanding of the document, task-specific reading comprehension, and polishing process) when humans read, comprehend and summarize a piece of text. Accordingly, we propose a hybrid learning model HATS to simulate the process of how humans summarize an article. As depicted in Figure 1, HATS also consists of three components: a knowledge-based attention module, a multi-task learning module, and a generative adversarial network module. Next, we will elaborate each component of the proposed HATS model in details.

Knowledge-based Attention Module

The knowledge-based attention module leverages the commonsense knowledge in KB as prior knowledge to learn a knowledge-aware document **encoder** of our sequence to sequence framework, getting the general understanding of the document.

Initial Context Representations

Each word x in the input text is mapped to a low-dimensional embedding $v \in \mathbb{R}^{d_e}$ by embedding layers, where d_e denotes the dimension of word embedding. Then, the hidden states of words in the document are learned by LSTM layers. Formally, given the input word embedding v_k at index k in the input text, the hidden state $h_k \in \mathbb{R}^{d_c}$ (d_c is the number of hidden states for each LSTM unit) can be updated from the previous hidden state h_{k-1}, which is computed by

$$h_k = \text{LSTM}(h_{k-1}, v_k) \quad (1)$$

Thus, given the input document X, we can obtain the initial contextual document representation $H^{init} = [h_1, ..., h_n]$, where n is the length of the input article X.

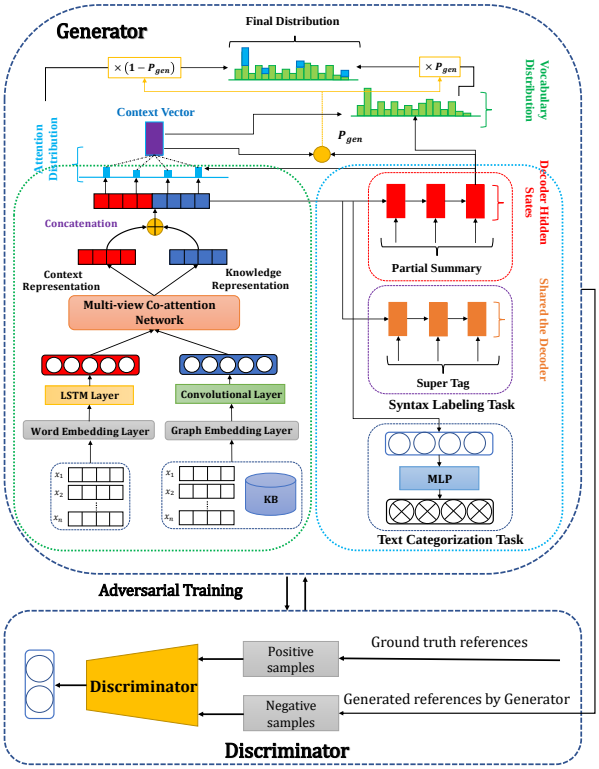

Figure 1: The overall architecture of our model.

Initial Knowledge Representations

We perform entity mention detection by n-gram matching and provide a set of top-N entity candidates from KB for each entity mention in the document. The embeddings of entities in KB are learned by DeepWalk (Perozzi, Al-Rfou, and Skiena 2014). Formally, we present candidate entities for the entity mention at time step k as $E_k = \{e_{k1}, e_{k2}, ..., e_{kN}\} \in \mathbb{R}^{N \times d_{kb}}$, where d_{kb} is the dimension of the entity embedding in KB. Then, the candidate entities are averaged to form the knowledge representation for the k-th word in the document: $\bar{E}_k = \frac{1}{N} \sum_{i=1}^{N} e_{ki} \in \mathbb{R}^{d_{kb}}$.

After obtaining the knowledge representation for each entity mention in the document, a CNN layer is then employed to capture the local n-gram information and learn a high level knowledge representation $E^{init} \in \mathbb{R}^{n \times d_k}$ (d_k is the number of hidden states of CNN):

$$E^{init} = \text{CNN}(\bar{E}) \quad (2)$$

We refer the interested readers to (Kim 2014) for the implementation details of CNN in text modeling.

Knowledge-aware Document Representations

We design a multi-view co-attention network (MCN) to distill the crucial information from both context and knowledge representations. Specifically, MCN makes uses of the interactive guidance between the context and knowledge representations to supervise the modeling of each other. In addition, MCN adopts the multi-view attention mechanism to

capture the important information from different representation subspaces at different positions.

Formally, MCN takes as input the *initial knowledge representation* as attention source to learn the knowledge-aware context representation:

$$H^{final} = flat(\Sigma \cdot H^{init}) \quad (3)$$

$$\Sigma = [\Sigma_1, \Sigma_2, \ldots, \Sigma_n] \quad (4)$$

$$\Sigma_i = softmax(\varrho(h_i, \mu(E^{init}))) \quad (5)$$

$$\varrho(h_i, \mu(E^{init})) = \tanh(U_1 h_i + V_1 \mu(E^{init}) + b_1) \quad (6)$$

where $flat$ is an operation that flattens matrix into vector form; U_1 and V_1 are attention parameters to be learned; b_1 is the bias term; $\Sigma \in \mathbb{R}^{b \times n}$ is the attention matrix, $\Sigma_i \in \mathbb{R}^b$ indicates the importance of the i-th word of the input document in multiple hops of attention, and b is the number of hops of attention; each row of attention matrix denotes one hop of attention on the whole document, namely a single-head attention.

Similarly, we can use *initial context representation* as attention source to learn the context-aware knowledge representation for the input document, denoted as E^{final}.

Finally, the knowledge-aware context representation and context-aware knowledge representation are concatenated to form the final knowledge-aware representation for input article X: $emb = \{[H_1^{final}, E_1^{final}], \ldots, [H_n^{final}, E_n^{final}]\} \in \mathbb{R}^{(n \times (d_c + d_k))}$.

Multi-task Learning Module

Motivated by the fact that humans have no difficulty performing text summarization because they have the capabilities of multiple domains (Zhao et al. 2018), we propose a multi-task encoder-decoder module to mimic the process of active reading.

Text Categorization Task

The abstractive text summarization task shares its *encoder* with the text categorization task. For the text categorization task, we feed the knowledge-aware document representation emb into a task-specific fully-connected layer followed by a *softmax* layer to predict the category probability distribution:

$$\hat{C} = softmax(V_{text} \cdot \tanh(U_{text} \cdot emb + b_{text})) \quad (7)$$

where $\hat{C}$ is prediction probabilities of text categories; V_{text}, U_{text} are weight matrices, and b_{text} is bias term.

The parameters of text categorization task are learned in a supervised manner. In particular, given a labeled training data set $\{(X_{1:M}, C_{1:M})\}$, we minimize directly the cross-entropy between the predicted label distribution $\hat{C}$ and the ground truth distribution C as the objective function:

$$J_{ML}^{text}(\theta_1) = -\sum_{i=1}^{M}\sum_{j=1}^{L} C_{ij} \log(\hat{C}_{ij}), \quad (8)$$

where $\hat{C}_i$ is the prediction probabilities of the i-th sample, C_i is the ground truth label of the i-th sample, L is the number of category classes, M is the number of training samples, θ_1 denotes the parameters related to text categorization.

Syntax Annotation and Abstractive Text Summarization Tasks

Shared LSTM Decoder LSTM decoder is shared by the abstractive summarization task and the syntax annotation task, which is essentially a language model for estimating the contextual probability of the next word except that it is conditioned on the output of the encoder. We use the knowledge-aware document representation (i.e., emb) as the initial state of the LSTM decoder. On each decoding step t, the decoder receives the input u_t (while training, u_t is the embedding of the previous word of the reference summary; at test time it is the embedding of the previous word emitted by the decoder) and update its hidden state s_t as

$$s_t = \text{LSTM}(s_{t-1}, c_t, u_t) \quad (9)$$

where c_t is the context vector at time step t. It can be computed as a weighted sum of the hidden states of the encoded input representation emb. Formally, we use the attention mechanism to calculate the attention weights β_t and the context vector c_t as

$$c_t = \sum_{i=1}^{n} \beta_{t,i} \cdot emb_i, \;\; \beta_{t,i} = softmax(f_{t,i}) \quad (10)$$

$$f_{t,i} = v^T \tanh(W_h \cdot emb_i + W_s \cdot s_t + b_{attn}) \quad (11)$$

where W_h, W_s and b_{attn} are learnable parameters.

The context vector c_t can be viewed as the representation of the source text at time step t. We then concatenate the context vector c_t and the decoder hidden state s_t at time step t and feed it to a linear function to produce the hidden vector of the decoder:

$$O_t = V[s_t, c_t] + b \quad (12)$$

The generation probabilities of the t-th word and the t-th CCG supertag can be computed by:

$$P_t^{sum\cdot} = P(y_t|\hat{Y}_{1:t-1}; X) = softmax(U^{sum\cdot} O_t + b^{sum}) \quad (13)$$

$$P_t^{syn\cdot} = P(z_t|\hat{Y}_{1:t-1}; X) = softmax(U^{syn\cdot} O_t + b^{syn\cdot}) \quad (14)$$

where the $U^{sum\cdot}$, $U^{syn\cdot}$, $b^{sum\cdot}$, $b^{syn\cdot}$ are parameters to be learned. The superscripts $syn.$ and sum denote the parameters related to supertag annotation and abstractive summarization, respectively. $\hat{Y}_{1:t-1}$ denotes the previously generated tokens. Note that P_t^{sum} denotes the word distribution over the whole vocabulary at time step t.

However, the pure generation model sometimes suffers from the out-of-vocabulary generation issue and produces many "UNK" tokens in the summary. To alleviate this limitation, copy mechanism is widely adopted in recent abstractive summarization systems (Gulcehre et al. 2016; Gu et al. 2016; See, Liu, and Manning 2017). Similar to the work (See, Liu, and Manning 2017), in this study the generation probability $p_{gen} \in [0, 1]$ for time step t is calculated from the context vector c_t, the decoder state s_t, and the decoder input u_t:

$$P_{gen} = \sigma(V_c^T c_t + V_s^T s_t + V_u^T u_t + b_{gen}) \quad (15)$$

where vectors V_c, V_s, V_u and scalar b_{gen} are learnable parameters.

For each timestep t, given a candidate token w_j (j denotes the index of the vocabulary), if w_j is out-of-vocabulary token, then $p_t^w(w_j) = 0$, if it does not appear in the source text, then $a_{t,j} = 0$.

$$\bar{P}_t^{sum\cdot}(w_j) = P_{gen} * P_t^{sum\cdot}(w_j) + (1 - P_{gen}) * \sum a_{t,j} \quad (16)$$

For the syntax annotation and summarization generation subtasks, we employ the minimum negative log-likelihood estimation. Specifically, the objective is the sum of the negative log likelihood of the target word/supertag at each decoding step.

$$J_{ML}^{sum\cdot}(\theta_2) = -\sum_t^m \log(\bar{P}_t^{sum\cdot}) \quad (17)$$

$$J_{ML}^{syn\cdot}(\theta_3) = -\sum_t^m \log(P_t^{syn\cdot}) \quad (18)$$

where m is the length of the sequence during decoding phase.

Joint Training

For the purpose of improving the shared LSTM encoder and LSTM decoder, we train these three related tasks simultaneously. The joint multi-task objective function is minimized by:

$$J_{ML}(\Theta) = \lambda_1 J_{ML}^{text} + \lambda_2 J_{ML}^{sum\cdot} + \lambda_3 J_{ML}^{syn\cdot} \quad (19)$$

where Θ denotes the collective parameters of the model. λ_1, λ_2 and λ_3 are hyper-parameters that determine the weights of the three objectives. Here, we set $\lambda_1 = \lambda_2 = 0.45$, and $\lambda_3 = 0.1$, which are determined by performing the grid search on a validation set.

Policy Gradient for Summary Generation However, the maximum likelihood estimation (MLE) method suffers from two main issues. First, the evaluation metric is different from the training loss. Second, the input of the decoder at each time step is often the previous ground-truth word during training. This exposure bias (Ranzato et al. 2016) leads to error accumulation at the testing phase. To alleviate the aforementioned issues when generating summaries, we also optimize directly for ROUGE-1 since it achieves best results among the alternatives such as METEOR (Lavie and Agarwal 2007) and BLEU (Papineni et al. 2002), by using policy gradient algorithm, and minimize the negative expected rewards:

$$J_{RL}^{sum} = (r(\hat{y}) - r(y^s)) \sum_t^m \log\ p(y_t^s | Y_{1:t-1}^s; X) \quad (20)$$

where $r(\hat{y})$ is the reward of a greedy decoding generated sequence $\hat{y}$, and $r(y^s)$ is the reward of sequence y^s generated by sampling among the vocabulary at each step.

After pre-training the proposed model by minimizing the joint ML objective (see Eq.(19)), we switch the model to further minimize a mixed training objective, integrating the reinforcement learning objective J_{RL}^{sum} with the original multi-task loss J_{ML}:

$$J_{mixed}(\Theta) = \beta J_{ML} + (1 - \beta) J_{RL}^{sum} \quad (21)$$

where β is a hyper-parameter, and we set β=0.1; Θ denotes the set of parameters of the encoder-decoder framework.

Generative Adversarial Network Module

Generative adversarial network (GAN) (Goodfellow et al. 2014) is proposed to refine the performance of text summarization. GAN consists of a generative model G and a discriminative model D that compete in a minimax game with two players: The discriminative model tries to distinguish the real high-quality summaries from the training dataset or generated by G, and the generative model G tries to fool the discriminative model to generate plausible summaries. Concretely, D and G play the following game on $L(D, G)$:

$$\min_G \max_D L(D, G) = \mathbb{E}_{Y \sim P_{data}}[\log D(Y)] + \mathbb{E}_{\bar{Y} \sim G}[\log(1 - D(\bar{Y}))] \quad (22)$$

Here, Y is the input data from the training set, $\bar{Y}$ is the data generated by the generative model.

Discriminative Model D

The discriminative model is a binary classifier and aims at distinguishing the input sequence as originally generated by humans or synthesized by machines. We encode the input sequence with a CNN as it shows great effectiveness in text classification (Kim 2014). We use multiple filters with varying window sizes to obtain different features and then apply a max-over-time pooling operation over the features. These pooled features are passed to a fully connected softmax layer whose output is the probability of being "original".

In the adversarial process, using the discriminator as a reward function can further improve the generative model iteratively by dynamically updating the discriminative model. Once we obtain more realistic and high-quality summaries generated by G, we re-train the discriminative model as:

$$\min_\Phi -\mathbb{E}_{Y \sim P_{data}}[\log D_\Phi(Y)] - \mathbb{E}_{\bar{Y} \sim G_\Theta}[\log(1 - D_\Phi(\bar{Y}))] \quad (23)$$

where Φ and Θ represent the parameter sets of discriminative model D and generative model G.

Generative Model G

When the discriminative model D is obtained and fixed, we are ready to update the generative model G. The loss function of our generative G is defined by Eq.(21). According to the policy gradient theorem (Sutton et al. 2000), we compute the gradient of J_{mixed} w.r.t. the parameters Θ:

$$\begin{aligned} \nabla_\Theta J_{\text{mixed}} &= \frac{1}{T} \sum_{t=1}^{T} \sum_{y_t} R((\hat{Y}_{1:t-1}, X), y_t) \cdot \nabla_\Theta(G_\Theta(y_t | \hat{Y}_{1:t-1}, X)) \\ &= \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{y_t \in G_\Theta}[R((\hat{Y}_{1:t-1}, X), y_t) \nabla_\Theta \log p(y_t | \hat{Y}_{1:t-1}, X)] \end{aligned} \quad (24)$$

where $R((\hat{Y}_{1:t-1}, X), y_t)$ is the action-value function, and we have $R((\hat{Y}_{1:t-1}, X), y_t) = D_\phi(\hat{Y}_{1:T})$, T is the length of the generated sequence. We update the parameters using stochastic gradient descent, $\hat{Y}_{1:t}$ is the generated summary up to time step t, X is the source text to be condensed.

Experimental Setup

Datasets Description

We conduct experiments on two widely used real-life datasets. The detailed properties of the datasets are described as follows.

CNN/Daily Mail Corpus We first evaluate our model on the CNN/Daily Mail Corpus (Hermann et al. 2015), which is widely used in abstractive text summarization. The dataset comprises news stories in CNN/Daily Mail websites paired with multi-sentences human-generated summaries. Totally, it consists of 287,226 training instances, 13,368 validation instances and 11,490 test instances. There are 781 tokens on average of articles and 56 tokens on average of summaries.

Gigaword Corpus The Gigaword corpus is originally introduced by (Graff et al. 2003). Following (Nallapati et al. 2016), we utilize publicly available scripts to preprocess the data[2]. Totally, there are about 3.8M training instances, 400K validation and test instances.

In all experiments, data preprocessing is performed. Each text is tokenized with a widely used natural language processing toolkit NLTK[3]. We build independent vocabularies for articles and summaries by keeping the top 20,000 words with the highest frequency. The rest words that are not included in the vocabulary are replaced by the "*UNK*" token.

For the text categorization task, we explore the source webpage of each news story, which provides a specific category for each article. We divide these data into 11 categories: Sports, Showbiz, Politics, Opinion, Tech, Travel, Health, Crime, Living, Business, and Other.

For the syntax annotation task, the training data is annotated with CCG supertags[4], where each word has a corresponding dependency label of supertags.

Implementation Details

Following the setting of (See, Liu, and Manning 2017), we use the non-anonymized version and truncate the input articles/target summaries to a maximum length of 400/100 words. We adopt Freebase as our KB in the experiments. 100-dimensional word2vec (Mikolov et al. 2013) embeddings are used to initialize the word embeddings for both datasets. For both datasets, the recurrent parameter matrices are initialized as orthogonal matrices, and we initialize the other parameters with the normal distribution $\mathcal{N}(0, 0.01)$. We set both d_c and d_k to 200. For the convolutional layer of discriminative model D, we set the number of feature maps of CNN to 200. The width of the convolution filters is set to be 2.

We first pre-train ML model for summarization with a learning rate of 0.15 (See, Liu, and Manning 2017). Then switch to *HATS* training using the Adam optimizer (Kingma and Ba 2014), with a mini-batch size of 16 and a learning rate of 0.001. We use the beam search with a beam size of 5 during decoding. Dropout (with the dropout rate of 0.2) and L_2 regularization (with the weight decay value of 0.001) are used to avoid overfitting.

Baseline Methods

In the experiments, we compare the proposed model with several strong competitors, including ABS and ABS+ (Nallapati et al. 2016), RAS-LSTM and RAS-Elman (Chopra, Auli, and Rush 2016), CopyNet (Gu et al. 2016), LenEmb (Kikuchi et al. 2016), PGC (See, Liu, and Manning 2017), DeepRL (Paulus, Xiong, and Socher 2017), and GANsum (Liu et al. 2018).

Experimental Results

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

Quantitative Evaluation

Following the same evaluation as in (Nallapati et al. 2016), we evaluate HATS items of Rouge-1, Rouge-2, Rouge-L, and human evaluation. Rouge-1 and Rouge-2 (Lin 2004) are widely used evaluation metrics for summarization tasks, which estimate the consistency between n-gram occurrences in the generated and reference summaries. Rouge-L compares the longest common sequence between the generated summary and the reference summary. For human evaluation, we evaluate the informativeness and fluency of the generated summaries by randomly select 1000 examples from the test set. Similar to (Yang et al. 2018), three human evaluators were invited to score each summary generated by all models based on their informativeness (if the summary captures important information in the article) and fluency (if the summary is written in well-formed English). They are required to score the summaries by taking the above 2 factors into consideration, where 1 indicates the lowest score and 10 indicates the highest score.

We report the ROUGE scores and human evaluation results in Tables 1-2. Our HATS model substantially and consistently outperforms the compared methods by a noticeable margin on both datasets. PGC consistently perform better than ABS. This may be because that the copy mechanism used in PGC can handle the out-of-vocabulary words. DeepRL and GANsum are better than PGC, because they utilize reinforcement learning to alleviate the exposure bias problem and optimize directly the evaluation metrics. **Our model** performs even better than the strong competitors by exploring the human-like reading strategy in abstractive text summarization.

To better understand the training process, we visualize the learning curves of HATS as shown in Figure 2. Due to the limited space, we only report the learning curves with respect to Rouge-L for CNN/Daily and Gigaword datasets. The other evaluation metrics exhibit a similar trend. As shown in Figure 2, during pre-training, our model converges after about 7 epochs for CNN/Daily and 10 epochs for Gigaword. The Rouge-L scores are further improved on both datasets when employing the GAN framework, verifying that the generator G becomes better with the effective feedback (reward) from the discriminator D.

[2]Code is available at https://github.com/kyunghyuncho/dl4mt-material

[3]http://www.nltk.org

[4]https://github.com/uwnlp/EasySRL

Methods	Rouge-1	Rouge-2	Rouge-L	Human
ABS	35.46	13.30	32.65	3.76
ABS+	35.63	13.75	33.01	3.99
RAS-LSTM	37.46	15.11	34.45	4.51
RAS-Elman	38.25	16.28	35.43	4.73
PGC	39.53	17.28	36.38	5.43
DeepRL	39.87	15.82	36.90	5.35
GANsum	39.92	17.65	36.71	5.72
HATS	**42.16**	**19.17**	**38.35**	**6.35**
w/o KB	41.54	18.64	37.53	5.86
w/o text	41.77	18.76	37.85	5.93
w/o syntax	42.05	18.98	38.12	5.82
w/o GAN	41.35	18.43	37.44	5.98

Table 1: Quantitative evaluation results for CNN/Daily Mail dataset. All the scores have a 95% confidence interval of at most $\pm$0.25.

Methods	Rouge-1	Rouge-2	Rouge-L	Human
ABS	29.55	11.32	26.42	4.05
ABS+	29.78	11.89	26.97	4.32
RAS-LSTM	32.55	14.70	30.03	4.78
RAS-Elman	33.78	15.97	31.15	4.92
PGC	33.44	16.09	31.43	5.94
DeepRL	35.16	16.75	31.68	5.33
GANsum	35.04	16.55	31.96	6.32
HATS	**36.78**	**18.65**	**33.96**	**6.53**
w/o KB	35.35	17.54	31.98	6.23
w/o text	36.08	18.17	32.35	6.35
w/o syntax	36.42	18.36	32.37	6.14
w/o GAN	35.27	17.23	31.71	6.09

Table 2: Quantitative evaluation results for Gigaword dataset. All the scores have a 95% confidence interval of at most $\pm$0.25.

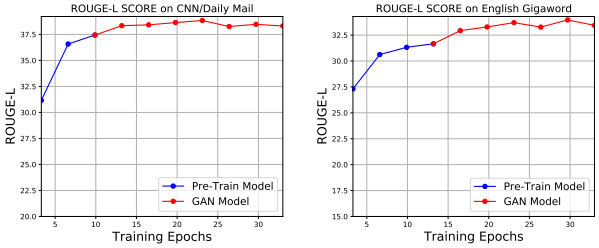

Figure 2: Learning curves of HATS in terms of Rouge-L.

Ablation Study

To investigate the effect of each component of the HATS model, we also perform the ablation test of HATS in terms of discarding external knowledge from KB (denoted w/o KB), text categorization (denoted as w/o text), syntax generation (denoted as w/o syntax), and generative adversarial network framework (denoted as w/o GAN), respectively.

The ablation results are summarized in Table 1-2 (bottom four lines). Generally, all three factors contribute a great improvement to HATS. From the results, we can observe that the Rouge and human evaluation scores decrease sharply when discarding the generative adversarial network framework. This is within our expectation since the RL reward signal coming from the discriminative model D guides the model to enjoy considerable success in generating high-quality and human-like summaries. In addition, commonsense knowledge from KB also contributes to the effectiveness of HATS. This verifies that the prior knowledge from KB helps to learn more comprehensive document representations. However, the improvement of Rouge scores by integrating syntax annotation is relatively limited. This may be explained by the fact that the issue of the incomplete sentence has little effect on the automatic evaluation metrics of summarization. In contrast, syntax annotation contributes a great improvement on human evaluation scores.

Case Study

To evaluate the proposed model qualitatively, we report some generated summaries by different models. Due to limited space, we randomly choose one generated summary by DeepRL and our model from test data for comparison. The results are reported in Table 3. We observe that our model tends to generate more specific and meaningful summary given the source article. For example, our model successfully catches the key point of "dark matter" and "the size of the map", while DeepRL attends over some trivial facts instead of the key facts.

Input: "University of Waterloo astrophysicists have created a 3D master map of the universe spanning nearly two billion light years. The innovative spherical map of galaxy superclusters is the most complete picture of our cosmic neighbourhood to date. It will help astrophysicists understand how matter is distributed in the universe and provide key insights into dark matter one of physics' greatest mysteries. Scroll down for video a slice through the Map of the nearby Universe. Our Milky Way Galaxy galaxy is in the centre, marked by a cross.(...)"
Ground-truth: "Map spans nearly two billion light years. Will help astrophysicists predict the universe's expansion. Could help identify where, and how much dark matter exists."
DeepRL: "University of Waterloo created a 3D master map of galaxy universe spanning nearly two billion light years. The innovative spherical map galaxy superclusters is the most complete picture of our cosmic neighbourhood to date. It will help astrophysicists understand how matter is distributed in the universe and provide key insights into dark matter. The lighter blue and white areas on the map represent greater concentrations of galaxies."
Ours: "3D master map of the universe spans nearly two billion light years. Innovative spherical map of galaxy superclusters is the most complete picture. Will help astrophysicists understand the universe distribution and provide key insights of dark matter."

Table 3: Example summaries.

Conclusion and Future Work

In this paper, we propose a hierarchical human-like strategy to mimic how humans approach the task of abstractive text summarization. The experimental results showed that the proposed HATS model achieved higher ROUGE scores and human evaluation results than several strong baseline methods. Moreover, the experimental results of human evaluation also verified that HATS could generate summaries with better informativeness and fluency. In the future, we would explore automatic evaluation metrics that may better match the human judgments.

Acknowledgment

This work was also partially supported by the National Natural Science Foundation of China (Grant No. 61803249), the Shanghai Sailing Program (Grant No. 18YF1407700), the SIAT Innovation Program for Excellent Young Researchers (Grant No. Y8G027), and the CAS Pioneer Hundred Talents Program. Min Yang was sponsored by CCF-Tencent Open Research Fund.

References

Avery, P. G., and Graves, M. F. 1997. Scaffolding young learners' reading of social studies texts. *Social Studies and the Young Learner* 9(4):10–14.

Cao, Z.; Li, W.; Li, S.; and Wei, F. 2018a. Retrieve, rerank and rewrite: Soft template based neural summarization. In *ACL*, volume 1, 152–161.

Cao, Z.; Wei, F.; Li, W.; and Li, S. 2018b. Faithful to the original: Fact aware neural abstractive summarization. *AAAI*.

Chopra, S.; Auli, M.; and Rush, A. M. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL*, 93–98.

Clark, S. 2002. Supertagging for combinatory categorial grammar. In *TAG*, 19–24.

Furbach, U.; Schon, C.; and Stolzenburg, F. 2014. Cognitive systems and question answering. *arXiv preprint arXiv:1411.4825*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Graff, D.; Kong, J.; Chen, K.; and Maeda, K. 2003. English gigaword. *Linguistic Data Consortium* 4(1):34.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, volume 1, 1631–1640.

Gulcehre, C.; Ahn, S.; Nallapati, R.; Zhou, B.; and Bengio, Y. 2016. Pointing the unknown words. In *ACL*, volume 1, 140–149.

Hermann, K. M.; Kočiskỳ, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *ICNIPS*, 1693–1701. MIT Press.

Kikuchi, Y.; Neubig, G.; Sasano, R.; Takamura, H.; and Okumura, M. 2016. Controlling output length in neural encoder-decoders. *EMNLP* 1328–1338.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lavie, A., and Agarwal, A. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *ACL*, 228–231.

Li, W.; Li, W.; and Wu, Y. 2018. A unified model for document-based question answering based on human-like reading strategy. In *AAAI*, 604–611.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Liu, L.; Lu, Y.; Yang, M.; Qu, Q.; Zhu, J.; and Li, H. 2018. Generative adversarial network for abstractive text summarization. In *AAAI*.

Masson, M. E. 1983. Conceptual processing of text during skimming and rapid sequential reading. *Memory & Cognition* 11(3):262–274.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B.; et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *SIGNLL*, 280–290.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

Paulus, R.; Xiong, C.; and Socher, R. 2017. A deep reinforced model for abstractive summarization. *ICLR*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: online learning of social representations. In *SIGKDD*, 701–710.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, 379–389.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, 1073–1083.

Steedman, M., and Baldridge, J. 2011. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar* 181–224.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 1057–1063.

Tarchi, C., et al. 2017. Comprehending and recalling from text: The role of motivational and cognitive factors. *Issues in Educational Research* 27(3):600.

Toprak, E. L., and Almacıoğlu, G. 2009. Three reading phases and their applications in the teaching of english as a foreign language in reading classes with young learners. *Journal of language and Linguistic Studies* 5(1):pp–20.

Yang, M.; Qu, Q.; Shen, Y.; Lei, K.; and Zhu, J. 2018. Cross-domain aspect/sentiment-aware abstractive review summarization by combining topic modeling and deep reinforcement learning. *Neural Computing and Applications* 1–13.

Zhao, W.; Wang, B.; Ye, J.; Yang, M.; Zhao, Z.; Luo, R.; and Qiao, Y. 2018. A multi-task learning approach for image captioning. 1205–1211.