Personalized Language Model Tailoring: Two-Phase Approach for Customized Content Generation

Samveg Shah
samvegvipuls@umass.edu
University Of Massachusetts Amherst
Amherst, Massachusetts, USA

Dhruvin Gandhi
dhruvinrakes@umass.edu
University Of Massachusetts Amherst
Amherst, Massachusetts, USA

ABSTRACT

This study emphasises the importance of personalising Language Models (LLMs) to accommodate individual preferences. We suggest a two-phase approach that makes use of two datasets from the LAMP [1] repository: Personalised News Headline Generation and Personalised Scholarly Title Generation. To guide future interactions, we first pull the top K documents using generative retrieval from the user's history. The next step is to select the most important chunks from these documents to improve prompts so that the LLM receives clear inputs. This method seeks to efficiently capture language style and semantics so that the LLM can produce customised headlines. We evaluate the effect of this two-phase method against a single-phase approach through a comparative study, showing that our method performs better because of the LLM's improved comprehension of language nuances from relevant content extraction.

ACM Reference Format:
Samveg Shah and Dhruvin Gandhi. 2018. Personalized Language Model Tailoring: Two-Phase Approach for Customized Content Generation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The progression of Language Models (LLMs) has been notably shaped by the concept of personalization, marking a response to the growing demand for tailoring content generation according to individual user preferences and contextual factors. The emphasis on personalized content generation represents a crucial aspect of research and development, as Language Models have the potential to enhance user satisfaction and engagement by understanding and adjusting to the specific needs of each user.

The incorporation of personalization in Language Models is driven by the recognition that a one-size-fits-all approach to content generation is no longer sufficient in the evolving landscape of digital communication. Users today have diverse preferences, contexts, and requirements, and a personalized approach allows Language Models to cater to these individual variations.

We begin by utilising generative retrieval to retrieve relevant documents from the user's past profile activity. This procedure is important because retrieval functions in an unsupervised way, which means that it finds and retrieves pertinent documents on its own without the need for labels or instructions. Because the retrieval process is unsupervised, it can adjust to a variety of user contexts and preferences, giving the system flexibility in gathering a broad range of pertinent content.

We introduce the generation of 'm' different hypothesis versions of the input text in order to improve the retrieval process' efficacy. These different theories help to broaden the scope of the search results, allowing the system to take into account different ways that the user may have phrased their query. This variety is necessary to gather a wide range of potentially pertinent documents and to take into account the various ways that users may express their information needs.

Following the generation of hypotheses, we employ embedding techniques to represent both the original query and the hypotheses as numerical vectors. These embeddings capture the semantic relationships within the text, facilitating a more nuanced understanding of content. Subsequently, a dense search is performed in the list of user profile documents using these embeddings. The dense search method, guided by embeddings, aims to identify documents with similar semantic content. This step ensures a sophisticated and context-aware search, contributing to the extraction of the most relevant documents from the user's profile history. These meticulously selected documents form the foundation for subsequent stages, allowing language models to better comprehend the user's unique language nuances and preferences.

From this larger content pool, we extract the top 'k' relevant documents using a dense retrieval model. These documents are essential inputs for the next stage because they lay the groundwork for the LLM to understand language nuances and styles unique to each user. A keystone of our approach is Phase 2, which focuses on document content reduction. Not every line makes a significant contribution to title generation, even though the retrieved documents provide insight into the user's writing style. As a result, the emphasis is on identifying and picking relevant passages or sentences from historical data that clearly support the thesis statement or title of the research paper. This technique makes sure the LLM gets brief but insightful inputs, which improves its capacity to identify pertinent data and produce customised titles. These developed techniques work well because they take a targeted approach to using historical data. Through careful selection and integration of only the most pertinent data into the LLM's query, we maximise the learning process of the model, allowing it to capture complex relationships between document content and title generation while highlighting the most important features of the data.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

2 LITERATURE SURVEY

Before starting off with our research undertaking we identified 2 research papers which were in our domain and studied them thoroughly to know what kind of research have been already performed and how we can outperform it

This research paper introduces Hypothetical Document Embeddings (HyDE) [2] to address the challenge of zero-shot dense retrieval without explicit relevance labels. HyDE utilizes instruction-following language models, like InstructGPT, to generate hypothetical documents based on query instructions, capturing relevance patterns. These generated documents, though unreal and potentially containing false details, are then encoded by an unsupervised contrastive encoder (e.g., Contriever). The experiments show that HyDE outperforms existing unsupervised dense retrievers across various tasks and languages. However, a drawback is the potential inclusion of factual errors in the generated documents, mitigated by the encoder filtering out incorrect details during retrieval.

This research paper introduces LaMP [3], a novel benchmark for training and assessing language models for customized outputs, and explores the emerging field of personalised natural language understanding and generation. LaMP emphasises personalised outputs by incorporating diverse language tasks and multiple user profile entries into its seven personalised tasks, which span classification and text generation. In order to create personalised prompts, the paper presents a retrieval augmentation technique that builds on user profiles to retrieve relevant items. According to experimental findings, language models that use profile augmentation perform better than those that don't have profile information in all tasks.Retrieval techniques like Contriever and BM25, in particular, show promise in improving model performance through the selection and integration of pertinent user profile data, highlighting the crucial role that meticulous profile selection plays in model prompts for customised outputs

3 METHODOLOGY

3.1 About the dataset

Two key datasets—Personalized News Headline Generation and Personalised Scholarly Title Generation—are the focus of our study. In the former, the output uses the user's previous articles, including their titles, to generate news article titles from the articles that are provided. On the other hand, the latter dataset uses the user's history of abstracts and titles to generate academic article titles based on abstracts. Because generation tasks are more complex than classification tasks, we purposefully selected them for this reason. We designed our study to take advantage of large user histories, which means that we must retrieve the top K documents and extract relevant data chunks. We made this intentional decision in order to address more complex language modeling problems.

3.2 Proposed Methodology

We have divided our entire architecture into two phases. The first phase is responsible for identifying the top K documents and providing it as an input to the next phase which identifies relevant chunks of data and finally appends it to the query of the LLM. The following section will be delving deeper into both the phases:

- `Generative task`: We utilized Palm, a language model renowned for its extensive training. Google AI created the 540 billion parameter PaLM (Pathways Language Model), a large language model based on transformers. In our approach, we feed the abstract (for scholarly title generation) or new content (for title generation) as the input query. This query prompts the model to generate articles or abstracts which could be of the similar context or similar semantic meaning. Since the retrieval from user profile is completely unsupervised we used this generative task to generate documents what could have been similar to input. This creates 'm' hypothetical versions of the content, maintaining the essence of the original query's meaning and style. This strategy broadens the comparative space, enhancing the efficiency of retrieving the top K documents for effective analysis.
- `Encoding`: We did a comparative analysis between 2 different encoding methodologies that is DistilBERT and Contriever. DistilBERT, a condensed form of the BERT model, compresses the original architecture while maintaining a significant amount of its contextual understanding capabilities, providing an effective yet potent embedding scheme. It is an excellent option for a variety of NLP tasks because it performs well in token-level embeddings and offers a sophisticated representation of words in context. Conversely, Contriever functions as a dense retrieval model that is intended to extract relevant and targeted data from large datasets. Its emphasis on obtaining contextually relevant data for a given query greatly improves the specificity and precision of information extracted, enabling more specialized and refined outputs in tasks related to language generation and understanding. The reason we decided to use these 2 encoding schemes is because DistilBERT, known for its computational efficiency and decent contextual embeddings, was chosen for its resource-friendly performance. Contriever, excelling in dense retrievals and nuanced context capture, complemented DistilBERT to enhance nuanced language understanding. We wanted to study of which performs better on the given dataset.
- `Index creation and dense search`: In our method, we created an index for our embeddings by utilising FAISS, a high-performance library for similarity search and indexing of dense vectors. FAISS's proficiency in efficiently managing high-dimensional data was essential for our dense search procedures. We build index for every user profile by encoding the documents with Contriver and then we map the index text to its corresponding IDs. By building this index, we were able to streamline the dense retrieval process and enable quick and efficient searches for similarity across a wide range of documents or embeddings. After generation of the hypothesis documents we encode it with Contriver and then use the embedding to do a dense search in the index. This improved our personalised language model framework's capacity to produce customised outputs based on user-specific data by enabling us to quickly and effectively retrieve pertinent data for prompt augmentation.

3.2.1 *Phase 2: Extracting chunks of data.*

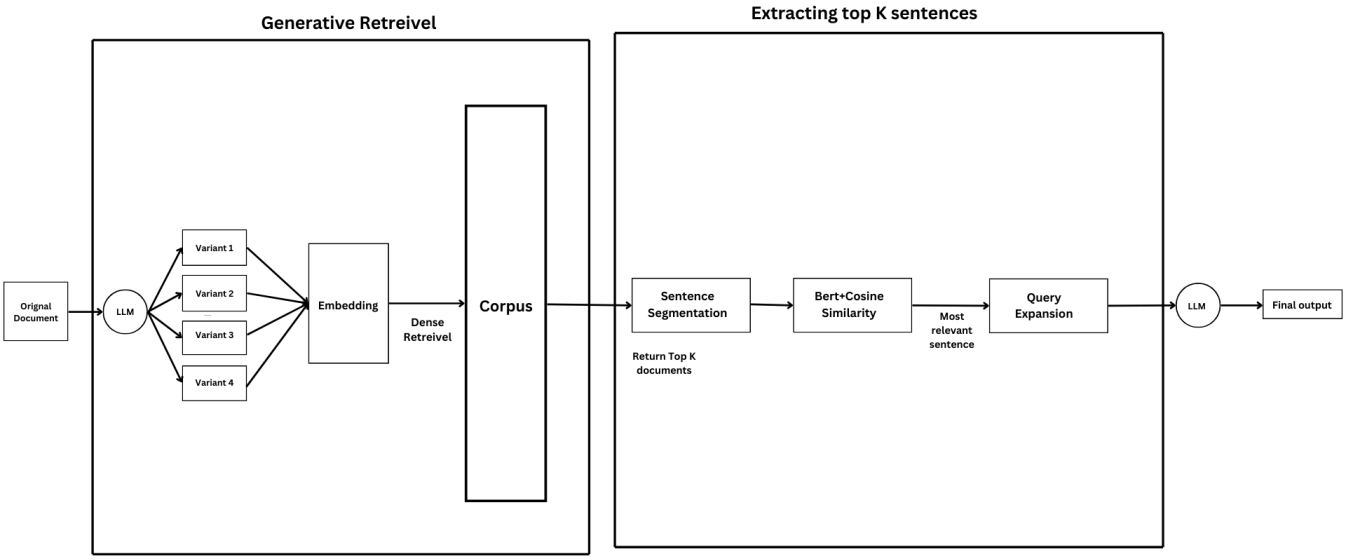

Figure 1: Architecture

- For the Language Model (LLM) to identify the most relevant information, it is essential to extract specific chunks from a user's historical data while maintaining both language style and semantic context. The LLM is better able to comprehend the user's writing style and the important passages in their articles or abstracts by breaking down the historical content into these bite-sized pieces. This procedure simplifies the model's understanding of which components are most important for producing correct titles or content. In order to improve the personalised generation process, the LLM can better mimic the user's style and generate outputs that are more contextually relevant when the semantic and linguistic essence is preserved. This helps the LLM focus on the important aspects of the user's historical data.
- `Sentence Segmentation and encoding`: After we have the top K documents from the first stage, we divide each document into sentences. In the next iteration, we apply the bert-base-uncased model to every sentence in the document. We also embed the document's title at the same time. Sentences are the main focus of the PALM language model, which is the reason for choosing the sentence as the window size. Sentences are better at holding the writing style and contextual details, which improves overall coherence when compared to shorter chunk sizes. The power of bert-base-uncased rests in its ability to produce sentence embeddings of superior quality. Through the use of this model, we are able to extract contextual information, semantic relationships, and minute details from the document. This method guarantees that the embeddings capture the main ideas of the content, which makes them useful for further processing and analysis.
- `Similarity calculation`: During the procedure, we compared the embeddings of the `bert-base-uncased-embedded` title with each sentence that corresponded to it in the document using cosine similarity. A metric called cosine similarity is used to calculate the cosine of the angle that separates two non-zero vectors. It ranges from 0 (orthogonal) to 1 (totally similar), with -1 denoting total dissimilarity.
  The mathematical equation for cosine similarity between two vectors A and B is given by:
  The formula for cosine similarity is represented as:

$$\text{cosine\_similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

  Here, $A \cdot B$ represents the dot product of vectors A and B, while $\|A\|$ and $\|B\|$ denote the Euclidean norms of vectors A and B, respectively. Since cosine similarity offers a normalized measure of similarity, it is useful for comparing vectors of different lengths and is thus appropriate when choosing the most similar line. Its ability to withstand changes in document length guarantees that the similarity metric is determined by the semantic content and not by text length. Because of this, cosine similarity is a good option for tasks

where it's important to comprehend the semantic similarity between text segments, like finding the most pertinent sentence in a document.
- `Feeding it to the LLM`: We have opted to integrate PALM throughout this process, taking advantage of its computational prowess. Using query expansion entails adding updated documents and titles to the original query. This method gives the Language Model access to a more comprehensive context, including writing style, language subtleties, and historical data about the user. The model produces results that are more in line with the user's preferences and contextual requirements by incorporating these elements. We receive the title as the output which is our final output

4 RESULTS

In our research undertaking, we conducted a comparative study to assess the effectiveness of different large language models (LLMs). We specifically examined two LLMs: FLAN T-5 and PALM. FLAN T-5 is a variant of the T-5 (Text-to-Text Transfer Transformer) that has been fine-tuned on a wide range of language tasks to enhance its adaptability and performance, while PALM (Pathways Language Model) is known for its robust and scalable architecture, designed to handle a diverse set of language processing tasks efficiently. In the second phase of our study, we replaced the LLM with FLAN T 5 and Palm, aiming to gauge the capacity of the LLMs in capturing user history and generating tweets. Our observations revealed that PALM outperformed FLAN T-5 in this context. The detailed results of our comparative analysis are encapsulated in the table below.

	FLAN T-5	PALM
rouge-1	0.174	0.258
rouge-L	0.160	0.235

Table 1: Personalized News Headline Generation

	FLAN T-5	PALM
rouge-1	0.381	0.448
rouge-L	0.354	0.405

Table 2: Personalised Scholarly Title Generation

Our second comparative study aimed to assess the impact of extracting the most pertinent sentence from a user's historical data, using it exclusively for prompt expansion in the LLM. We contrasted this approach with supplying the entire user history to the LLM for result generation. In the latter method, we directly fed the phase 1 results as input to the final LLM for generating outcomes.

Our experiments revealed that extracting the most relevant sentence significantly influenced the accuracy obtained. Not all lines within a user's history contribute equally to determining the headline. To maintain the user's semantic style and ensure the LLM focuses solely on the pertinent content, it's crucial to extract only the single line relevant to the headline. Thus, we adopted this strategy.

Throughout both phase 1 and phase 2, we utilized the PALM LLM. This decision stemmed from our initial comparative study, where PALM consistently demonstrated superior results.

	Without sentence extraction	With Sentence extraction
rouge-1	0.172	0.258
rouge-L	0.135	0.235

Table 3: Personalized News Headline Generation

	Without sentence extraction	With Sentence extraction
rouge-1	0.352	0.448
rouge-L	0.316	0.405

Table 4: Personalised Scholarly Title Generation

5 CONCLUSION

The study highlights the significance of tailoring Language Models (LLMs) to individual preferences, employing a two-phase approach sourced from datasets within the LAMP repository: Personalised News Headline Generation and Personalised Scholarly Title Generation. The methodology involves extracting top K documents from the user's history and identifying essential sentences to enhance LLM prompts, ensuring a focused grasp of user language nuances for generating customised headlines. Comparative analyses affirm the superiority of the two-phase method over single-phase approaches, particularly emphasizing the impact of extracting the most relevant sentence to boost accuracy. Consistent use of the PALM LLM across both phases was justified by its superior performance, and experiments underscore PALM's efficacy in capturing user history for targeted outputs. The presented tables demonstrate the tangible impact of sentence extraction in headline generation across tasks, affirming its pivotal role in improving accuracy. Overall, the research advocates a refined approach to user data integration for improved LLM customisation and performance enhancement. The scope for improvement is to test over a larger dataset. Due to system limitations, the testing is performed only on 100 records for every experiment. Another improvement could be using a better LLM like GPT 4 whose API is paid.

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

REFERENCES

[1] Salemi A, Mysore S, Bendersky M, Zamani H. LaMP: When Large Language Models Meet Personalization. arXiv preprint arXiv:2304.11406. 2023 Apr 22.
[2]Gao L, Ma X, Lin J, Callan J. Precise zero-shot dense retrieval without relevance labels. arXiv preprint arXiv:2212.10496. 2022 Dec 20.
[3]Qian H, Dou Z, Zhu Y, Ma Y, Wen JR. Learning implicit user profile for personalized retrieval-based chatbot. Inproceedings of the 30th ACM international conference on Information Knowledge Management 2021 Oct 26 (pp. 1467-1477).