2024 IEEE Spoken Language Technology Workshop (SLT) | 979-8-3503-9225-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/SLT61566.2024.10832316

DOMAIN ADAPTION AND UNIFIED KNOWLEDGE BASE MOTIVATE BETTER RETRIEVAL MODELS IN DIALOG SYSTEMS WITH RAG

*Huadong Lin[1], Yirong Chen[1], Wenyu Tao[1], Mingyu Chen[1], Xiangmin Xu[1,2], Xiaofen Xing[1]**

[1]South China University of Technology, Guangzhou, China
[2]Pazhou Lab, Guangzhou, China

ABSTRACT

Retrieval augmented generation (RAG) has emerged as a paradigm to address problems like hallucination in dialog systems based on large language model (LLM). Retrieval model is a key component in RAG framework for recalling relevant information. This paper describes our solution for FutureDial-RAG Challenge Track 1. We identify two primary challenges in this track: domain specificity and heterogeneity of knowledge base. To address the two challenges, we first adopt continual pre-training of a pre-trained retrieval model on both labeled and unlabeled data for domain adaption. Subsequently, we modify and expand the knowledge base, ensuring that each piece of knowledge is uniformly structured in a question-answer (QA) format. Finally, we construct negative samples based on the labeled data and the unified knowledge base, and fine-tune the retrieval model using contrastive learning. Our solution achieves a score of 2.023 on the dev set, which significantly outperforms the baseline.

Index Terms— Domain Adaption, Unified Knowledge Base, Dialog System, RAG

1. INTRODUCTION

The intelligent dialog system is an important research branch in the field of artificial intelligence. In recent years, with advancements in algorithms and the growth of data, significant progress has been made in building dialog systems [1, 2]. At the end of 2022, the emergence of ChatGPT [3, 4] made the generative large language model (LLM) a research hotspot. Open-domain dialog systems driven by LLM are impressive in terms of response generation quality. However, in specific application scenarios such as customer service systems, LLM-based dialog systems may encounter issues like hallucination. Retrieval augmented generation (RAG) [5, 6] technology helps address this problem by retrieving external knowledge, enabling LLM to generate more accurate responses. The architecture of a basic RAG-based dialog system is shown in Fig. 1.

*Corresponding author. Email: xfxing@scut.edu.cn

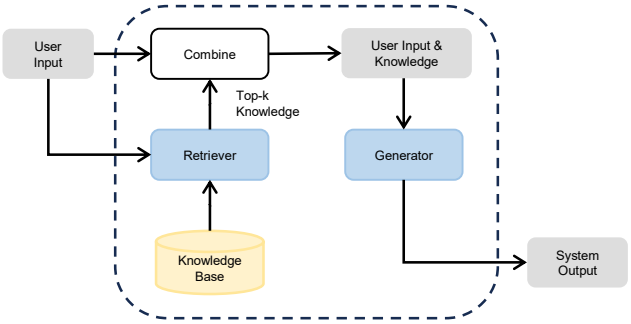

Fig. 1. An illustration of a basic RAG-based dialog system. User input is combined with retrieved top-k knowledge from a knowledge base to generate informed outputs.

To facilitate research on RAG-based dialog systems, the FutureDial-RAG Challenge released a new dataset called MobileCS2 (Mobile Customer Service) [7], which contains a lot of manually annotated information. The challenge requires participants to utilize MobileCS2 to build a more robust and more powerful RAG-based customer service dialog system. There are two tracks: 1) Track 1 (Information retrieval based on knowledge bases and dialog context) aims to build the retrieval model for the dialog system. 2) Track 2 (Dialog systems with retrieval augmented generation) aims to build a retrieval-augmented dialog system in the customer service scenario.

In this paper, we present our solution for Track 1. We analyze the data of MobileCS2 and identify two potential difficulties to overcome in Track 1: domain specificity and heterogeneity of knowledge base. To address these difficulties, we first construct a large number of training samples using both labeled and unlabeled data, and continue pre-training the retrieval model on these samples to help it learn domain-specific text features. Second, we modify the knowledge base. We use LLMs to generate multiple questions related to product knowledge and manually define fixed questions for user-related knowledge. This unifies all knowledge into a question-answer (QA) format while expanding its scale,

Table 1. The statistics of MobileCS2 dataset.

Metric	Labeled		Unlabeled
	Train	Dev	
Total # dialogs	1926	412	3598
Total # turns	16119	3246	32218
Total # tokens	1327912	257734	2641498
Avg # turns per dialog	8.37	7.88	8.95
Avg # tokens per turn	82.38	79.40	81.99

Table 2. The statistics of knowledge base.

Metric	KB_{FAQ}	$KB_{product}$	KB_{user}
Total # knowledge	2837	754	1234
Avg # knowledge per dialog	-	-	0.53

increasing the number of positive samples available for training. Finally, we construct negative samples based on the labeled data and unified knowledge base, and fine-tune the domain-adapted retrieval model using contrastive learning to further improve its retrieval performance. Our solution finally achieves a score of 2.023 on the dev set.

We summarize the main contributions of our solution as follows:

- We fully utilize both labeled and unlabeled data, employing continual pre-training to adapt the model to the specific domain of customer service.
- We modify the original knowledge base, unifying the format of all knowledge to address the heterogeneity of knowledge base.
- We construct negative samples and further optimize the model using contrastive learning.

2. BACKGROUND

2.1. Data description

The dataset MobileCS2 is derived from the China Mobile real-world conversational scenarios and comprises dialog logs between customers and customer service representatives. MobileCS2 is available in both Chinese and English versions. Each version contains labeled data and unlabeled data, where the labeled data includes train set and dev set. The full data statistics are shown in Table 1. For labeled data, there are several additional annotated data:

- **Api_query.** Given a dialog, the annotators are required to identify the intent (annotated as Api_query) of the customer service representatives in each turn of the dialog. Api_query can be classified into the following categories: *1) QA, 2) NULL, 3) Search for products information, 4) Search for user information, 5) Search for other information, 6) Cancel business, 7) Handle business, 8) Verify identity*.
- **Api_result.** For some dialog turns where Api_query is categorized as an API-Inquiry type, for example, when Api_query is *"Search for products information"* or *"Search for user information"* or *"Search for other information"*, the annotators are required to additionally label the corresponding query result (annotated as Api_result).

By aggregating the annotated data, a knowledge base can be constructed for retrieval. Specifically, for the turns where the Api_query is *"QA"*, the utterance of customer service is used as the answer, and the utterance of customer concatenated after the previous two turns of dialog context is used as the question. These QA pairs are collected to construct the FAQ (Frequently Asked Questions) sub-knowledge base (KB_{FAQ}). In addition, the challenge organizers provided 39 pre-built FAQs, which were also part of the final KB_{FAQ}. For turns where the Api_query is *"Search for products information"*, the corresponding Api_result is collected and a product sub-knowledge base ($KB_{product}$) is constructed. For turns where the Api_query is *"Search for user information"*, the corresponding Api_result is collected and a user sub-knowledge base (KB_{user}) is constructed. KB_{user} is unique for each dialog while KB_{FAQ} and $KB_{product}$ is fixed across the entire dataset. The statistics of knowledge base are shown in Table 2. For the dialog X, the knowledge base KB_X can be denoted as

$$KB_X \triangleq KB_{FAQ} \cup KB_{product} \cup KB_{user}. \quad (1)$$

2.2. Task description

The FutureDial-RAG challenge consists of two tracks. We focus on Track 1: Information retrieval based on knowledge bases and dialog context. Given a dialog X, the task of Track 1 is to retrieve knowledge pieces most relevant to the dialog context c_t of current turn t from the knowledge base KB_X, so that the customer service dialog system can generate more accurate and helpful responses. The definition of dialog context c_t is

$$c_t \triangleq u_1 \oplus r_1 \oplus \cdots u_{t-1} \oplus r_{t-1} \oplus u_t, \quad (2)$$

where u_t represents the utterance of customer, r_t represents the utterance of customer service representative, $\oplus$ means sequence concatenation.

Therefore, the participants need to train a high-performance retrieval model based on the existing data. When finally evaluating on the test set, those turns where Api_query is not

categorized as a API-inquiry type do not participate in the calculation of the metric. Track 1 uses the commonly used recall metrics to assess the retrieval model, and the final score is

$$score = recall@1 + recall@5 + recall@20, \quad (3)$$

where $recall@k$ represents the recall metric when retrieving the $top\text{-}k$ knowledge pieces.

3. METHODOLOGY

Based on the analysis of the task and data, we identify two primary challenges to address in Track 1:

- **Domain specificity.** The MobileCS2 dataset consists of consultation dialog logs between China Mobile customer service representatives and customers, containing a significant amount of domain-specific product information and technical terms. However, most retrieval models in the open-source community are trained on general-purpose corpora, with texts from the telecommunications customer service scenarios representing only a tiny fraction of the training data. Consequently, compared to general text, these retrieval models have weaker representation capabilities for texts in the telecommunications customer service domain, further limiting their ability to retrieve relevant information effectively.

- **Heterogeneity of knowledge base.** The knowledge base corresponding to each dialog consists of a fixed FAQ sub-knowledge base (KB_{FAQ}), a fixed product sub-knowledge base ($KB_{product}$), and a user sub-knowledge base (KB_{user}) that dynamically changes according to the dialog ID. The knowledge in different sub-knowledge bases varies in form. In KB_{FAQ}, knowledge is stored in the form of QA pairs. In $KB_{product}$, each piece of knowledge is an introduction to a business product. In KB_{user}, knowledge typically pertains to user package information and product subscription details. Additionally, there are differences in the scale of each sub-knowledge base. As shown in Table 2, the fixed part of the knowledge base includes 2837 entries in KB_{FAQ}, significantly more than the 754 entries in $KB_{product}$. The misalignment in knowledge form and scale results in heterogeneity within the knowledge base, thereby affecting the performance of retrieval model.

To address the aforementioned challenges, we propose a solution comprising the following steps: **1) domain adaptation**, **2) unified knowledge base**, and **3) contrastive learning**. The overall workflow of the solution is illustrated in Fig. 2.

3.1. Domain adaptation

To adapt the general retrieval model to the domain of telecommunications customer service, we conduct continual pre-training on a large corpus of domain-specific data. We leverage the approach presented in RetroMAE [8], which is a retrieval oriented pre-training paradigm based on Masked Auto-Encoder (MAE). The masked text is encoded into its embedding, from which the full text is recovered on top of a light-weight decoder. The optimization objective can be expressed as

$$\min \sum_{x \in X} -\log \mathrm{Dec}\left(x \mid \mathbf{e}_{\tilde{X}}\right), \mathbf{e}_{\tilde{X}} \leftarrow \mathrm{Enc}(\tilde{X}), \quad (4)$$

where Enc and Dec represent the encoding and decoding operations, X and $\tilde{X}$ represent the full and masked text.

We utilize dialog logs from both labeled and unlabeled data, as well as all knowledge from the knowledge base, to construct the corpus for continual pre-training. Each sample from the dialog logs represents a single turn of dialog in the format *"customer: {the utterance of customer} customer Service: {the utterance of customer service}"*. Samples from the knowledge base are used without modification, retaining their original textual form. Finally, we obtained 53,571 samples for continual pre-training.

3.2. Unified knowledge base

To eliminate heterogeneity in the knowledge base, we unify the knowledge into a QA format. The QA format provides more comprehensive information, with the question part aligning the knowledge more closely with the query, thereby increasing the likelihood of relevant knowledge being recalled.

For KB_{FAQ}, the knowledge is already in the form of QA, so no transformation is needed for this part. For $KB_{product}$, we leverage the capabilities of LLM[1] to generate questions for each piece of knowledge. Specifically, for a given product knowledge k, we sample a dialog turn where the Api_result is k and concatenate it with the previous two dialog turns to obtain a reference dialog context $dial\_context$ corresponding to k. Based on k and $dial\_context$, we instruct the LLM to generate five questions related to k that customers might ask, resulting in five QA pairs as new knowledge. Additionally, the scale of $KB_{product}$ will be expanded to five times its original size, bringing it to a level comparable to KB_{FAQ}, thus addressing the issue of imbalance in the size of sub-knowledge bases. For KB_{user}, since it dynamically changes with the dialog, generating questions in the same manner would require instruct the LLM for each dialog during evaluation, which would impact the response speed of the retrieval system. Therefore, we manually define a fixed

[1]We use Qwen-Max [9] to generate questions.

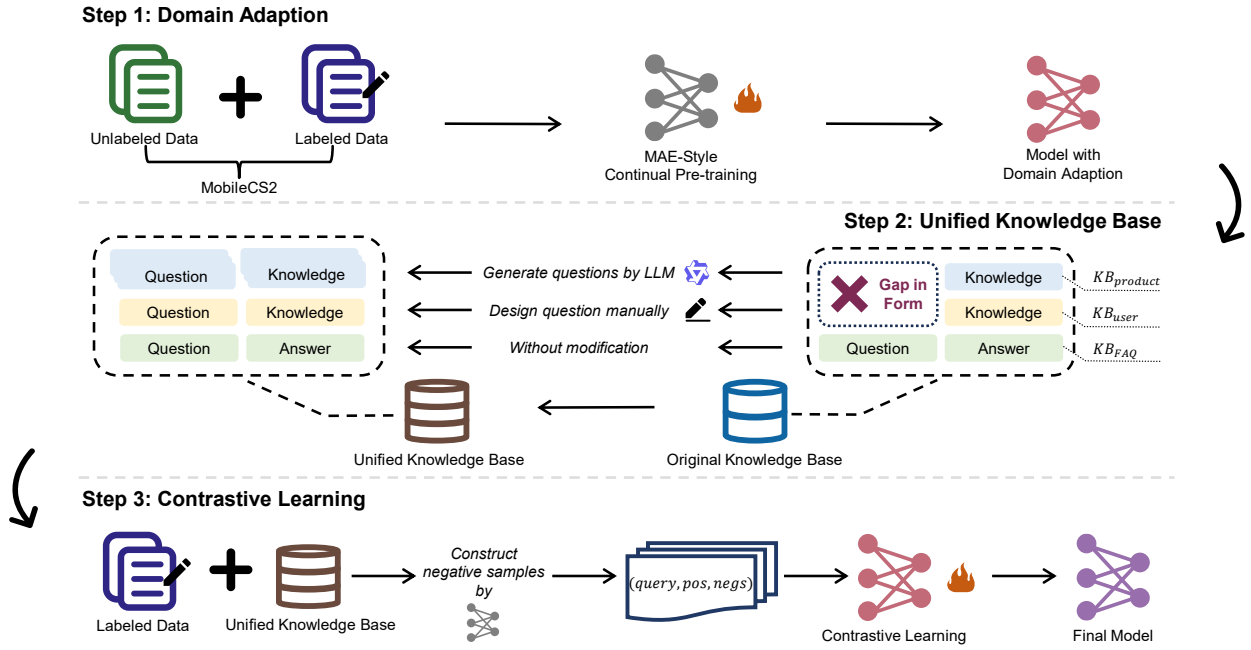

Fig. 2. The overall workflow of the proposed solution. Through domain adaption, unified knowledge base and contrastive learning, the final model performs well on the dev set.

question *"Query my personal business information for me."* for the knowledge from KB_{user}.

During the evaluation of Track 1, the retrieval system needs to output the ID of the recalled knowledge, which is fixed for each piece of knowledge. Since we have expanded $KB_{product}$, we assign the original ID to homologous knowledge within $KB_{product}$. We then deduplicate the recalled ids and select the top-20. This approach prevents disruption to the original knowledge base indexing structure after the expansion.

3.3. Contrastive learning

Based on the train set from the labeled data and the expanded unified knowledge base, we construct training data containing negative samples. Each training sample is a triplet of the form $(query, pos, negs)$. We consider the turn t where Api_query is categorized as API-Inquiry type and use a portion of c_t that $c'_t = u_{t-2} \oplus r_{t-2} \oplus u_{t-1} \oplus r_{t-1} \oplus u_t$ as the $query$. The QA-formatted knowledge corresponding to the Api_result of these turns in the unified knowledge base is used as pos, representing a positive example for the query. We employ a general retrieval model to compute the similarity scores between the query and all knowledge entries in the unified knowledge base and sample seven knowledge entries within a specific score range to form negative samples $negs$. Too difficult negative samples are detrimental to model training. They may confuse the model during training and affect its final performance [10]. Therefore, we choose to sample negative samples from the score range between top-1000 and top-2000. Ultimately, we obtain 8,434 samples for contrastive learning.

With the constructed training data, we use contrastive learning to fine-tune the retrieval model that has undergone continual pre-training. This aims to further enhance retrieval performance. The optimization objective of contrastive learning can be expressed as

$$\min \sum_{(q,p,N)} -\log \frac{e^{\langle \mathbf{e}_q, \mathbf{e}_p \rangle / \tau}}{e^{\langle \mathbf{e}_q, \mathbf{e}_p \rangle / \tau} + \sum_{n \in N} e^{\langle \mathbf{e}_q, \mathbf{e}_n \rangle / \tau}}, \quad (5)$$

where (q, p, N) represents $(query, pos, negs)$, τ represents temperature coefficient.

4. EXPERIMENT

4.1. Baseline

The organizers of the challenge provided a baseline for Track 1 [7]. The baseline trains the retrieval model based on the original annotated data and knowledge base, without manually constructing negative samples. To train the retrieval model $p_\eta(h_t|c_t, KB_X)$ to get the relevant knowledge h_t, the baseline considers each knowledge piece z_i $(i = 1, 2, \cdots, K)$

Table 3. The evaluation results on the dev set.

Methods	Recall@1	Recall@5	Recall@20	Score
Baseline	0.225	0.387	0.573	1.185
CL	0.453	0.578	0.673	1.704
DA+CL	0.488	0.577	0.701	1.767
DA+UK+CL	**0.528**	**0.694**	**0.800**	**2.023**

in KB_X and model the retrieval distribution of $p_\eta\left(z_i|c_t\right)$:

$$p_\eta(z_i \mid c_t) \propto \exp\left(\text{Encoder}_p(z_i)^\top \text{Encoder}_c(c_t)\right). \quad (6)$$

The probability is optimized with the standard cross entropy loss, with the positive pieces $z \in Z_+$ labeled in the dataset:

$$\mathcal{L}_{\text{ret}} = -\frac{1}{|Z_+|} \sum_{z \in Z_+} \log \frac{p_\eta(z \mid c_t)}{p_\eta(z \mid c_t) + \sum_{i=1, z_i \neq z}^{K} p_\eta(z_i \mid c_t)}. \quad (7)$$

The baseline uses $c_t = u_{t-1} \oplus r_{t-1} \oplus u_t$ and uses the retrieval model bge-large-zh-v1.5[2] [11] to initialize Encoder_p and Encoder_c. In particular, the baseline only optimizes the parameters of Encoder_c, and the parameters of Encoder_p are frozen during training.

4.2. Implementation details

We conduct our solution starting from a pre-trained retrieval model named stella-large-zh-v3-1792d[3]. This model achieves a good ranking on the C-MTEB leaderboard[4] [12]. It is based on the BERT [13] architecture and includes an optional linear layer that maps the original embedding dimension from 1024 to 1792, providing a certain performance improvement. However, to balance performance and efficiency, we do not use this additional linear layer in our implementation.

In the continual pre-training stage, the learning rate is set to 2e-5, the batch size is set to 64, and the maximum length of the input sequence is set to 512. The model is trained for 2 epochs.

In the contrastive learning stage, the learning rate is set to 1e-5, the batch size is set to 32, and the maximum length of the input sequence is set to 512. The model is trained for 20 epochs and the checkpoint with the highest score on the dev set is used to report the results. We use the Faiss[5] [14] library to build vector index of the knowledge base. It helps improving the retrieval efficiency during the evaluation process.

4.3. Results

Since the participants could not access the test set during the challenge, we conduct experiments on the dev set. All experiments use stella-large-zh-v3-1792d as the retrieval model. We perform three methods and calculate the scores on the dev set:

- **CL:** It does not perform domain adaptation and does not unify the knowledge base. Negative examples are constructed from the original knowledge base, and the model was fine-tuned using contrastive learning.
- **DA+CL:** It does not conduct knowledge base unification. The model was first adapted to the domain using continual pre-training, and then negative examples were constructed from the original knowledge base for contrastive learning.
- **DA+UK+CL:** It follows our proposed solution, implementing domain adaptation, knowledge base unification, and contrastive learning.

As shown in Table 3, comparing the baseline and CL results reveals that manually constructing negative examples can improve model performance. The results of CL, DA+CL, and DA+UK+CL demonstrate that both domain adaptation and unified knowledge base can enhance performance. Among them, the improvement brought by unified knowledge base is the most significant. Our proposed solution DA+UK+CL achieved the highest score on the dev set, far ahead of the baseline, which shows its effectiveness.

5. CONCLUSION

In this paper, we propose a solution for Track 1 of the FutureDial-RAG Challenge. We utilize both labeled and unlabeled data to continue pre-training the model. Additionally, we unify the knowledge base and expand its scale. Finally, we construct negative samples based on the unified knowledge base and train the model using a contrastive learning algorithm. Through these steps, we address the two major challenges in Track 1: domain specificity and heterogeneity of knowledge base. The experimental results demonstrate the effectiveness of our approach. Our solution achieves a score of 2.023 on the dev set, showing a significant improvement compared to the baseline.

6. ACKNOWLEDGMENTS

The work is supported by the Guangdong Provincial Key Laboratory of Human Digital Twin 2022B1212010004.

7. REFERENCES

[1] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić, "MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling," in *Proceedings of the 2018 Conference*

[2] https://huggingface.co/BAAI/bge-large-zh-v1.5

[3] https://huggingface.co/infgrad/stella-large-zh-v3-1792d

[4] https://huggingface.co/spaces/mteb/leaderboard

[5] https://ai.meta.com/tools/faiss/

on Empirical Methods in Natural Language Processing, Brussels, Belgium, Oct.-Nov. 2018, pp. 5016–5026, Association for Computational Linguistics.

[2] Zhijian Ou, Junlan Feng, Juanzi Li, Yakun Li, Hong Liu, Hao Peng, Yi Huang, and Jiangjiang Zhao, "A challenge on semi-supervised and reinforced task-oriented dialog systems," *arXiv preprint arXiv:2207.02657*, 2022.

[3] OpenAI, "Introducing chatgpt," `https://openai.com/blog/chatgpt`, 2022.

[4] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al., "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27730–27744, 2022.

[5] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang, "Retrieval augmented language model pre-training," in *Proceedings of the 37th International Conference on Machine Learning*. 13–18 Jul 2020, vol. 119 of *Proceedings of Machine Learning Research*, pp. 3929–3938, PMLR.

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[7] Yucheng Cai, Si Chen, Yi Huang, Junlan Feng, and Zhijian Ou, "The 2nd futuredial challenge: Dialog systems with retrieval augmented generation (futuredial-rag)," *arXiv preprint arXiv:2405.13084*, 2024.

[8] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao, "RetroMAE: Pre-training retrieval-oriented language models via masked auto-encoder," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 538–548, Association for Computational Linguistics.

[9] Qwen Team, "Notes on qwen-max-0428," `https://qwenlm.github.io/blog/qwen-max-0428/`, 2024.

[10] Inc. NetEase Youdao, "Bcembedding: Bilingual and crosslingual embedding for rag," `https://github.com/netease-youdao/BCEmbedding`, 2023.

[11] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof, "C-pack: Packaged resources to advance general chinese embedding," *arXiv preprint arXiv:2309.07597*, 2023.

[12] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers, "MTEB: Massive text embedding benchmark," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia, May 2023, pp. 2014–2037, Association for Computational Linguistics.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 4171–4186, Association for Computational Linguistics.

[14] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou, "The faiss library," *arXiv preprint arXiv:2401.08281*, 2024.