Learning-to-Learn Agent Adaptation Policy for Abstractive Summarization

Hongzhang Mu
Institute of Information Engineering, Chinese Academy of Sciences School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
muhongzhang@iie.ac.cn

Tingwen Liu
Institute of Information Engineering, Chinese Academy of Sciences School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
liutingwen@iie.ac.cn

Hongbo Xu
Institute of Information Engineering, Chinese Academy of Sciences School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
hbxu@iie.ac.cn

***Abstract*—Neural abstractive text summarization models have gained a lot of popularity. Recent researches on this tasks resort to a single generative model to generate high-quality summaries. However, people adopt adaptive strategies in the process of obtaining text abstracts, such as searching for combinations from textbooks or abstracts from domain knowledge. In this paper, we use dynamic reasoning to address abstractive summarization tasks. To achieve dynamic reasoning for high flexibility when faced with diverse summarizations, we propose a Learning-to-Learn Agent Adaption policy (LLAA-Policy) network to seamlessly adapt the selection over generative agent and retrieval agent, guided by the final summarization reward. Extensive experiments on the TT News Corpus dataset demonstrate the superiority of LLAA-Policy model over previous methods and the automatically learned policy reveals the underlying summarization aspects.**

Keywords—abstractive summarization, dynamic reasoning, Learning-to-Learn Agent Adaption policy

I. Introduction

Text summarization needs not only shorter sentences with similar semantic features, but we also need it to be precise enough to contain important details in the original text, the generated sentences are semantically consistent. Human often spontaneously adopts adaptive strategies to extract important information from different sentences, such as searching from external knowledge for factoid information or reasoning over specific experience for generate summarization. Nevertheless, recent research efforts on abstractive summarization tasks have been devoted to developing a single generative model to generate high-quality summaries. Despite the distinct advantages of each model strategy, these data-driven methods have sacrificed the generalization capability by only fitting towards a few typical questions with more training samples. Usually, a single generative model is insufficient to generate factual summaries with different types in a domain-specific text summarization task.

To address these realistic difficulties, in this paper, we propose a Learning-Learn Agent Adaption policy (LLAA-Policy) network to seamlessly adapt the selection over two widely-used models, generative agent and retrieval agent. To achieve dynamic reasoning for high flexibility when faced with diverse summarization, For our challenging text summarization task, these two agents can produce complementary summarizations and our LLAA-Policy network is able to adaptively generate the better one for each specific summary.

As shown in Fig. 1, for abstracting summary, although there is information extracted, it is inconsistent with the facts; The generated summary found the truth and lacked information. In this case, our LLAA-Policy successfully adopted the right strategy to generative summarization. Specifically, Specifically, we take advantage of reinforcement learning [1], [2] and design a policy network to judge the quality of the candidate sentences provided by all agents, which incorporates a bi-directional attention architecture to match summary sequence. Moreover, to train our network in an end-to-end pipeline, we propose an optimization algorithm based on policy gradient [3] to optimize the policy network, which can be optimized towards any metric as generative summary reward depending on the task requirement. For abstractive summarization task, we adopt a combination of BLEU [4] and ROUGE-L [5] score as the reward. Extensive experiments on the abstractive summarization task demonstrate the effectiveness and superiority of our LLAA-policy network under both automatic evaluation metrics as well as human evaluation [6]. Qualitative analysis shows that our method can successfully learn a policy, which is able to adaptively Extract better sentences under the consideration of abstractive summarization types.

Text：一位摊主告诉记者，他卖蔬菜好几年，一直有豆芽，但是现在查处严格，生产豆芽必须有许可证。在位于二道街的市供销农贸市场里，大部分蔬菜摊都没有豆芽销售，只有个别摊位把少量豆芽放在隐蔽处卖。

Extractive Summarization：摊主告诉记者，他卖蔬菜好几年，一直有豆芽。现在查处严格，生产豆芽必须有许可证，

Abstractive Summarization：豆芽查处严格，商贩不卖豆芽。

Text：青岛中能后场传球失误，张璐断球反击中远射，门将王琦扑球脱手，王赟门前补射空门得分[点击观看进球视频]，上海申花2-0领先。第87分钟，吕征利用速度突破青岛中能防线后横传门前，恩里克得球回做，跟进的范凌江凌空抽射，门将王琦再次出现失误，扑球直接扑入自家球门[点击观看进球视频]，上海申花3-0领先。

Extractive Summarization：青岛中能后场传球失误，张璐断球反击中远射，门将王琦扑球脱手，王赟门前补射空门得分上海申花2-0领先。

extractive Summarization：申花客场3:0完胜青岛中能，吕征头球首开纪录，王赟补射扩大比分，范凌江替补建功。

Fig. 1. Comparison between summary from the abstractive agent and the extractive agent. The final summarization selected by our LLAA-Policy is underlined. Keywords relevant to sentences are marked as red in wrong summarization and green for correct summarization.

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892457

Our contributions are summarized in the following three aspects: 1) We make the first attempt to tackle a challenging domain specific abstractive summarization task. In particular, we introduce the abstractive summarization task and a new benchmark which requires more sophisticated dynamic reasoning to generate diverse summary. 2) We propose a Learning-to-Learn Agent Adaption policy (LLAA-Policy) network to adaptively adopt the better generate strategy over the abstractive agent and the retrieval agent for generative summary. 3) We demonstrate the superiority and effectiveness of our LLAA-Policy on the abstractive summarization task, which surpasses all previous methods under both automatic evaluation and human evaluation.

II. RELATED WORK

A. Text Summarization Task

The process of using computers to process large amounts of text and generate concise and refined content is called text summarization[7]. People can read the summary to grasp the main content of the original text, and because it reduces reading time, text summarization can bring considerable business value in time-sensitive work. There are two main approaches to text summarization: extractive summarization and abstractive summarization. Extractive summarization approaches writes summarizations based entirely on the original paragraphs extracted directly from the source document, while the abstractive approaches, which can mimic the way humans write summarization, has the potential to produce new words and phrases that are not in the source document[8] In general, the extractive approach is easier to implement because copying text directly from the source document ensures a baseline level of grammatically correctness and detail accuracy [9]. However, the complex capabilities that are crucial for high-quality summaries, such as paraphrasing, generalizing, or incorporating real-world knowledge, can only be achieved in an abstractive summarization framework. Automatic text summarization technology is an important research content in the field of natural language processing and artificial intelligence [10] It can automatically produce a concise and coherent summary from a long text or text set through computer, in which the summary should accurately reflect the central themes of source text[11]. At present, the implementation methods of automatic summarization are mainly divided into extraction method and generation method. The former extracts key text units from the original document to form the summary, and the text units include but are not limited to words, phrases, sentences, etc. the summary generated by this method usually retains the significant information of the source Chapter and has correct syntax, but it is inevitable that it is easy to produce a large amount of redundant information, The latter is based on the understanding of the input original text [12]. The model tries to understand the content of the text and can generate words that are not in the original text [13], which is closer to the essence of the summary and has the potential to generate high-quality summaries [14].

B. Extractive Summarization Approaches

Most existing abstract systems use sentences as the basic unit of extraction, because they are the smallest grammatical unit that can be expressed as sentences [15].

Yin et al. [16] first trained the expression of sentences using CNN language model, then calculated the importance of sentences using PageRank algorithm, and iteratively selected important sentences [17].

Nallapati et al. [18] regarded the abstract as a sequence classification problem and adopted Gru as the basic module of the basic sequence classifier, In addition, this work is based on CNN / daily mail data set and uses unsupervised learning to construct the data set of extraction summary.

Chen et al. [19] proposed an extraction summary generation model based on interactive text summary technology by observing the fact that humans read and understand documents many times when generating summaries.

C. Abstractive Summarization Approaches

Abstractive method belongs to the field of text generation in natural language processing [20]. Its abstract is not from the sentence splicing in the original text, but generated by using generation technology through the understanding of the semantics of the original text [21].

At present, the most popular is based on sequence to sequence. The model of (sequence to sequence, seq2seq) framework [22] can avoid tedious manual feature extraction, weight calculation, content selection and other modules. It only needs enough input and output to start training the model. Relevant researchers have proposed many interesting techniques to improve the seq2seq model and improve the performance of the model.

Vaswani et al. [23] proposed a new seq2seq network structure transformer, which only relies on feedforward network and attention mechanism to realize seq2seq architecture. The model can perform parallel computing and speed up training while improving machine translation performance. Zhang et al. [24] combined the pre training language model Bert with Combining transformer structure, a two-stage decoding model is proposed, which has achieved leading results on CNN/daily mail dataset.

Gao et al. [25] introduced the pointer generator mechanism based on the seq2seq and attention model to construct a new text summary model. This model can select and copy words from the source text, while retaining the ability to generate words from a fixed vocabulary set [26].

However, they handled the task of Language Modeling and randomly picked an existing sentence in the training corpus. Unfortunately, their experiments do not evaluate whether the techniques improve summarization correctness. In comparison, we develop Learning-Learn Agent Adaption policy (LLAA-Policy) network to seamlessly adapt the selection over two widely-used models, generative agent and retrieval agent.

III. PROPOSED METHOD

In this section, we first introduce our problem definition, which is the goal of our proposed method. Then we describe our designing of policy network and learning procedure in detail. Finally, we introduce the details about two agents adopted in our LLAA-policy network.

A. Problem Definition

LLAA-policy network is designed for text summarization tasks. Suppose the task contains m texts, denoted as $\{\mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_\mathrm{m}\}$. We have n agents which we could adapt to, that we need to select a n agent's summaries as output give

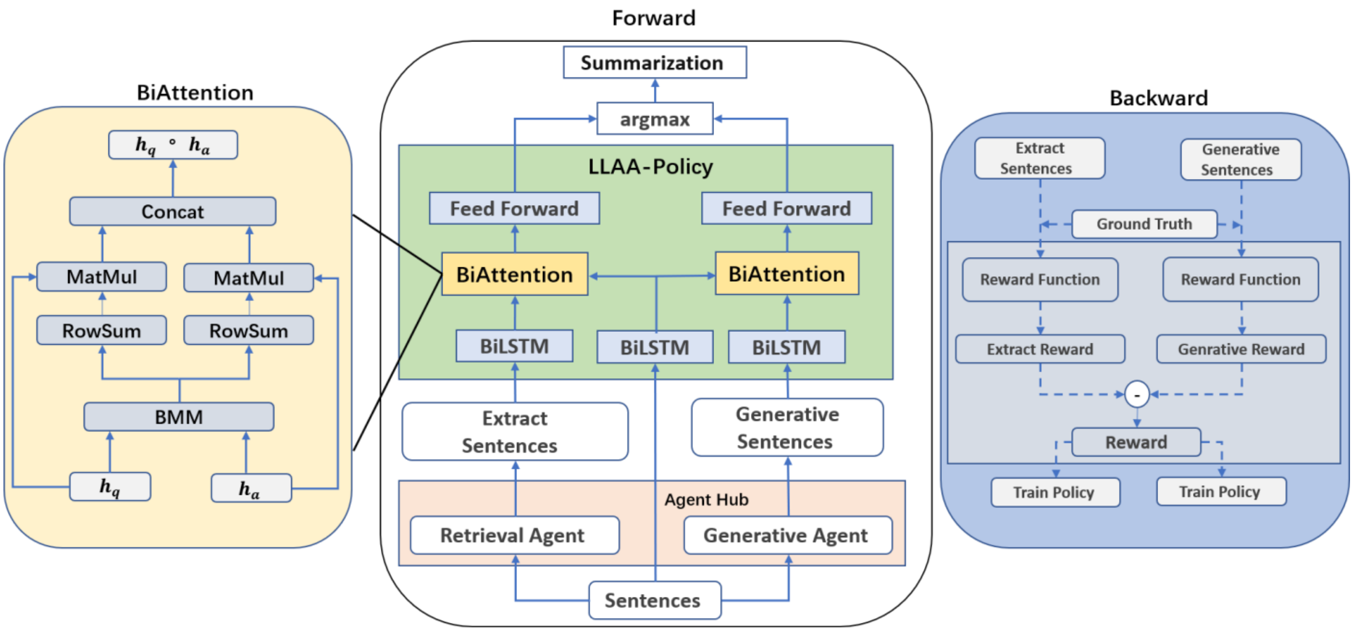

Fig. 2. Comparison between summary from the abstractive agent and the generative agent. The final summarization selected by our LLAA-Policy is underlined.Keywords relevant to sentences are marked as red in wrong summarization and green for correct summarization.

each text q_i . Thus for each text q_i we have a list of summarization candidates with size n to be selected, $\{a_{i,1}, a_{i,2}, \dots, a_{i,n}\}$, and the target ground truth summarization is denoted as a_i^* The goal of our method is to select the summarization index k from the summarization list which could maximize the reward of each text:

$$k = \underset{k}{\operatorname{argmax}}\ R(a_i^*, a_i, k),\ \ k \in [1,\ n], \tag{1}$$

where R(·) is a reward function measuring the alignment between ground truth and selected summarization. So we need to learn a policy to intelligently select the right agent for each text, by optimizing the parameters θ of policy network:

$$\theta = \underset{\theta}{\operatorname{argmax}}\ \Pi\left(a_{i,k} \mid q_i, a_i^*\right), \tag{2}$$

where Π denotes our policy function, estimated by our policy network.

B. Model Architecture

The overall architecture of LLAA-Policy network is shown in Fig. 2, which consists of a policy network and two agents, generative agent and retrieval agent. Note that this is a specific designing of the method we proposed. The method could be generalized to incorporate more agents as long as they are effective for the text summarization task. We describe the details of each part in the following.

Policy Network: Policy network aims to measure the matching quality between text and summarization sequence, and we adopt an architecture based on the attention mechanism between each text and summarization tokens. We first use two separate Bi-LSTM layers to encode text and summarization sequence, we represent the hidden states of text and summarization sequence as $\mathbf{H}_q \in \mathbb{R}^{d \times n_q}$ and $\mathbf{H}_a \in \mathbb{R}^{d \times n_a}$, where d is dimension of hidden states and n_q, n_a are the sequence length of text and summarization. We then perform a matrix multiplication between $\mathbf{H}_q$ and $\mathbf{H}_a$ to get an attention matrix $\boldsymbol{M} \in \mathbb{R}^{n_q \times n_a}$. We conduct sum operation along the column and row axis to get the attention vector for text and summarization.

$$\boldsymbol{m}_q = \sum_{j=1}^{n_a} \boldsymbol{M}_{:,J}\, , \boldsymbol{m}_a = \sum_{i=1}^{n_q} \boldsymbol{M}_{i,:}, \tag{3}$$

$$\mathbf{h}_q = \sum_{i=1}^{n_q} \mathbf{m}_{q(i)} * \mathbf{H}_{q(i)}\, , \mathbf{h}_a = \sum_{i=1}^{n_a} \mathbf{m}_{a(i)} * \mathbf{H}_{a(i)}\, , \tag{4}$$

The $\mathbf{h}_q$ and $\mathbf{h}_a$ are vectors with dimension d, representing the attended features of text and summarization sequence. Then we cat two vectors together to a feed forward network with two linear layers.

$$z_1 = \mathrm{Relu}\left(W_{z_1}\left(\mathbf{h}_q \circ \mathbf{h}_a\right) + b_{z_1}\right), \tag{5}$$

$$y = \mathrm{Sigmoid}\left(W_{z_2} z_1 + b_{z_2}\right)\ , \tag{6}$$

where W_{z1} , W_{z2} , b_{z1} , b_{z2} are parameters of linear layers, and y is our final estimation of the matching probability between text and summarization sequence

Retrieval Agent: We consider both retrieval accuracy as well as time efficiency when designing a retrieval agent. We adopt a strategy combining ROUGE-1 and ROUGE-2 score as a similarity measure to find the most similar text from the training set, considering the computation efficiency compared to ROUGE-L. Concretely, given a source text, we first try to find the target text with the highest ROUGE-2 score to the source text. If all texts in the training set have zero ROUGE-2 scores with source text, we select the text with the highest ROUGE-1 score instead as the target text. In such cases, the source texts usually have high specificity and these conditions occur only a few times in NLPCC dataset.

Generative Agent: As for the designing of the generative agent, we adopt a one layer Transformer model as our generative agent architecture. Transformer is an effective architecture in modeling the feature alignment between input and output sequence, proposed for machine translation task, which is also effective in our problem settings. For our abstractive summarization task, the encoding and decoding process of Transformer could be represented as follows:

$$\mathbf{h^e} = f_{enc}(\boldsymbol{X}), \mathbf{h^d} = f_{enc}(\boldsymbol{h^e}), \quad (7)$$

$$\mathbf{y} = \mathrm{Softmax}(\mathrm{Linear}(\mathbf{h^d})), \quad (8)$$

where $\boldsymbol{x}$ represents the embedding matrix of input sequence, $\mathbf{h}^{\mathrm{e}}$ and $\mathbf{h}^{\mathrm{d}}$ represent the matrix of encoding and decoding hidden states, and f_{enc}, f_{enc} denote the encoding and decoding layer transformation in Transformer with multi-head attention.

C. Policy Learning

We adopt the policy gradient algorithm to train our policy network. As for the reward function design, we observe that some retrieval summarizations have higher ROUGE-L score while generative summarizations usually have higher BLEU score, which means retrieval agent summarizations are more fluency consider sentence structure while generative agents have higher accuracy on the token level. Therefore, we use the combination of BLEU-4 and ROUGE-L score as our reward as following:

$$reward = \begin{cases} R(Y_r, \mathrm{Y}^*) - R(Y_g, \mathrm{Y}^*) & if\ Y_r \\ R(Y_g, \mathrm{Y}^*) - R(Y_r, \mathrm{Y}^*) & if\ Y_g \end{cases}, \quad (9)$$

where Y_r, Y_g represents the summarization of retrieval and generative agent, and Y^* is ground truth sequence. We calculate the reward depends on action estimation from the policy network. $\mathrm{R}(\cdot)$ is the function measure summarization quality described in equation 1. In our implementations, we adopt the $\mathrm{R}(.)$ as a weighted average between BLEU-4 score and ROUGE-L score:

$$R(Y, \mathrm{Y}^*) = \alpha_1\ \mathrm{BLEU\text{-}4}(\mathrm{Y}, \mathrm{Y}^*) + \ \alpha_2\ \mathrm{ROUGE\text{-}L}(\mathrm{Y}, \mathrm{Y}^*), \quad (10)$$

where α_1, α_2 are hyperparameters. During training, we find $reward\ =\ \pm\, log(|\ reward\ |)$ could smooth the training process and avoid fluctuation. The loss function could be written as:

$$Loss = -\log p(Y)\mathrm{R}(\mathrm{Y}, \mathrm{Y}^*), \quad (11)$$

where $\mathrm{R}(\cdot)$ is the reward function described in Equation 10, and $p(Y)$ is the probability of the sampled action calculated in Equation 6.

IV. Experiments and Analysis

A. NLPCC

NLPCC is a Chinese Computing Conference organized by CCF Chinese information technology special committee. One of its tasks is the news summary for Chinese microblog, which provides the required experimental data on the official website. NLPCC2015 includes 140 news articles with titles collected from major news portals, and each article corresponds to 2 manually generated standard abstracts, The original text length of different samples in the data set varies greatly, but the length of the standard summary provided is no more than 140 Chinese characters. NLPCC2017 [27] provides two training data sets with and without standard summary. Each data set contains 5000 news documents, in which each document in the data set with standard summary corresponds to one summary, The length of the abstract shall not exceed 60 Chinese characters.

B. Implementing Details

For generative agent, we adopt Transformer model with 1 layer and head number as 8 for multi-head attention. We set the dropout rate as 0.1 for the decoder module. As for the policy network, the hidden size of the LSTM [28] cell is 128, and the hidden size of feed forward layer is 512.

For the training procedure, we first train generative agent with 100 epochs with learning rate 1e-3 gradually annealing to 1e-7, then select the best model based on evaluation on the valid dataset. This pretrained generative agent is unified with the retrieval method by our LLAA-Policy network. During the training process of the policy network, parameters of the generative agent are frozen without finetuning. We keep the learning rate as 1e-5 for policy network training.

C. Baselines

Retrieval Baselines: We choose 3 retrieval methods as baselines. 1) We adopt the Document Retriever of DrQA [29] to retrieve the most similar summarization in the training data, which is used to retrieve relevant documents from Wikipedia based on keywords similarities in original work. 2) We adopt ROUGE-L as a similarity measure to retrieve the most similar text in the training set and choose its corresponding summarization as the summarization. This method is not adopted in LLAA-Policy network due to poor calculation efficiency of the ROUGE-L score. 3) The retrieval agent which we incorporate in our LLAA-Policy network, which combines the ROUGE-1 and ROUGE-2 score as the similarity measure.

Generative Baselines: For generative baselines, we consider three methods. 1) We adopt Seq2Seq model with 2-layer Bi-directional LSTM layers as encoder and decoder with the same architecture as encoder. Hidden size of LSTM cell is kept the same as our policy network. 2) The generative agent used in LLAA-Policy network, which is a 1-layer Transformer model [30]. 3） For the third baseline method, we adopt BERT [31] model as encoder and Transformer's decoder module as decoder with 2 layers, so the layer number of encoder is 12 and decoder is 2.

LLAA-Softmax: To prove the effectiveness of our learning algorithm based on policy gradient optimization, we adopt a baseline with the same architectures as LLAA-Policy except the final activation function is softmax function instead of sigmoid function, and the whole network is trained with supervised cross entropy loss. The label of summarizations provided by two agents is 1 if the reward score is higher, otherwise 0, with the same reward function as Equation 10.

For all abstractive baselines, we initialize the embedding layer with the same word2vec model [32] used in LLAA-Policy network and train all models with a combination of pretrain and train set of abstractive summarization. For retrieval baselines, we also adopt the combination of pretrain and train set as retrieval candidates pool for a fair comparison.

Note that for LLAA-Policy network, the generative agent is trained on pretrain set and the policy network is trained on the train set, so the training data in total is equivalent to all baseline methods.

D. Evaluation Metrics

Automated Metrics To evaluate the quality of predicted summarizations, we consider multiple common automated metrics, including ROUGE-1, ROUGE-2, ROUGE-L, and METEOR [33]. In addition, we also use BERT Score [34] , which computes sentence similarity using contextualized BERT embeddings and is proved to correlate better with human judgments than existing metrics[35]. Human Evaluation Due to the limitation of automatic metrics in evaluating predicted summarizations quality, that paraphrase could hardly be detected and judged fairly, we adopt human evaluation as a supplement. Specifically, we hired 3 people to evaluate each text considering three aspects. 1)Relevance. We ask people to consider whether each summarization provided by different methods contains related information to the text or it is replying with irrelevant information. 2)Fluency. Whether the summarization is fluency in semantic level and grammar error free. 3)Completeness. Whether each summarization contains all information replying each sub-text if the text contains more than one sub-texts. For the evaluation process, each person is provided with 100 texts randomly selected from the test set and asked to impersonate real text to evaluate the candidate summarizations predicted by four different methods, including Retrieval(ROUGE-N), Transformer, BERT and our LLAA-Policy network. For each text, people are asked to select the best summarization from the summarization candidates under each of the three standards. We compare the count number people choose as best for comparison, if two summarizations are the same because LLAA-Policy just selects one summarization from two agents, we add the count number to both methods.

TABLE I. AUTOMATIC EVALUATION RESULTS ON NLPCC

model	*R-1*	*R-2*	*R-L*	*METEOR*
Retrieval (ROUGE-L)	15.58	6.57	15.79	13.14
Retrieval (ROUGE-N)	15.73	6.96	15.99	13.73
Seq2Seq	17.64	7.58	17.54	15.87
Transformer	19.33	8.85	19.14	17.43
Bert	22.05	10.39	21.62	18.05
LLAA-Softmax	22.82	12.04	21.19	20.40
LLAA-Policy	23.89	12.81	22.12	20.63

E. Result Analysis

Automatic Evaluation Results TABLE I. shows the automatic evaluation results of baselines methods and our proposed method. As for evaluation metrics of ROUGE-1, ROUGE-2, ROUGE-L and METEOR, our LLAA-Policy network achieves the highest scores. So the improvements over Transformer model prove that our policy successfully makes adaptation on those texts where Retrieval(ROUGE-N) method performs better than Transformer model. We could hypothesize that if we choose a better retrieval method for LLAA- Policy model, further improvements could be made. At last, the comparison between LLAA-Softmax and LLAA-Policy on all evaluation metrics proves the effectiveness of our optimization algorithm.

We further evaluate abstractive models based on adequacy and deducibility level. We use retrieval agent, general agent and LLAA-Policy network to understand the contribution of each part. The result shows our model adaptively adopts the better generate strategy over the generative agent and the retrieval agent .We also use the Transformer encoder and decoder for abstractive summarization. These Transformer-based abstractive models use the same transformer encoder as the extractive ones and a transformer decoder with 6 layers for generation.

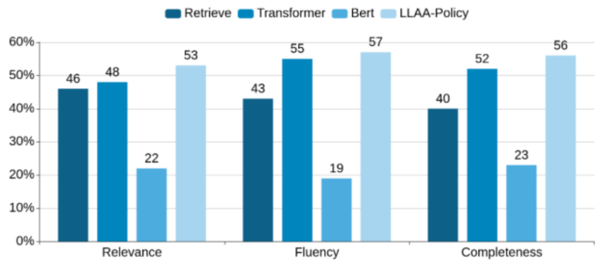

Fig. 3. Human Evaluation Results

Human Evaluation Results Human evaluation results are shown in Fig. 3. We can see that our LLAA-Policy network achieves the highest count score under all evaluation standards, including relevance, fluency, and completeness, which is in correlated with automatic evaluation results. The results prove that the adaptation policy we learned could select summarizations which not only have higher quantitative scores but also are more readable for humans and relevant to the texts.

Qualitative Analysis To further illustrate the rationality of our learned policy, we visualize the attention matrix in our policy network for one specific example in Fig. 4. In this particular example, The text contents the of football score. Although both agents provide the score, retrieval agents provide a more general definition which is lack of facts for this text. We could tell that our policy network successfully attends to "lead to" despite two summarizations share plenty of common words.

Furthermore, attention differences even for the same word in two summarizations prove that our model captures the semantic difference of the whole sentence, not just on the tokens level.

Text Types Comparison We compare the performance of our LLAA-Policy network with Retrieval(ROUGE-L) and Transformer baselines under each type of texts by METEOR score. We could see that our model achieves the highest score on consultation texts, which is the most complex text type in generative summarization, proving the superiority of our policy learning for complex scenarios. The Transformer model performs best of factoid texts probably because of the short length of this type of texts, that Transformer is better at modeling short sequence. All methods perform worst on opinion texts, and we hypothesize that opinion texts may receive more than one distinct correct summarization for a similar text, which may disturb the judgment of algorithms. Furthermore, the high scores achieved by all methods on consultation type may prove the relevance between texts of this type, proving the potential of abstractive summarization in reducing factual errors.

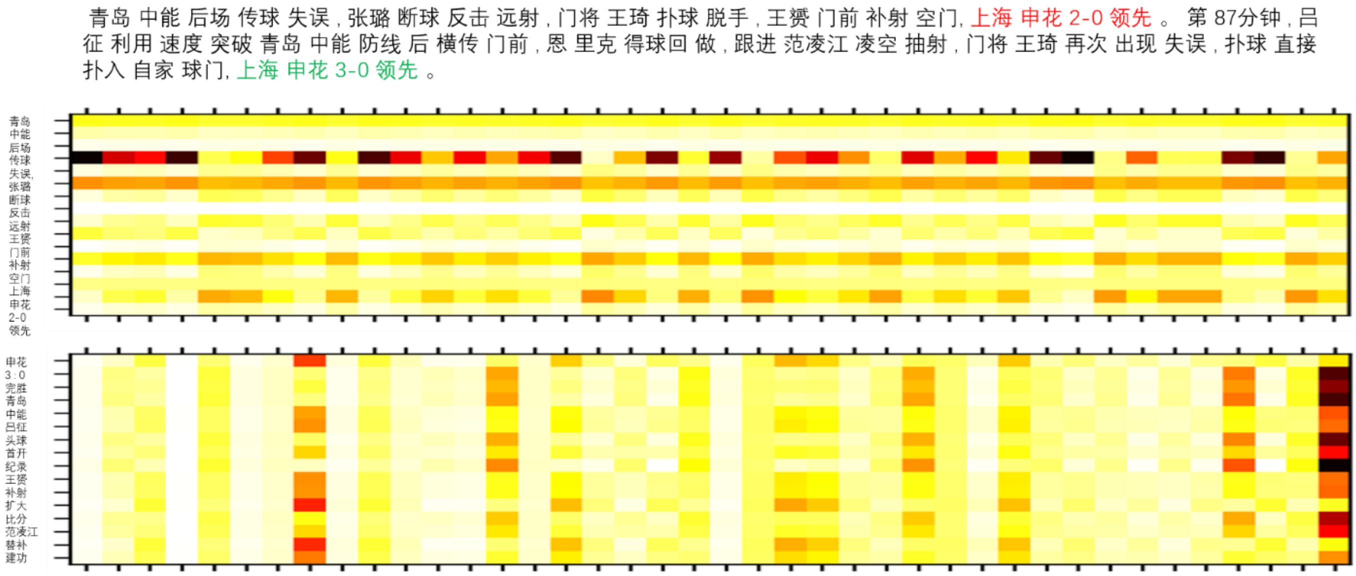

Fig. 4. Comparison between summary from the abstractive agent and the generative agent. The final summarization selected by our LLAA-Policy is underlined.Keywords relevant to sentences are marked as red in wrong summarization and green for correct summarization.

V. Conclusion

In this work, we study the current problem of generative text summarization. We introduce the Generative summarization task and current methods, which requires dynamic reasoning to resolve current question. Furthermore, we propose a Learning-to-Learn Agent Adaption policy (LLAA-Policy) network to seamlessly adapt the selection over the generative agent and the retrieval agent. Extensive experiments demonstrate the superiority of our LLAA-Policy network over all existed methods.

Acknowledgment

We would like to thank anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China under Grant No.2021YFB3100600, the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No.XDC02040400, and the Youth Innovation Promotion Association of CAS under Grant No.2021153.

References

[1] Abacha, Asma Ben, and Dina Demner-Fushman. "Recognizing question entailment for medical question answering." *AMIA Annual Symposium Proceedings*. Vol. 2016. American Medical Informatics Association, 2016.

[2] Arumae, Kristjan, and Fei Liu. "Reinforced Extractive Summarization with Question-Focused Rewards." *Proceedings of ACL 2018, Student Research Workshop*. 2018.

[3] Sutton, Richard S., et al. "Policy gradient methods for reinforcement learning with function approximation." *Advances in neural information processing systems*. 2000.

[4] Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002.

[5] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." *Text summarization branches out*. 2004.

[6] Lin, Chin-Yew, and Eduard Hovy. "Automatic evaluation of summaries using n-gram co-occurrence statistics." *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. 2003.

[7] Cheng, Jianpeng, and Mirella Lapata. "Neural summarization by extracting sentences and words." *arXiv preprint arXiv:1603.07252* (2016).

[8] Durrett, Greg, Taylor Berg-Kirkpatrick, and Dan Klein. "Learning-based single-document summarization with compression and anaphoricity constraints." *arXiv preprint arXiv:1603.08887* (2016).

[9] Dong, Yue, et al. "Banditsum: Extractive summarization as a contextual bandit." *arXiv preprint arXiv:1809.09672* (2018).

[10] Hermann, Karl Moritz, et al. "Teaching machines to read and comprehend." *Advances in neural information processing systems* 28 (2015): 1693-1701.

[11] Liu, Peter J., et al. "Generating wikipedia by summarizing long sequences." *arXiv preprint arXiv:1801.10198* (2018).

[12] Liu, Yang, and Mirella Lapata. "Text summarization with pretrained encoders." *arXiv preprint arXiv:1908.08345* (2019).

[13] Liu, Yang, Ivan Titov, and Mirella Lapata. "Single document summarization as tree induction." *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019.

[14] Luo, Ling, et al. "Reading like HER: Human reading inspired extractive summarization." *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019.

[15] Xu, Tianyang, and Chunyun Zhang. "Reinforced Generative Adversarial Network for Abstractive Text Summarization." *arXiv preprint arXiv:2105.15176* (2021).

[16] Yin, Wenpeng, and Yulong Pei. "Optimizing sentence modeling and selection for document summarization." *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[17] Erkan, Günes, and Dragomir R. Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." *Journal of artificial intelligence research* 22 (2004): 457-479.

[18] Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou. "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents." *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[19] Chen, Xiuying, et al. "Iterative document representation learning towards summarization with polishing." *arXiv preprint*

[20] Gehrmann, Sebastian, Yuntian Deng, and Alexander M. Rush. "Bottom-up abstractive summarization." *arXiv preprint arXiv:1808.10792* (2018).

[21] Li, Christy Y., et al. "Hybrid retrieval-generation reinforced agent for medical image report generation." *arXiv preprint arXiv:1805.08298* (2018).

[22] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.

[23] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.

[24] Zhang, Xueying, et al. "DSGPT: Domain-Specific Generative Pre-Training of Transformers for Text Generation in E-commerce Title and Review Summarization." *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021.

[25] Gao, Yang, et al. "Neural abstractive summarization fusing by global generative topics." *Neural Computing and Applications* 32.9 (2020): 5049-5058.

[26] Banerjee, Satanjeev, and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments." *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005.

[27] Hua, Lifeng, Xiaojun Wan, and Lei Li. "Overview of the NLPCC 2017 shared task: Single document summarization." *National CCF Conference on Natural Language Processing and Chinese Computing*. Springer, Cham, 2017.

[28] Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. "LSTM neural networks for language modeling." *Thirteenth annual conference of the international speech communication association*. 2012.

[29] Chen, Danqi, et al. "Reading wikipedia to answer open-domain questions." *arXiv preprint arXiv:1704.00051* (2017).

[30] Rush, Alexander M. "The annotated transformer." *Proceedings of workshop for NLP open source software (NLP-OSS)*. 2018.

[31] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[32] Rong, Xin. "word2vec parameter learning explained." *arXiv preprint arXiv:1411.2738* (2014).

[33] Carbonell, Jaime, and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries." *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998.

[34] Zhang, Tianyi, et al. "Bertscore: Evaluating text generation with bert." *arXiv preprint arXiv:1904.09675* (2019).

[35] Wei-Nan, Zhang , et al. "A Topic Clustering Approach to Finding Similar Questions from Large Question and Answer Archives." Plos One 9.3(2014):e71511.