2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

HGATs: Hierarchical Graph Attention Networks for Multiple Comments Integration

Huixin Zhan[§], Kun Zhang[†], Chenyi Hu[‡] and Victor S. Sheng[§*]
[§]Department of Computer Science, Texas Tech University
[†]Department of Computer Science, Xavier University of Louisiana
[‡]Department of Computer Science, University of Central Arkansas
Email: [§]{huixin.zhan,victor.sheng}@ttu.edu, [†]kzhang@xula.edu, [‡]chu@uca.edu
[*]Corresponding author: Victor S. Sheng, victor.sheng@ttu.edu

***Abstract*—For decades, research in natural language processing (NLP) has focused on summarization. Sequence-to-sequence models for abstractive summarization have been studied extensively, yet generated summaries commonly suffer from fabricated content, and are often found to be near-extractive. We argue that, to address these issues, summarizers need to acquire the co-references that form multiple types of relations over input sentences, e.g., 1-to-N, N-to-1, and N-to-N relations, since the structured knowledge for text usually appears on these relations. By allowing the decoder to pay different attention to the input sentences for the same entity at different generation states, the structured graph representations generate more informative summaries. In this paper, we propose a hierarchical graph attention networks (HGATs) for abstractive summarization with a topic-sensitive PageRank augmented graph. Specifically, we utilize dual decoders, a sequential sentence decoder, and a graph-structured decoder (which are built hierarchically) to maintain the global context and local characteristics of entities, complementing each other. We further design a greedy heuristic to extract salient users' comments while avoiding redundancy to drive a model to better capture entity interactions. Our experimental results show that our models produce significantly higher ROUGE scores than variants without graph-based attention on both SSECIF and CNN/Daily Mail (CNN/DM) datasets.**

***Index Terms*—summarization, multiple comments, graph**

I. Introduction

Summarization based on the weakly-structured text has drawn the attention of the data mining research community [1]. However, generated summaries commonly suffer from fabricated content, and are often found to be near-extractive [2]. To address these issues, some works acquire high-structured data over input, e.g., via structured representation. This line of works draw inspiration from highly-structured objects [3]. In these works, highly-structured data such as entity relationships, molecules and programs are modeled using graphs [4]. The structured knowledge for text usually appears on different types of relations. The co-references related with the same entity may span multiple sentences, making it challenging for existing sequential models to capture. A graph representation, on the contrary, produces a structured summary and highlights the proximity of relevant concepts. Motivated by the promising results of graph attention networks (GATs) on highly structured data, we propose to make use of dual decoders, a sequential sentence decoder and a graph-structured decoder, to introduce both the rich meaning and the long-distance relationships, complementing each other. Specifically, we introduce a topic-sensitive PageRank with a graph-based attention mechanism to allow the decoder to pay different attention to the input sentences for the same entity at different generation states.

Recently, with the prosperity of Web 2.0, users can freely provide their comments or reviews for any product and service. It is difficult for users to read all comments to make buying options. Thus, in order to reduce the users' workload of reading these comments, we will integrate these comments together to generate a comment summarization. However, integrating multiple comments to a comment summarization is much difficult than multi-document summarization since (1) the users' comments are informal, and (2) the data are noisy and include possibly conflicting and redundant users' comments. Therefore, our hierarchical graph attention networks (HGATs) first generate graph-based attention sentence representations via topic-sensitive PageRank for co-references that form different semantics, and then break down a typical document summarization task into salience estimation and salience selection via a greedy heuristic to address the noise.

The major contributions of this work are as follows: **1)** We propose to make use of dual decoders, a sequential sentence decoder, and a graph structure decoder to allow the decoder to pay different attentions to the input sentences for the same entity (with different semantics) at different generation states via topic-sensitive PageRank. **2)** HGATs is proposed to break down the multiple users' comments summarization into salience estimation and salience selection, and then we use a greedy heuristic to extract salient sentences while avoiding the noise in the text. **3)** We integrate HGATs into a range of existing graph based algorithms and investigate their corresponding

ASONAM '21, November 8-11, 2021,Virtual Event, Netherlands

ACM ISBN 978-1-4503-9128-3/21/11…$15.00
http://dx.doi.org/10.1145/3487351.3488322

performance on two weakly structured summarization datasets.

II. Method

A. Knowledge Graph Construction

Our knowledge graph is constructed from a set of triples, where each triple is composed of a subject, the predicate, and its object. We utilize Stanford CoreNLP [5] to first obtain outputs from co-references and open information extraction (OpenIE) models [6]. Next, we extract the $< subject, predicate, object >$ triples from the OpenIE. As an example, we will have the triple in the form of *(Louvre, islocated, Paris)* for the sentence "*The Louvre is located in Paris.*" Our KG embeddings utilize TransR [7].

B. GRU Encoder

Given a cluster $\mathbf{C}$ of K multiple comments with I sentences $(s_1, s_2, \cdots, s_I)$ in total. For each sentence s_i of L words $(w_1, w_2, \cdots, w_L)$, the word encoder, GRU^{word}, sequentially updates its hidden state after receiving each word: $h^k_{i,l} = GRU^{word}(h^k_{i,l-1}, w^k_{i,l})$, where k is the index of the comment, i is the index of the sentences, and l is the index of each word. The last hidden state (after the word encoder receives "**eos**") is denoted as $h^k_{i,-1}$, and is used as the embedding representation of the sentence s^k_i, denoted as x^k_i. We then use gated recurrent units GRU^{sent} to recurrently update hidden states at each time step t: $h^k_i = GRU^{sent}(h^k_{i-1}, x^k_i)$. For each comment k, a pseudo sentence of an "**eod**" token is appended at the end of the comment. Note that for the $k+1$-th comment, the next hidden state when the sentence encoder receives "**eod**" is treated as the representation of the last hidden state $\mathcal{H}$ in the k-th comment, denoted as $\mathcal{H} = h^k_{-1}$.

C. Graph Decoder with Attention

The decoder is used to generate output sentences $\{s'_j\}$ according to the representation of the input sentences in multiple comments. The GRU-based sentence decoder GRU^{dec_sent} receives the last representation h^k_{-1} as the initial state $h^{k'}_0 = \mathcal{H}$, and predicts the decoded sentences sequentially, by $h^{k'}_i = GRU^{dec\_sent}(h^{k'}_{i-1}, x^{k'}_{i-1})$, where $x^{k'}_{i-1}$ is the encoded representation of the previously generated sentence x^k_{i-1}. The word decoder GRU^{dec_word} receives a sentence representation $h^{k'}_i$ as the initial state and predicts the word representations sequentially, by $h^{k'}_{i,l} = GRU^{dec\_word}(h^{k'}_{i,l-1}, w^{k'}_{i,l})$, where $w^{k'}_{i,l}$ is the embedding of the previously generated word $h^k_{i,l}$. The predicted word representations are mapped to vectors of the vocabulary size dimension, and then normalized by a softmax layer as the probability distribution of generating the words in the vocabulary. In the k-th comment, the attention mechanism sets a different $\mathcal{H}_j$ (the j-th sentence representation) when generating the j-th sentence to allow the decoder to pay different attention to the input sentences with different semantics at different generation states by

$$\mathcal{H}^k_j = \sum_i \alpha^j_i h^k_i, \tag{1}$$

where α^j_i indicates how much the i-th original sentence contributes to generating the j-th sentence. in our topic-sensitive PageRank augmented summarization, a graph G is constructed to rank the original sentences. The nodes $\mathcal{V}$ are the set of n sentences to be considered, and the edges $\mathcal{E}$ are the relations between the sentences, which are typically modeled by ranking the triples in a topic relevance order. Let $W \in \mathcal{R}^{n\times n}$ be the adjacent matrix. Then the salience of the sentences are determined by making use of the global information on the graph recursively as follows [8]:

$$\mathbf{f}^j = (1-\lambda)(I - \lambda W^j D^{j^{-1}})^{-1}\mathbf{y}, \tag{2}$$

where $\mathbf{f} = [f_1, \cdots, f_n] \in \mathcal{R}^n$ denotes the rank scores of the n sentences, λ is a decay factor, W^j is the adjacency matrix added with $h^{k'}_j$, D^j is a diagonal matrix with its (i,i)-element equal to the sum of the i-th column of W^j. In order to rank the sentences with the concern of their relevance to the topic of the multiple comments, we realize the topic-sensitive PageRank vector $\mathbf{y}$ by

$$\mathbf{y} = \begin{cases} \frac{1}{|T|}, & \mathbf{y} \in T \\ 0, & \mathbf{y} \notin T \end{cases}. \tag{3}$$

Since the attention (importance) score α^j_i is determined by the relation between h^k_i and $h^{k'}_j$, we treat the current decoding state $h^{k'}_j$ as the topic T and add it into the graph as the 0-th pseudo-sentence. Therefore, $\mathbf{y}$ is always a one hot vector and only $\mathbf{y}_0 = 1$, indicating the 0-th sentence is $x^{k'}_j$. Therefore, the scores vector $\mathbf{f}$ can be used to compute the graph-based attention when decoding $h^{k'}_j$. Inspired by [9], we adopt a distraction mechanism to compute the final attention value α^j_i, which obtains a normalization of the subtractions as the rank scores $\mathbf{f}$ of the previous step to penalize the model from attending to previously attended sentences. The graph-based attention is finally computed as follows:

$$\alpha^j_i = \frac{\max(f^j_i - f^{j-1}_i, 0)}{\sum_v(\max(f^j_v - f^{j-1}_v, 0))}. \tag{4}$$

Therefore, the graph-based attention will only focus on the sentences ranked higher over the previous decoding step. That is it concentrates more on the sentences which are both salient and novel. We can use Equation 4 to replace the typical attention and then compute a different state $\mathcal{H}_i$ by the decoder via Equation 1.

D. Sentence Salience Estimation

In addition, in order to compute the salience for a sentence given the global multiple comments cluster per product, we build a cluster embedding to represent the entire comments cluster. Given a comments cluster $\mathbf{C}$ with K comments with totally I sentences per cluster, the decoder computes the comments representation $\mathbf{d}_k$ as $\mathbf{d}_k = h^{k'}_I$, where k is the comment's index, and i is the sentence's index. For each sentence s_i in the cluster $\mathbf{C}$, we calculate the salience of s_i in the following equations, similarly to the attention mechanism in neural machine translation:

$$f(s_i) = \sigma(nn(\mathbf{d}_k, \mathcal{H}_i^k, h_i^{k'})) \quad (5)$$

$$salience(s_i) = \frac{f(s_i)}{\sum_{s_v \in \mathbf{C}} f(s_v)}, \quad (6)$$

where $\sigma(nn(\mathbf{d}_k, \mathcal{H}_i^k, h_i^{k'}))$ acts as a soft attention mechanism that decides which nodes are relevant to the current graph-level task. nn is a neural network that take the concatenation of $\mathcal{H}_i^k$ and $h_i^{k'}$ as input and outputs real-valued vectors.

E. Greedy Heuristic

Given the salience score estimation, we apply a simple greedy procedure to select sentences. Sentences with higher salience scores have higher priorities to be selected. First, we sort sentences in the descending order of the salience scores. Then, we select one sentence from the top of the list and append it to the summary if the sentence is of a reasonable length (8-55 words) and is not redundant. The sentence is redundant if the tf-idf cosine similarity between the sentence and the current summary is above 0.5. We select sentences this way until we reach the length limit (up to 455 words in our experiments).

F. Parameters Training in Above Modules

The model parameters include the parameters in the GRU encoder (subsection II-B), the weights in the graph layers of the graph decoder with attention (subsection II-C) that apply recursively, and the parameters for the sentence salience estimation (subsection II-D). These parameters are trained end-to-end to minimize the following cross-entropy loss between the salience prediction and the normalized ROUGE score of each sentence:

$$\mathcal{L} = -\sum_{\mathbf{C}} \sum_{s_v \in \mathbf{C}} R(s_v) \log(salience(s_v)), \quad (7)$$

where $R(s_v)$ is represented by $R(s_v) = softmax(\beta \times r(s_v))$ and $r(s_v)$ is the ROUGE-1 score by measuring with the summarization. β is a rescaling factor that can be determined from the validation dataset.

III. Experiments

A. Dataset and Metrics

We investigate the performance of our HGATs and baselines on the CNN/DM dataset[1] [10], using the exact data split provided by [11] and a specialized real-world dataset, denoted as SSECIF 200 [12]. The SSECIF 200 dataset contains comments on 200 books from Amazon. For each book, comments from 10 participants, yet with different lengths, have been packaged as a "source document". The ground truth summarization of each cluster is generated and verified by professional researchers. For evaluation, we use the ROUGE score metrics with stemming and stop words not removed as suggested by [13].

[1]For the CNN/DM data, each article is considered as a cluster.

B. Implementation Details

In the experiments with graphs, we tokenize all clusters into sentences via Stanford CoreNLP (version 3.9.1) [5]. We use the trick of [14], where all graphs in a minibatch are "flattened" into a single graph with multiple disconnected components. We use HGATs with the size of a node vector h_v^t set to $D = 15$ and four hidden layers ($L = 2$). The hidden states in GRU^{sent} and GRU^{word} are all 152 dimensional vectors. For both datasets, we additionally perform an experiment with the model of [11], as implemented in OpenNMT [15], but using our parameters and proposed attention mechanism.

C. Quantitive Evaluation

We show quantitive evaluation results in Table I, where GAttention represents graph-based attention. Results for models from the literatures are obtained after retraining these models with our parameter settings. Across all tasks, the results show the advantage of our dual decoders in maintaining both the global context and the local characteristics of entities. We use the ROUGE scores [16] that evaluate the overlapping of N-grams between the system and reference summaries. We use $(\cdot)$+$(\cdot)$ to represent different encoder and decoder combinations. The results in performance between the different encoder and decoder configurations nicely show that their effects are largely orthogonal.

On the CNN/DM dataset, our HGATs gives a much better performance than (BiLSTM) + (LSTM) and See et al. (2017). We can see that all GAttention augmented models (in blue) are able to outperform the alternative methods, such as See et al. (2017) + (Pointer) and See et al. (2017) + (Pointer + Coverage) (in red). HGATs with GAttention achieves the best performance. In [11], the addition of Coverage gives a slightly better performance. However, by simply extending the seq2seq method with the graph-based attention, our method achieves a even better performance. As we can see, all the ROUGE scores for See et al. (2017) + (Pointer + GAttention) is better than the performance for See et al. (2017) + (Pointer + Coverage). On the SSECIF 200 dataset, we make similar observations. First, all GAttention augmented models (in blue) are able to outperform the alternative methods, such as See et al. (2017) + (Pointer) and See et al. (2017) + (Pointer + Coverage) (in light blue). Second, HGATs with GAttention achieves the best performance. Third, by simply extending the seq2seq method with the graph-based attention, our method achieves a even better performance. For example, HGATs with GAttention obtains 51.0 in ROUGE-1 and 36.2 in ROUGE-2.

To gain a global view of the performance of our HGATs, we also compare our approaches with other baseline multi-document summarizers for the SSECIF 200 dataset. As shown in Table II, the performances for HGATs without (w/o) attention and HGATs with traditional attention (in red) are slightly lower than the state-of-the-art RASG [17] (in blue). However, with our proposed graph-based attention and more fine-grained PageRank relation indicators for multiple comments, we observe that our HGATs with GAttention significantly outperforms the traditional graph approaches, e.g., **Centroid**,

TABLE I: Evaluation results for the sentence relation graph on two datasets respectively. The results of our HGATs and our extensions are in bold.

Methods	ROUGE-1	ROUGE-2	ROUGE-L
CNN/DM			
(BiLSTM) + (LSTM)	33.3	11.2	27.6
See et al. (2017) + (Pointer)	36.2	15.5	33.2
See et al. (2017) + (Pointer + GAttention)	**40.1**	**18.5**	**35.2**
See et al. (2017) + (Pointer + Coverage)	37.0⇓	16.3⇓	34.8⇓
See et al. (2017) + (Pointer + Coverage + GAttention)	**43.0**	**19.9**	**39.7**
HGATs with Attention	**42.0**	**19.9**	**38.4**
HGATs with GAttention	**44.7**	**21.9**	**41.8**
SSECIF 200			
(BiLSTM) + (LSTM)	34.7	11.7	28.3
See et al. (2017) + (Pointer)	44.4	26.7	38.4
See et al. (2017) + (Pointer + GAttention)	**47.9**	**28.3**	**43.0**
See et al. (2017) + (Pointer + Coverage)	46.7⇓	27.7⇓	39.9⇓
See et al. (2017) + (Pointer + Coverage + GAttention)	**50.0**	**29.7**	**44.1**
HGATs with Attention	**49.1**	**33.0**	**44.8**
HGATs with GAttention	**51.0**	**36.2**	**47.0**

LexRank, and **G-Flow** and many state-of-the-art summarization approaches such as **SSECIF** and **RASG**. This indicates the advantage of the combinatorial dual decoders used in our HGATs.

TABLE II: Comparing our HGATs with conventional multi-document summarizers. The results for our introduced methods are in bold.

Methods	ROUGE-1	ROUGE-2
G-Flow [18]	34.0	15.4
SSECIF [12]	48.4	29.8
RASG [17]	49.3	34.1
HGATs w/o Attention	**40.1**	**18.1**
HGATs with Attention	**49.1**	**33.0**
HGATs with GAttention	**51.0**	**36.2**

D. Qualitative Evaluation

Here we highlight some observations to point out several advantages in terms of the summarization quality of our HGATs. The following text shows one sample summarization. For our proposed HGATs, the final summarization is composed from several segments in blue in two original documents, as shown in Figure 1. Besides, our HGATs does not suffer from repetition of information when comparing with other approaches. When we compare our HGATs with other baselines, we can see that the (BiLSTM) + (LSTM) model makes factual errors, wich include a nonsensical sentence and some out of vocabulary words (marked in red). The See et al. (2017) + (Pointer) model is accurate but repeats itself (marked in green). Our HGATs provides a fluent summarization while eliminates repetition.

E. Ablation Study

Figure 2a and Figure 2b show the maximum average ROUGE-2 scores achieved when the model is trained using different decay factor λ within 200 and 300 epochs for both test sets. For both datasets, when using a larger λ, the performance is better and the convergence is faster. When $\lambda = 1.0$, the model fails to train because of running into a singular matrix.

IV. Related Works

In a crowdsourcing scenario, individuals or organizations obtain goods and services from a large, relatively open and often rapidly evolving group of internet users [19, 20]. In this paper, we aim at summarizing multiple comments for

Original Comment 1 (truncated): lagos, nigeria (cnn) a day after winning nigeria's presidency, muhammadu buhari told cnn's christiane amanpour that he plans to aggressively fight corruption that has long plagued nigeria and go after the root of the nation's unrest. buhari said he'll "rapidly give attention" to curbing violence in the northeast part of nigeria, $\cdots$, he said his administration is confident it will be able to thwart criminals and others contributing to nigeria's instability,$\cdots$, the win comes after a long history of military rule, coups and botched attempts at democracy in africa's most populous nation.

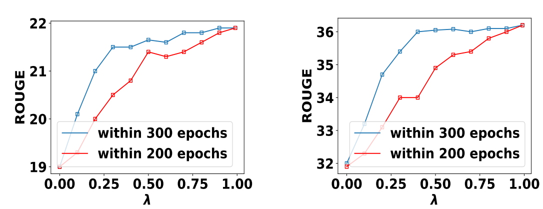
Original Comment 2 (truncated): Lagos, Nigeria (CNN)A day after winning Nigeria's presidency, muhammadu buhari told CNN's Christiane Amanpour that he plans to aggressively fight corruption that, $\cdots$, the economy is another major issue. Nigeria overtook South Africa last year as the region's largest economy. Nigeria is one,$\cdots$,doesn't trickle down to the average citizen. As many as 70% of Nigerians live below the poverty line, surviving on less than a dollar a day.

(BiLSTM) + (LSTM): UNK UNK says his administration is confident it will be able to destabilize nigeria's economy. UNK says his administration is confident it will be able to thwart criminals and other nigerians. he says the country has long nigeria and nigeria's economy.

See et al. (2017) + (Pointer): muhammadu buhari says he plans to aggressively fight cor- ruption in the northeast part of nigeria. he says he'll "rapidly give attention" to curbing violence in the northeast part of nigeria. he says his administration is confident it will be able to thwart criminals.

See et al. (2017) + (Pointer + GAttention) : muhammadu buhari tells cnn 's christiane amanpour that he will fight corruption in nigeria. nigeria is the most populous country in africa and is grappling with violent boko haram extremists. up to 70 % of nigerians live on less than a dollar a day.

Fig. 1: Comparisons of the outputs of three abstractive models for multiple documents integration. The (BiLSTM) + (LSTM) model makes factual errors, a nonsensical sentence and struggles with OOV words. The See et al. (2017) + (Pointer) model is accurate but repeats itself. The final integration is composed of several sentences in our proposed See et al. (2017) + (Pointer + GAttention).



(a) CNN/DM ROUGE-2 score vs. λ

(b) SSECIF 200 ROUGE-2 score vs. λ

Fig. 2: The results of different setting of the hyperparameter λ for both CNN/DM and SSECIF 200 test sets.

any products or services, which are posted by participants (customers) with high inconsistency and redundancy. It is obvious that integrating such multiple comments together is a challenging problem. According to our knowledge, there are a few works focus on this problem. For example, [12] proposed a self-play DQN approach for multiple comments integration. [21] proposed summarization on social context, and [17] proposed summarization based on a seq2seq framework with traditional attention. These methods usually focus on extending existing sequence encoders with a graph component. However, there are models that introduce substantial novelty in the structure or training objective of the decoder [22]. However, there are not motivated to extract the structured knowledge, e.g., co-references and their relations, in the weakly structured text. [23] learns to identify and merge coreferent concepts (entities) to reduce redundancy, determines their importance with a strong supervised model and finds an optimal summary concept map via integer linear programming. However, based on human supervised knowledge to determine the importance for entities is too expensive. Our method unsupervisely induces the attention mechanism to determine the importance instead.

V. Conclusion

In this paper, we presented a novel multiple comments summarization system HGATs that exploits the representational graph structure of co-references. Briefly, We propose to make use of dual decoders, a sequential sentence decoder, and a graph-structured decoder, to maintain the global context and local characteristics of entities, complementing each other. Our HGATs, unlike traditional sequential models or graph neural network models, demonstrated its improved salience prediction and summarization quality in both quantitive evaluation and qualitative evaluation. For quantitive evaluation, it achieved a much better performance (e.g. 51.0 in terms of ROUGE-1 and 36.2 in terms of ROUGE-2 in the SSECIF 200 dataset) than the current state-of-the-art methods do. Besides, HGATs produce natural-looking integrated comments with no noticeable negative impact on the fluency of the language over existing methods.

Acknowledgement

Chenyi Hu is partially supported by US National Science Foundation through the grant award OIA 1946391.

References

[1] K. Yao, L. Zhang, T. Luo, and Y. Wu, "Deep reinforcement learning for extractive document summarization," *Neurocomputing*, vol. 284, pp. 52–62, 2018.
[2] W. Kryściński, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," *arXiv preprint arXiv:1808.07913*, 2018.
[3] M. Allamanis, M. Brockschmidt, and M. Khademi, "Learning to represent programs with graphs," *arXiv preprint arXiv:1711.00740*, 2017.
[4] L. Huang, L. Wu, and L. Wang, "Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward," *arXiv preprint arXiv:2005.01159*, 2020.
[5] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.
[6] G. Angeli, M. J. J. Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 344–354.
[7] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.
[8] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1171–1181.
[9] Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang, "Distraction-based neural networks for document summarization," *arXiv preprint arXiv:1610.08462*, 2016.
[10] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.
[11] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.
[12] H. Rong, V. S. Sheng, T. Ma, Y. Zhou, and M. A. Al-Rodhaan, "A self-play and sentiment-emphasized comment integration framework based on deep q-learning in a crowdsourcing scenario," *IEEE Transactions on Knowledge and Data Engineering*, 2020.
[13] K. Owczarzak, J. Conroy, H. T. Dang, and A. Nenkova, "An assessment of the accuracy of automatic evaluation in summarization," in *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, 2012, pp. 1–9.
[14] M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton, "A survey of machine learning for big code and naturalness," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–37, 2018.
[15] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.
[16] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.
[17] S. Gao, X. Chen, P. Li, Z. Ren, L. Bing, D. Zhao, and R. Yan, "Abstractive text summarization by incorporating reader comments," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6399–6406.
[18] J. Christensen, S. Soderland, O. Etzioni *et al.*, "Towards coherent multi-document summarization," in *Proceedings of the 2013 Conference of the North American Chapter of The Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 1163–1173.
[19] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 614–622.
[20] J. Zhang, X. Wu, and V. S. Sheng, "Learning from crowdsourced labeled data: a survey," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 543–576, 2016.
[21] M.-T. Nguyen, C.-X. Tran, D.-V. Tran, and M.-L. Nguyen, "Solscsum: A linked sentence-comment dataset for social context summarization," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 2409–2412.
[22] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," *arXiv preprint arXiv:1805.11080*, 2018.
[23] T. Falke, C. M. Meyer, and I. Gurevych, "Concept-map-based multi-document summarization using concept coreference resolution and global importance optimization," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017, pp. 801–811.