December 2023

Topic Knowledge based Controlled Generation for Long Documents using Retrieval-based Language Models

Xuefei Zhang [a,1], Peiyang He , Tomal Deb [a], Guang Yang [a], Xuefeng Liu [a], Ziqing Hu [a], Tianyi Mao [a]

[a] *Amazon*

Abstract. Current LLM summarization systems Produce broad overviews which are disconnected from people specific interests and expectations. Basically, people preferences (topics) can be expressed by a collection of semantic keywords. Previous work exploit these keywords as extra input to generate summary. That requires additional human annotations. To tackle these constraints, we propose a novel framework, Topic Knowledge based Controlled Generation (TKCG), to control generated summaries through a set of topic keywords that are extracted automatically from source documents. First, as large language models (LLMs) are limited by context window length, we need to split the documents into small pieces like chapters acccording to the document format, as one chapter is a semantically complete section. Secondly we extract some topic keywords from source documents with a transformer-based model. These topic keywords are used to retrieve the chapters that are related to the topic. We then input the combination of topic keywords and chapters as prompts into LLM to get conditional summaries. We also demonstrate the effectiveness of TKCG on two standard datasets, MACSum and arXiv.

Keywords. text generation, LLM, word embedding, RAG

1. Introduction

Text summarization systems aim to compress documents into a condensed summary. Now extractive and abstractive summarization are two primary research topics. Extractive summarization identifies and copies key portions of the original text [1]. Abstractive summarization is focused on generating texts to expresss main ideas of documents [2]. This work focuses on abstractive summarization using LLM. While current LLM summarization systems can generate generic summaries, they often fail to align with individual people preferences and expectations. we propose a generic framework, Topic Knowledge based Controlled Generation (TKCG), that moves beyond generic summarization to provide truly people-centric experiences tailored to each user.

Summarization methods typically generate generic summaries from source documents by arbitrarily selecting content to include. However, for automatically generated summaries to be useful, they should cover information deemed important by people. For

[1] Corresponding Author, xuefegzh@amazon.com

instance, as shown in Figure 1, summaries can be tailored to specific attributes of interest, like "Pope Francis" or "blood moon." This allows for mixed-attribute control so that summaries align with people business-specific interests in various topics.

Motivated by the need for conditional summarization that aligns with people interests, in this work, we propose a novel framework, TKCG, to control summaries through specified keywords. Since people data consists of long documents that exceed language models' limited context window, we split documents into chapters, which typically convey semantically isolated idea. To identify relevant chapters related to people interests, TKCG uses KeyBERT [3] to extract topic keywords from the source document as topic knowledge for retrieving pertinent chapters. The model is instructed to generate summaries based on both the source document and keywords. We then input the combined topic keywords and selected chapters as prompts into LLMs to control the summaries. This allows people to steer summaries to cover information they care about.

We introduce a novel framework for controllable text summarization. This allows people to generate personalized summaries that fully utilize the fact that automatic summaries are created on demand. Our main contributions are: (1) We automatically extract topic keywords from source documents using transformer-based model (KeyBERT). (2) These topic keywords as knowledge guide the summarization process. (3) Our summaries adhere to people-personalized preferences. In comparisons with strong LLM summarization methods on the standard datasets MACSum [4] and arXiv [5], which contain long document summarization examples, our approach demonstrated clear advantages.

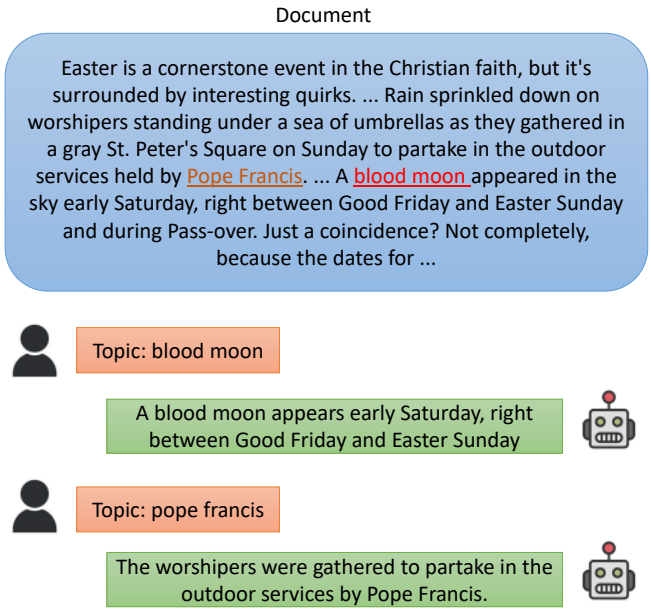

Figure 1. An illustration of topic conditional summary. For the same input source text, the system needs to generate different reference summaries (green boxes) for different topics (orange boxes).

2. Related Work

Large Language Model. OpenAI ChatGPT, Google Bard, Anthropic Claude and other LLMs are certainly having a moment—the next generation of conversational software tools promise to do everything. Pre-trained on general text, LLMs could be fine-tuned for specific tasks such as translation, question answering and text generation. Summarization is one of tasks.

Conditional Summarization. Conditional summarization is a automatic text summarization where the generated summary is controlled based on certain specified conditions or attributes. Prior conditional summarization methods often need labeled control codes for supervision [6]. Fan et al. utilized an entity tagger to extract entities from reference summaries as control codes. He et al. [7] trained a BERT-based model to label keywords for entity control. That requires additional human annotations to train the BERT-based model from entity extraction. Our proposal automatically extracts topic keywords from documents and is unsupervised. Keyword-guided summarization has been used before in various ways. These work are then provided keywords as extra input to enhance unconstrained summarization or decrease hallucinations [7], the goal is to leverage topic knowledge to guide the summarization process. Our approach can automatically extract topic keywords from source document.

3. Method

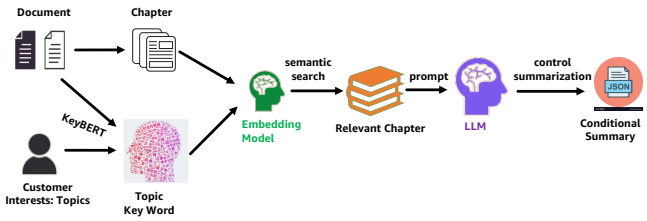

Figure 2. Topic Knowledge based Controlled Generation (TKCG) Framework

3.1. Formulation

Neural summarization models are typically trained on the conditional probability of generating a summary y given a source document x, or $p(y|x)$. This means the model produces summaries solely based on the input document without any other guidance. To better control the summary content, we can provide the model with keywords z representing user preferences. The model would then generate summaries based on the conditional probability distribution $p(y|x,z)$ - generating summary y given document x and keywords z. The keywords allow users to steer the model's summarization towards desired topics. This keyword-conditioned approach enables more controllable and customizable text summarization.

Unlike previous work, we propose the Topic Knowledge based Controlled Generation (TKCG) framework, which automatically extracts topic keywords using KeyBERT. It

incorporates user interests as topic seed words to guide keyword extraction from the document:

$$z = g(x, c) \tag{1}$$

$$y = f(x, z) \tag{2}$$

where g denotes the automatic keyword extraction algorithm, and c represents people preferences (topics), and f is conditional generation algorithm. c can be instantiated as topic seed words list. In this framework, keywords z are extracted from the source document x based on specified topics c. The extracted keywords z then serve as conditional inputs to guide the summarization process. This allows generating summaries tailored to particular user preferences. Figure 2 shows the whole TKCG framework.

3.2. Topic Keywords Extraction

Guided Topic Modeling based on BERT embedding [3] utilizes pre-defined seed topics to guide the topic modeling process. First, embeddings are created for each seed topic by passing the concatenated seed words through BERT. These embeddings are compared to document embeddings using cosine similarity to assign topic labels. Documents most similar to a seed topic receive that topic label, while other documents receive a default label. These labels are fed into Uniform Manifold Approximation and Projection to create a semi-supervised approach that nudges topic creation towards the seeded topics.

Similar to KeyBERT [3], seed words can convey the semantic meaning of a topic. It is a sensible assumption that a document containing the seed words of a topic likely belongs to that topic.

In particular, given a document x, we can determine its topic distribution z_x^c in this manner:

$$f(x, c) = \sum_{s \in c} tf(s, x) \tag{3}$$

$$z_x^c = \frac{\ln(1 + f(x, c) + \gamma)}{\sum_{s \in c}(\ln 1 + f(x, c)) + R\gamma} \tag{4}$$

where $tf(s, x)$ represents the frequency of a seed word s in document x. γ is a Dirichlet smoothing parameter set to 0.01. $R is the total number of seed words per topic$.

Instead of tf, we can use TF-IDF to better capture the importance of a word to a topic. Here, the "document" is the collection of seed words per topic. The inverse document frequency is replaced with inverse class frequency, measuring a term's information content across topics. It is computed as the log of the average seed words per topic divided by the term frequency across all topics, plus one for positivity [8].

This class-based TF-IDF models the significance of words s within a topic c. It allows deriving topic-word distributions for each document. We sum the topic seed words from a single KeyBERT run, which extracts keywords per topic [3].

3.3. Retrieval Augmented Knowledge

LLMs can acquire a substantial amount of knowledge implicitly from data, without needing external memory. However, LLMs have limitations - they struggle to expand or edit their knowledge on their own, explain their predictions, and can generate incorrect "hallucinated" information [9]. To address this, Lewis et al. [9] proposed retrieval-augmented generation (RAG) models. RAG combines the parametric memory of an LLM with a non-parametric dense vector index of Wikipedia, accessed through a retriever. RAG conditions on the same retrieved passages consistently throughout the full generated sequence.

Similarly, we use faiss as the retriever and knowledge store. Since documents are long and LLMs have limited context, we split documents into chapter-sized pieces, like paragraphs or sentences, as defined by the author. Chapters convey content on different topics. We split each document into chapters. We retrieve relevant chapters t using the topic keywords z_x^c as the knowledge. This treats the retrieved document x as a single latent variable, marginalized via a top-K approximation to get the summary probability $p(y|x)$. Specifically, the top-K chapters t are retrieved by faiss using the topic keywords. The generator then produces the output summary probability $p(y|x)$ for each document, and these are marginalized across the retrieved chapters.

$$t = \max_k f(x, z_x^c) \tag{5}$$

$$p(y|x) \approx p(y|t, z_x^c) p(z_x^c|x) \tag{6}$$

3.4. Prompt Engineering

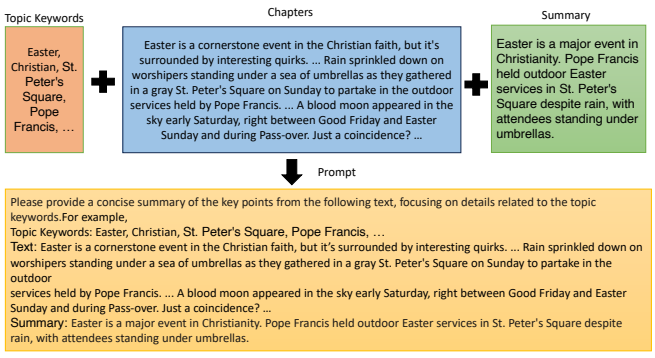

Figure 3. Prompt

Motivated by recent progress in prompting pre-trained models, we explore combining keywords and prompts for summarization. Specifically, we use both topic keywords and top retrieved chapters as prompt. We also leverage in-context learning [10] to provide a few examples as demonstration context. These examples follow natural language

Table 1. Statistic of Datasets. Train, Validation and Test are the number of samples in dataset. Avg. Token is average token length of documents in dataset.

	Train	Validation	Test	Avg. tokens
MACSum	2013	272	266	2754
arXiv	1545	247	240	5946

templates. In-context learning concatenates the instruction and demonstration context to form a prompt, which is then fed into LLMs for prediction.

In summary, our approach utilizes topic keywords, retrieved chapters, and demonstration examples via in-context learning prompts to perform summarization with LLMs. The prompts allow us to provide both content input and format examples to guide the model's summarization predictions. Figure 3 shows the prompt example.

4. Experiments

In this section, we present datasets, test TKCG quantitatively and show experimental results. Our experiment results demonstrate that TKCG significantly outperforms the baselines, highlighting the advantages of topic keywords.

4.1. Dataset and Evaluation

Dataset: We conduct experiments on two long document summarization datasets from different domains: **MACSum** [4] and **arXiv** academic papers [5]. MACSum [2] contains news articles with human-written summaries controlled for various attributes like length, extractiveness, specificity, topic, and speaker perspective. Since customrs are focused on topic-based summarization, we only condition on the topic attribute in our experiments. The arXiv dataset [3] contains lengthy scientific papers with standard structure and abstractive summaries. We use key phrases as topic seed words. Table 1 shows statistic about the two datasets.

Hyper-parameters: The number of topic seed words is 4 for each document. The hyper parameters for Claude generation is temperature as 0.1, top p as 0.9 and top k as 50.

Metric: We evaluate summarization quality using ROUGE metrics and BERTScore [11] when ground truth summaries are available.

* **ROUGE-1** calculates the overlap of individual words (unigrams) between the system and reference summaries.
* **ROUGE-2** calculates the overlap of word pairs (bigrams) between the system and reference summaries.
* **ROUGE-L** measures the length of the longest common subsequence in the reference and hypothesis texts.
* **BERTScore** evaluates language generation using BERT embeddings. BERTScore computes similarity as the sum of cosine similarities between token embeddings in the system and reference.

[2] https://github.com/psunlpgroup/MACSum
[3] https://info.arxiv.org/help/datasets.html

Table 2. F1 scores for ROUGE (1/2/L) on arXiv and MACSum datasets

Model	MACSum			arXiv		
	Rouge-1	Rouge-2	Rough-L	Rouge-1	Rouge-2	Rough-L
Claude	27.8	13.1	26.9	23.2	11.9	21.4
K-means + Claude	28.9	13.6	26.3	**25.4**	14.6	22.3
TKCG + Claude	**31.3**	**14.7**	**29.4**	24.9	**16.7**	**23.7**

Baseline: We evaluate the design choices of our model and the impact of topic keywords. As our method is designed for unlabeled data in an unsupervised fashion, we focus comparisons on unsupervised baselines. For all methods, we use Claude v1 in BedRock Service as LLM.

* **K-means clustering + Claude** This clusters sentences of text semantically into topics itteratively and passes topic information between clustering and then uses Claude to do summarization steps. In experiments, the number of clusters k is 3.
* **Claude** Regular summarization without topic keywords guidance where the LLM summarizes the full input text.

4.2. Results

Table 2 shows the performance of *Claude*, *K-means + Claude* and *TKCG + Claude* on arXiv and MACSum datasets. *K-means + Claude* decently outperforms the strong *Claude* and *K-means + Claude* baselines in terms of Rouge Score. It also performs comparably to *Claude* and *K-means + Claude* on those two datasets in terms of BERTScore as Figure 4 shows. There is a performance gap between *TKCG + Claude* and *K-means + Claude*, possibly due to the effect of topic keywords as TKCG uses KeyBERT [3] to extract key phrases from documents based on BERT Embedding.

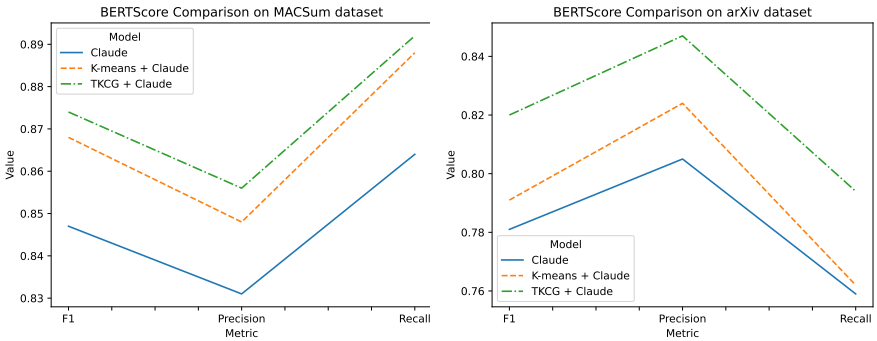

Figure 4. BERTScore for Precision, Recall and F1 on MACSum and arXiv datasets

5. Conclusion and Future Work

Current LLM summarization systems produce generic summaries that do not align with user preferences. To address this, we propose the Topic Knowledge based Controlled Generation (TKCG) framework to control summaries using automatically extracted topic keywords from documents. Experiments on two datasets demonstrate TKCG's efficacy.

TKCG has limitations in handling unrelated input, potentially causing problematic model behaviors. For example, if a user mistakenly inputs "war gun kill" for an NBA article, the model may fabricate content. TKCG does not sufficiently handle such unrelated input, representing an important area for future work. Regarding keyword extraction, TKCG relies on pre-defined seed words per topic being provided. An interesting extension could automatically generate synonyms for topics using a lexical database.

References

[1] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[2] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[3] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

[4] Yusen Zhang, Yang Liu, Ziyi Yang, Yuwei Fang, Yulong Chen, Dragomir Radev, Chenguang Zhu, Michael Zeng, and Rui Zhang. Macsum: Controllable summarization with mixed attributes. *Transactions of the Association for Computational Linguistics*, 11:787–803, 2023.

[5] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[6] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[7] Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. CTRLsum: Towards generic controllable text summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5879–5915, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[8] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[9] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[10] Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. Understanding in-context learning via supportive pretraining data. *arXiv preprint arXiv:2306.15091*, 2023.

[11] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.