ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10890022

Integrating Concept Associations for Query Focused Knowledge Summarization

Jian Wang
Shandong University
Jinan, China
wangjian026@126.com

Zhi Liu
Shandong University
Jinan, China
liuzhi@sdu.edu.cn

Yuqing Sun*
Shandong University
Jinan, China
sun_yuqing@sdu.edu.cn

Xin Li*
Shandong University
Jinan, China
lixincas@126.com

***Abstract*—Knowledge summarization task aims to summarize the knowledge scattered in multiple documents for answering a query. In this paper, we adopt the concept relation knowledge base *ConceptNet* for the task and propose the *ConceptNet* integrated summarization method *CNSum*, where the concepts are adopted as a bridge to find the latent associations between the query and segments in documents. Besides, a semantic mixture mechanism is introduced to combine the *concept-centered* associations with the contextual semantics of segments for summarization. To evaluate the knowledge in summary without references, we introduce a Question Answer (QA) based knowledge labeling method to construct training samples. The training samples are used for training a neural evaluation model. We compare *CNSum* with multiple methods and large language models (LLMs). The results show that *CNSum* outperforms these baselines. We also evaluate the knowledge in the generated summaries by human evaluation and our neural evaluation. The results show that *CNSum* is also better than baselines on knowledge completeness. Besides, these two evaluation results are highly correlated.**

***Index Terms*—knowledge summarization, ConceptNet, neural evaluation**

I. Introduction

Query focused knowledge summarization (QFS) is widely used in many application scenarios, such as the question and answer site Quora[1]. A query may involves diverse knowledge points that have no direct correlation with the query, which makes the QFS task difficult. Although large language models (LLMs) show good ability to summarize texts, it is difficult to apply them for the task due to the limitations on the text length and the high deployment overhead. The current methods are in the form of two phases, i.e., the query-related segment selection and the summary generation. For the first phase, these methods select segments from the given documents by the semantic similarity to query [1], [2], [3], [4] or train a neural segment selector using the segments that have high word overlap with the reference summary [5], [6]. For the second phase, models such as BART [7] and Vicuna [8] can be adopted to summarize the selected segments [6]. However, some segments that play important roles in associating the query-related knowledge are overlooked by the above methods since these segments have latent associations with query.

Another challenge is to evaluate the knowledge in summaries without references since it is difficult to construct references and label key knowledge in the practical scenarios. The commonly used metrics are based on references and treat each word equally. Such as ROUGE [9] and QA-based methods [10] evaluate summaries by comparing the gram or word semantics with references. There are also some reference-free metrics. For example, the LLM based evaluation methods G-Eval[11] and ChatGPT-ZS [12] provide the summary to a LLM and ask it to generate a score. However, LLMs have its own challenges including the large overhead and phantom issues.

To find the latent associations between query and segments, inspired by human organizing the knowledge with concepts and the concept associations are the basis for judgmental thinking and reasoning, we adopt the concept associations in *ConceptNet* to build the latent segment associations, as well as using the word co-occurrences to build the direct segment associations. Based on these associations, we use Random Walk algorithm [13] to explore multi-hop segment associations and select query related segments. Then the *concept-centered* associations and the contextual semantics of segments are mixed for generating the summaries. To quantify the knowledge in summaries, we introduce a QA based knowledge labeling method to construct pseudo-samples for training a neural evaluation model. The experiments show that our summarization method outperforms multiple baselines and there is a high correlation between our neural evaluation and human evaluation.

II. Query Focused Summarization

Given a query q and a document set D, the QFS task generates a summary that captures the knowledge in D related to q. The proposed **C**oncept**N**et integrated **Sum**marization method **CNSum** includes two phases: the segment selection phase and summarization phase, as shown in Fig.1.

A. Concept Associations based Segment Selection.

The following discussions focus on analyzing segments, i.e., paragraphs, as they are more informative than sentences. In order to select query-related segments, we construct a *query sensitive* association graph on segments, as shown in

This work was supported by the National Natural Science Foundation of China (62376138), the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007) and the Key R&D Program of Shandong Province (2023CXGC010801).

*Corresponding author

[1]https://www.quora.com/

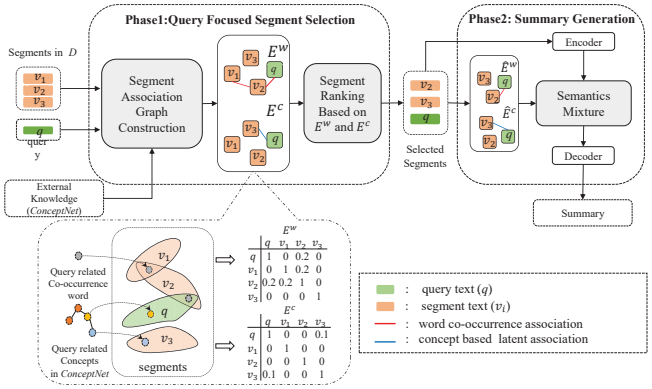

Fig. 1. Two-phase summary generation framework.

the bottom of Fig.1, where the direct associations are based on the word co-occurrences and the latent associations are constructed by the conceptual linkage in *ConceptNet*. Let $G_q = (V, E^w, E^c)$ denote the graph, where the node set $V = \{q, v_1, v_2, \ldots, v_N\}$ includes the query q and the segments v_i in D. The weighted edges include the direct and latent associations between nodes, denoted by E^w and $E^c \in R^{|V|*|V|}$, respectively.

The q-sensitive Direct Associations E^w. Nouns often play important roles in segments. So we focus on the query related nouns to build the direct associations E^w. We introduce a function $N(x)$ to find the nouns in a given text x. For any given noun k, the query related weight w_k^q is computed against its relevance to $N(q)$:

$$w_k^q = \max_{k' \in N(q)}((sim(k, k') \geq \alpha)?sim(k, k') : 0) \quad (1)$$

Here, $sim()$ computes the semantic similarity between two nouns, and α is a threshold to ignore the less relevant nouns. An edge is established between v_i and $v_j \in V$ if they have the same nouns. The weight E_{ij}^w of this edge is the sum of the weights of all co-occurring nouns:

$$E_{ij}^w = \sum\nolimits_{k \in N(v_i) \cap N(v_j)} w_k^q \quad (2)$$

The q-sensitive latent Associations E^c. We use the query related concepts in *ConceptNet* to build the latent associations E^c. Let $C(x)$ denote a function that returns the concepts appearing in both text x and *ConceptNet*. The query related concepts $CN(q)$ include $C(q)$ as well as their neighbor concepts in *ConceptNet*. The weight of any concept $c \in CN(q)$ involves two parts: the weight of concept in query that has paths in *ConceptNet* to it, and the length of the path. We use *PyTextRank*[14] to compute the weight $w_{c'}$ of concept c' in $C(q)$. Then for each $c \in CN(q)$, its weight w_c^q is computed by equation (3), where $len(c, c')$ denotes the minimum path length between c and c' in *ConceptNet*:

$$w_c^q = \max_{c' \in C(q)}(\frac{w_{c'}}{len(c, c') + 1}) \quad (3)$$

Segment v_i and v_j are associated if their concepts have a path in *ConceptNet*. Let $I(c_i, c_j)$ denote an indicative function, if there is a path between the concepts c_i and c_j, then $I(c_i, c_j)$=1, otherwise $I(c_i, c_j)$=0. The association weight E_{ij}^c is computed by the weights of the linked concepts in v_i and v_j, where $c_i \in C(v_i)$, $c_j \in C(v_j)$:

$$E_{ij}^c = \sum\nolimits_{\forall I(c_i, c_j)=1} (w_{c_i}^q + w_{c_j}^q)/2 \quad (4)$$

Segment Selection. In order to jointly utilize the latent and direct associations for segment selection, we combine E^w together with E^c, denoted as $E = Norm(E^w) + \beta Norm(E^c)$, where, β is a hyper parameter to control the effects of the latent associations, $Norm()$ is the row normalization. Then the matrix E is converted to a transition matrix by the normalization operation, based on which the Random Walk algorithm [13] is adopted to explore the multi-hop segment associations. By the iterations of random walks, we get the importance score for each segment and select the *top-M* segments $\{v_1, \cdots, v_M\}$ as the final segments.

B. Multiple Semantics Mixture for Summarization

We adopt an encoder-decoder architecture for summarization. A semantics mixture mechanism is introduced between the encoder and decoder for combining the contextual semantics of segments with their *concept-centered* associations.

Let x_i denote the text of segment v_i in the selected segments. To obtain the contextual semantics of x_i, we first encode q and the selected segments for obtaining the vector $\mathbf{x}_{ie}$ of the e-th word in x_i, then the contextual segment semantic vector $\mathbf{x}_i$ is computed as the mean of word vectors it contains: $\mathbf{x}_i = \sum_{e=0}^{l_i} \mathbf{x}_{ie}/l_i$, where l_i is the length of segment text. To obtain the *concept-centered* associations of these segments, the segment associations $\hat{E}^w$ and $\hat{E}^c$ for the M segments are pruned from E^w and E^c. Then we combine $\hat{E}^w$ and $\hat{E}^c$ by:

$$\hat{E} = Norm(\hat{E}^w) + \gamma Norm(\hat{E}^c) \quad (5)$$

We incorporate the *concept-centered* associations in $\hat{E}$ into the contextual semantics of segments, as depicted in equation (6), where $N_b(x_i)$ denotes the text set of segments that have associations with the $i-th$ segment in $\hat{E}$, e_{ij} is the attention weight predicted by an MLP layer, i.e., $e_{ij} = MLP(\mathbf{x}_i; \mathbf{x}_j)$.

$$\mathbf{x}_i = \mathbf{x}_i + \sum\nolimits_{x_j \in N_b(x_i)} (\frac{e^{(e_{ij} + \hat{E}_{ij})}}{\sum_j e^{(e_{ij} + \hat{E}_{ij})}}) \mathbf{x}_j \quad (6)$$

Next, the combined segment semantics are merged to the word vector $\mathbf{x}_{ie} = LayerNorm(W_1 * Relu(W_2(\mathbf{x}_i + \mathbf{x}_{ie})))$, where, $LayerNorm()$ is layer normalization, $Relu()$ is the activation function and W_1, W_2 are the network weights.

The decoder receives all word vectors $\mathbf{x}_{ie}$ and generates summary relying on the previously generated text, query and the selected segments.

III. Knowledge Evaluation

To evaluate the knowledge in summaries, we construct training samples by labeling the knowledge completeness to train our **Neu**ral **K**nowledge **E**valuation model(**NeuKE**). For each reference summary, we generate the pseudo-summaries by perturbing its sentences, where some knowledge are dropped

or replaced. Then we generate a set of questions based on the reference. With the help of a QA model, the pseudo-summaries can be labeled by the above question set.

Pseudo-summary Generation. For a reference summary y, its pseudo-summary set $D_y = \{t_1, t_2, \cdots\}$ is obtained by the following operations: 1) Knowledge Enhancement. Adding the sentences that are randomly selected from the source documents to y. 2) Knowledge Reduction. Randomly delete some sentences in y; Generating the summaries from each source document by the pre-trained summarization model. These summaries have less knowledge than y. 3) Knowledge Replacement. Randomly replace some sentences in y with the sentences from the source documents. 4) Knowledge Transformation. Replacing some nouns and verbs in y with their synonyms obtained from a lexical database *WordNet* [15].

Question Set Construction. We use *spacy* [16] to select noun phrases in the reference y as answers. Based on these phrases, we generate multiple questions using a t5-based QG model [17]. We use a QA model [2] to filter out the questions that can not be correctly answered by the reference y, and get the final question set $Q_y = \{q_i, a_i, p_i\}_{i=1}^{R}$, where $a_i, p_i = QA(y, q_i)$, q_i, a_i and p_i denote the question, the correct answer, and the confidence probability, respectively, R is the size of the question set.

Pseudo-summary Labeling. To label the knowledge completeness of pseudo-summaries. For a pseudo-summary $t \in D_y$ and a question q_i in Q_y, we use q_i to check the knowledge in t, denoted as $\hat{a}_i, \hat{p}_i = QA(t, q_i)$. Then the check result c_{ti} of q_i is computed by the correctness of the answers and the difference of confidence probabilities:

$$c_{ti} = \begin{cases} \min(\hat{p}_i/p_i, 1) & \text{if } a_i = \hat{a}_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The knowledge completeness label c_{t} for t is obtained by aggregating the check results of all questions in Q_y:

$$c_{\mathrm{t}} = \sum_{i=1}^{R} c_{ti}/R. \tag{8}$$

Since the pseudo-summaries generated by synonyms replacement have the same semantics as the reference, we set $c_t = 1$.

The evaluation model NeuKE. NeuKE includes the roberta-base encoder [18] and an MLP layer. We concatenate a special character $[CLS]$, q and a pseudo-summary t, i.e. $[CLS, q, t]$ as the input. The knowledge completeness is predicted by the MLP based on the embedding of $\mathbf{CLS} = Enc([CLS, q, t])$, where $Enc()$ is the encoder. We pre-train NeuKE with the cross-entropy loss on a QA corpus SQuAD2.0 [19], and fine-tune NeuKE using the *Mean Square Error* loss on the labeled pseudo-summaries.

IV. Experiments

A. Datasets and Metrics

We use the benchmark datasets DUC2005-2007[3] for experiments. Each dataset contains 45-50 queries, and each query corresponds to 30 documents. Following the commonly used mode of data slicing, when DUC2007 is used for testing, DUC2005 and DUC2006 are used for training and validating. When DUC2006 is used for testing, other two sets are used for training and validating. We evaluate results by the gram based metric ROUGE and the semantic metric BERTScore[20].

B. CNSum Implementations and comparison methods

When constructing the association graph, we set α=0.3, and use *spacy* [16] to obtain the nouns and their embeddings. The $sim()$ refers to the cosine similarity. In most cases, a concept associates many concepts within one-hop in *ConceptNet*, which is enough to build multi-hop associations between segments, thus we focus on the direct neighbor concepts. For the segment selection, we set β=0.2 and M=20. For the summary generation, we set γ as 0.1 for DUC2007 and 0.2 for DUC2006. We adopt the BART-base model [7] pre-trained on Multi-News[21] and CNN/DailyMail[22] datasets as the base model, and freeze the parameters of encoder in BART and train the model with a learning rate of 2e-5.

We select the following methods for comparison: **BERTQA** and **BERTMRC** [2] train a QA model for selecting sentences. **QUERYSUM** [2] applies a series of rules to filter sentences. **BART-CAQ** [24] first selects segments by a QA model and then summarizes the segments. **PQSUM** [25] generates a summary for each document in document set, and uses a QA model to rank sentences. **QFS-CL** [26] constructs samples of varying quality by LLM and adopts the contrastive learning to train the summarization model. **MARGE** [6] constructs pseudo samples for training the summarization model. MARGE-MN and MARGE-CD represent the model trained on Multi-News[21] and CNN/DailyMail[22] datasets, respectively. **QFS-BART**[27] utilizes the results of QA model to focus on key words during the decoding process. **Con-QFS** [28] restricts the token distribution of the summaries to conform to the constraints. **Vicuna** [8] and **ChatGPT** [29] are LLMs that show good performance on multiple tasks. For the above methods, when the input length is limited, we select the query-related segments as the input by their cosine similarity to query.

C. Results

The main comparison results are shown in Table I, where the first block is for the extractive methods and the second block for the abstractive methods. The R-1, R-2, R-SU4 and BS stand for the F1 of ROUGE-1, ROUGE-2, ROUGE-SU4 and BERTScore, respectively. We can see that CNSum achieves higher scores than the strong extractive method QUERYSUM on ROUGE-2 and ROUGE-SU4, and outperforms all abstractive methods. It also shows that LLMs still have a gap compared to the task-specific models.

The Ablation Studies. We evaluate the effects of the segment selection strategy, semantics mixture mechanism, and *ConceptNet*. The results are shown in Table II. In the -SS setting, we select segments by the cosine similarity between the embeddings of segments and query, where the embeddings are obtained by BERT[30]. In the -GA setting, we remove the

[2]https://huggingface.co/deepset/roberta-base-squad2

[3]https://duc.nist.gov/

TABLE I
COMPARISON RESULTS OF THE SUMMARIZATION METHODS.

Models	DUC2006				DUC2007			
	R-1	R-2	R-SU4	BS	R-1	R-2	R-SU4	BS
BERTQA*	38.6	8.4	13.9	–	39.8	10	14.9	–
BERTMRC*	39.6	7.8	13.6	–	39.9	8.9	14.3	–
QUERYSUM*	41.6	9.5	15.3	–	43.3	11.6	16.8	–
BART-CAQ*	38.3	7.7	12.9	–	40.5	9.2	14.4	–
PQSUM*	40.8	9.4	14.8	–	42.2	10.8	16.0	–
QFS-CL	–	–	–	–	40.2	**16.1**	–	–
MARGE-MN	39.1	9.1	14.3	0.826	42.1	11.7	16.5	0.827
MARGE-CD	40.2	9.7	15.1	0.833	42.5	12	16.9	0.834
QFS-BART	39.4	8.6	14.1	–	39.2	9.4	14.3	–
Con-QFS	39.8	8.1	13.6	0.818	38.3	9.70	14.0	0.821
CNSum	**41.3**	**10.5**	**16.0**	**0.855**	**42.9**	12.3	**17.3**	**0.854**
Vicuna	38.04	9.3	14.2	0.852	38.8	10.8	15.4	0.853
ChatGPT3.5	40.8	9.3	14.6	0.854	41.3	10.5	15.5	0.853

The results marked with '*' are taken from the paper [6]. The results of QFS-CL and QFS-BART are taken from the origin papers.

TABLE II
ABLATION RESULTS

Model	DUC2006			DUC2007		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
CNSum	41.3	10.5	16.0	42.9	12.3	17.3
-SS	-0.6	-0.4	-0.6	-1.1	-0.7	-0.7
-GA	-0.9	-0.4	-0.4	-0.9	-0.7	-0.6
-CN	-1.7	-1	-1.1	-1.5	-0.7	-0.6

latent associations $\hat{E}^c$ in semantics mixture mechanism. In the -CN setting, we eliminate the effects of *ConceptNet* in all phases. The results indicate that all the three components contribute to the model performance, where the conceptual knowledge plays the most crucial role.

hyper-parameter analysis. The hyper-parameter β and γ reflect the effects of conceptual knowledge for segment selection and summarization, respectively. The higher, the more effects contributed by *ConceptNet*. As shown in Fig. 2, introducing conceptual knowledge benefits the model performance. But the moderate small values are recommended because most of the query related knowledge can be found by the word-level direct associations.

Human Evaluation. We select 50 of the total 95 samples to five annotators who are highly educated for human evaluation. Each annotator provides a score ranging from 0 to 5 on the following dimensions: 1) Knowledge completeness(**Kc**)-The summaries should contain enough knowledge to answer the queries. 2) Fluency(**Flu**)-The summaries should not contain grammatical errors, spelling mistakes. 3) Conciseness(**Con**)-The summaries do not contain irrelevant information. Since MARGE-CD and Vicuna show good results on ROUGE and BERTScore, we choose them for comparison. We use *Krippendorff's alpha coefficient* to measure the inter-annotator agreements, and the values are 0.44 for knowledge completeness, 0.31 for fluency, 0.35 for conciseness, which are acceptable agreements [31], [32]. The results in Table III show that CNSum outperforms MARGE-CD and Vicuna on several dimensions. Unsurprisingly, Vicuna demonstrates a human-level ability in terms of language fluency.

Evaluation by NeuKE To validate the knowledge evaluation capability of NeuKE, we compared it with ROUGE and SummaCConv [33] by the correlation to human. SummaCConv evaluates summaries by computing the entailment scores

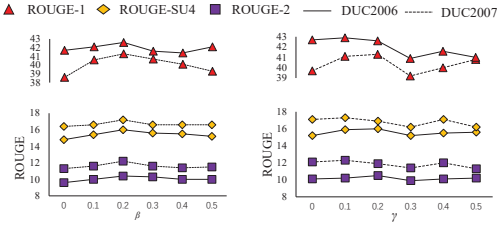

Fig. 2. Hyper-parameter analysis.

TABLE III
HUMAN EVALUATION RESULTS

Model	DUC2006			DUC2007		
	Kc	Flu	Con	Kc	Flu	Con
CNSum	**3.97**	3.73	**4.06**	**3.74**	3.64	**3.89**
MARGE-CD	3.40	3.42	3.71	3.19	3.21	3.46
Vicuna	3.60	**3.87**	3.81	3.37	**4**	3.78
GOLD	4.36	4.18	4.39	4.25	4.03	4.17

[1] GOLD is for the reference summaries.

TABLE IV
CORRELATION COMPARISON RESULTS OF EVALUATION METRICS.

Metric		DUC2006+DUC2007		
		Ken	***Spe***	***Pea***
with references	R-1	7.6	11.2	16.8
	R-2	7.1	10.4	12.2
	R-SU4	9.6	13.9	18.5
	BERTScore	20.5	28.6	29.4
w/o references	SummaCConv	9.9	13.9	7.7
	NeuKE	**18.9**	**28.5**	**27.5**

TABLE V
COMPARISON RESULTS BY NEUKE

Models	DUC2006	DUC2007
MARGE-MN	0.26	0.51
MARGE-CD	0.30	0.52
Con-QFS	0.41	0.63
Vicuna	0.58	0.63
CNSum	**0.68**	**0.72**
GOLD	0.87	0.95

between the documents and each sentence in the summary. Based on the human evaluation results of knowledge completeness on the generated summaries, we adopt the *Kendall's tau rank* (***Ken*** for short), *Spearman* (***Spe*** for shot) and *Pearson* (***Pea*** for short) coefficient to measure the correlation. The results in Table IV show that NeuKE outperforms ROUGE and SummaCConv on all coefficients. Then we compare the summarization model CNSum with the baseline methods by NeuKE. The results in Table V indicate that CNSum outperforms other methods on DUC2006 and DUC2007.

V. CONCLUSIONS

This paper proposes the knowledge summarization model CNSum and the evaluation model NeuKE. CNSum incorporates the conceptual knowledge to build the latent associations between segments so as to select segments with sufficient knowledge. Multiple metrics show that CNSum outperforms the comparison methods. NeuKE is trained on the constructed samples with varying knowledge and has a high correlation with human evaluation.

REFERENCES

[1] Tal Baumel, Matan Eyal, and Michael Elhadad, "Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models," 2018.

[2] Yumo Xu and Mirella Lapata, "Coarse-to-fine query focused multi-document summarization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, Nov. 2020, pp. 3632–3645.

[3] Ruifeng Yuan, Zili Wang, Ziqiang Cao, and Wenjie Li, "Few-shot query-focused summarization with prefix-merging," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 3704–3714.

[4] Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang, "Domain Adaptation with Pre-trained Transformers for Query-Focused Abstractive Text Summarization," *Computational Linguistics*, vol. 48, no. 2, pp. 279–320, 06 2022.

[5] Yang Liu and Mirella Lapata, "Hierarchical transformers for multi-document summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019, pp. 5070–5081.

[6] Yumo Xu and Mirella Lapata, "Generating query focused summaries from query-free resources," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online, Aug. 2021, pp. 6096–6109.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020, pp. 7871–7880.

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023.

[9] Chin-Yew Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain, July 2004, pp. 74–81.

[10] Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong, "QAFactEval: Improved QA-based factual consistency evaluation for summarization," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States, July 2022, pp. 2587–2601.

[11] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu, "G-eval: Nlg evaluation using gpt-4 with better human alignment," 01 2023, pp. 2511–2522.

[12] Zheheng Luo, Qianqian Xie, and Sophia Ananiadou, "Chatgpt as a factual inconsistency evaluator for text summarization," 2023.

[13] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The pagerank citation ranking : Bringing order to the web," in *The Web Conference*, 1999.

[14] Paco Nathan, "PyTextRank, a Python implementation of TextRank for phrase extraction and summarization of text documents," 2016.

[15] George A Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[16] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[19] Pranav Rajpurkar, Robin Jia, and Percy Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, July 2018, pp. 784–789.

[20] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[21] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev, "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019, pp. 1074–1084.

[22] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, vol. 28.

[23] Muhidin Mohamed, Mourad Oussalah, and Victor Chang, "Sdbqfsum: Query-focused summarization framework based on diversity and text semantic analysis," *Expert Systems*, vol. 41, no. 1, pp. e13462, 2024.

[24] Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J Barezi, and Pascale Fung, "Caire-covid: A question answering and query-focused multi-document summarization system for covid-19 scholarly information management," *arXiv preprint arXiv:2005.03975*, 2020.

[25] Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang, "WSL-DS: Weakly supervised learning with distant supervision for query focused multi-document abstractive summarization," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online), Dec. 2020, pp. 5647–5654.

[26] Shaoyao Huang, Ziqiang Cao, Luozheng Qin, Jun Gao, and Jun Zhang, "Contrastive learning with high-quality and low-quality augmented data for query-focused summarization," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11536–11540.

[27] Dan Su, Tiezheng Yu, and Pascale Fung, "Improve query focused abstractive summarization by incorporating answer relevance," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, Aug. 2021, pp. 3124–3131.

[28] Zhichao Xu and Daniel Cohen, "A lightweight constrained generation alternative for query-focused summarization," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2023, SIGIR '23, p. 1745–1749.

[29] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, "Language models are few-shot learners," 2020.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, vol. 30.

[31] Alex Wang, Kyunghyun Cho, and Mike Lewis, "Asking and answering questions to evaluate the factual consistency of summaries," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020, pp. 5008–5020.

[32] Ekaterina Ageeva, Mikel L. Forcada, Francis M. Tyers, and Juan Antonio Pérez-Ortiz, "Evaluating machine translation for assimilation via a gap-filling task," in *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, Antalya, Turkey, May 2015, pp. 137–144.

[33] Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst, "Summac: Re-visiting nli-based models for inconsistency detection in summarization," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 163–177, 2022.