MeanCache: User-Centric Semantic Cache for Large Language Model Based Web Services

Waris Gill [1] Ali Anwar [2] Muhammad Ali Gulzar [1]

[1]Virginia Tech [2]University of Minnesota Twin Cities

Introduction

Services like ChatGPT revolutionize web search but incur high costs due to billions of parameters; caching repeated queries (31%) could reduce costs.

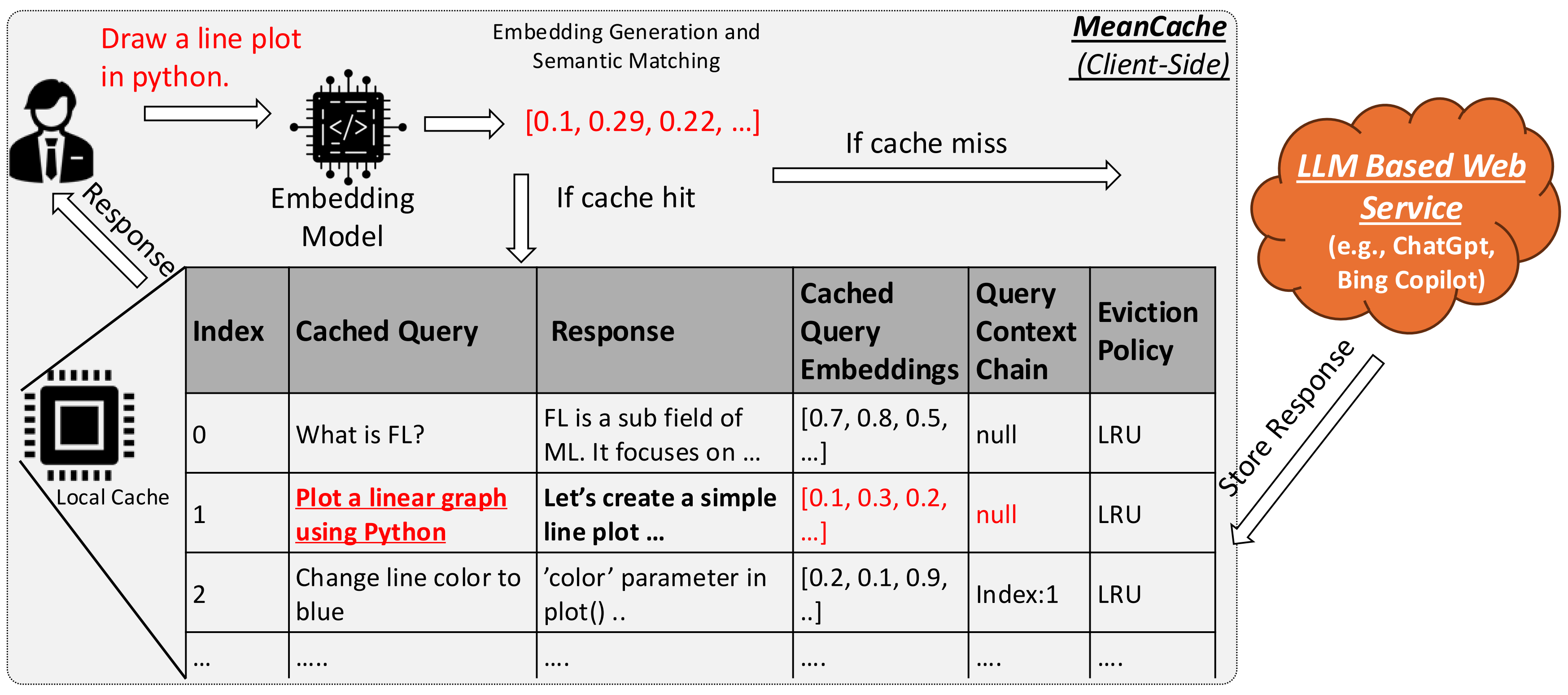
Problem Statement

- Existing caching methods rely on keyword matching, failing to capture semantic similarities.
- High false hit rates, especially with contextual queries (GPTCache [2]).
- Centralized caches raise privacy concerns.

MeanCache: Our Solution

- **User-Centric:** Local user's cache to preserve privacy and reduce server load.
- **Federated Learning:** Collaborative training of smaller embedding models without the need to share user data.
- **Context Awareness:** Encodes context chains to handle contextual queries.
- **Embedding Compression:** Applies PCA to reduce embedding dimensions, saving storage and speeding up searches.

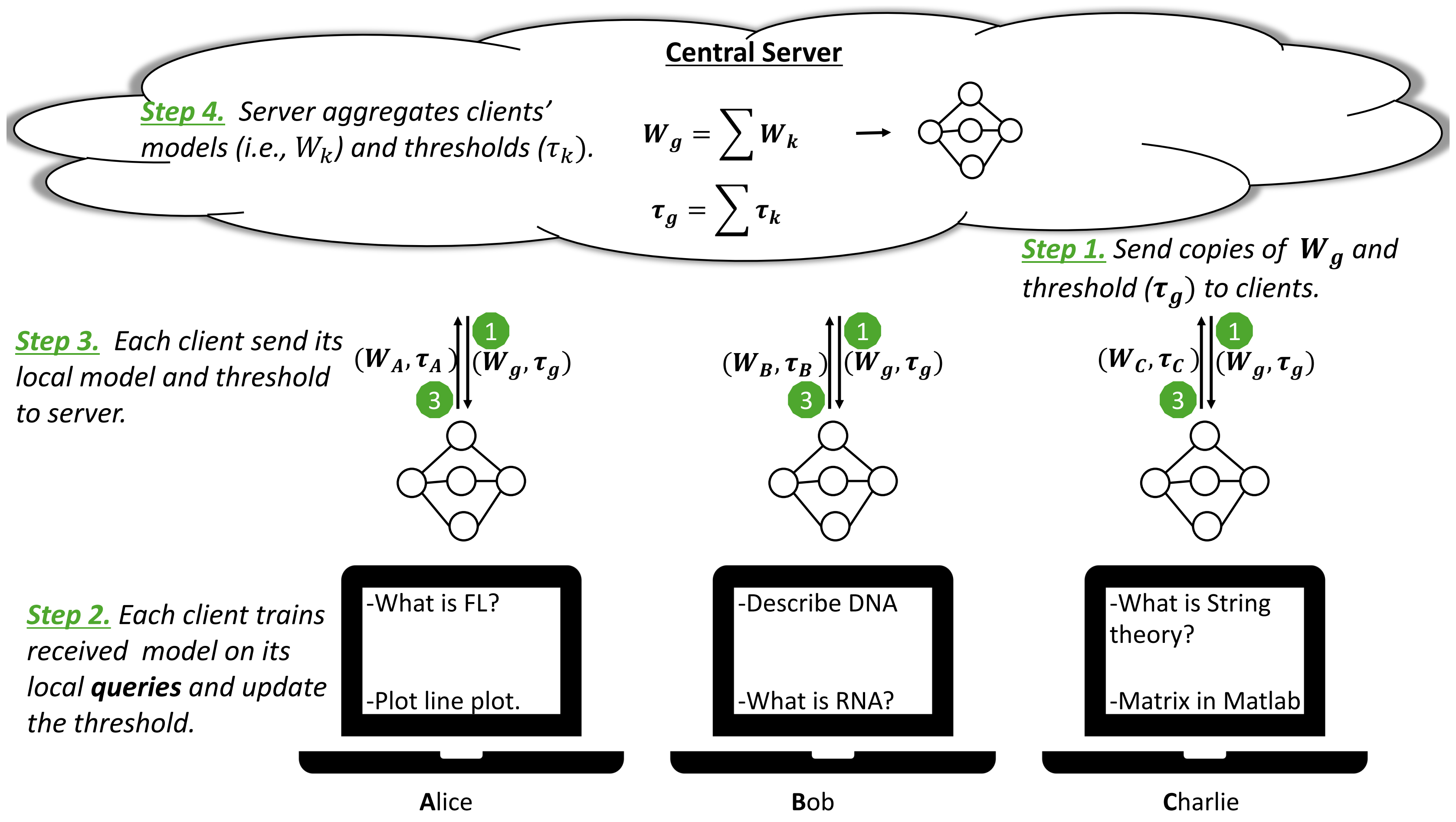
"Draw a line plot in Python" is a duplicate query. MeanCache [1] retrieves a semantically matched response from the user's cache (Index-1 Response).


Draw a line plot in python.
Embedding Generation and Semantic Matching
MeanCache (Client-Side)
Embedding Model
[0.1, 0.29, 0.22, ...]
If cache hit
If cache miss
Response
Local Cache
LLM Based Web Service (e.g., ChatGpt, Bing Copilot)
Store Response

Index	Cached Query	Response	Cached Query Embeddings	Query Context Chain	Eviction Policy
0	What is FL?	FL is a sub field of ML. It focuses on ...	[0.7, 0.8, 0.5, ...]	null	LRU
1	Plot a linear graph using Python	Let's create a simple line plot ...	[0.1, 0.3, 0.2, ...]	null	LRU
2	Change line color to blue	'color' parameter in plot() ..	[0.2, 0.1, 0.9, ..]	Index:1	LRU
...	.....	....	....	....	....

Use of Federated Learning and PCA in MeanCache

Privacy-Preserving Embedding Model FL Training in MeanCache.



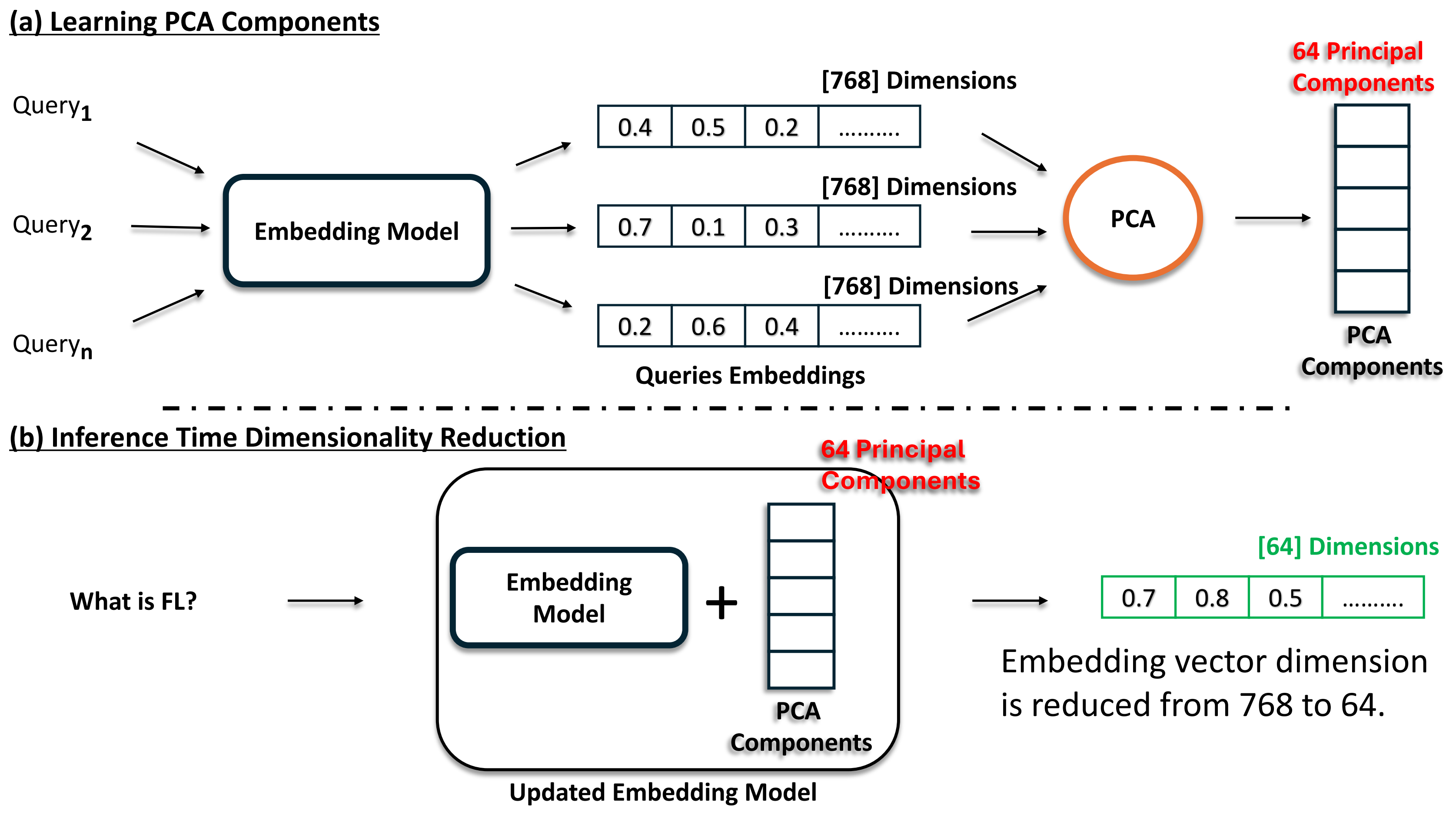
Step 4. Server aggregates clients' models (i.e., W_k) and thresholds (τ_k).

Step 1. Send copies of W_g and threshold (τ_g) to clients.

Step 3. Each client send its local model and threshold to server.

Step 2. Each client trains received model on its local **queries** and update the threshold.

Embeddings Compression using PCA in MeanCache.

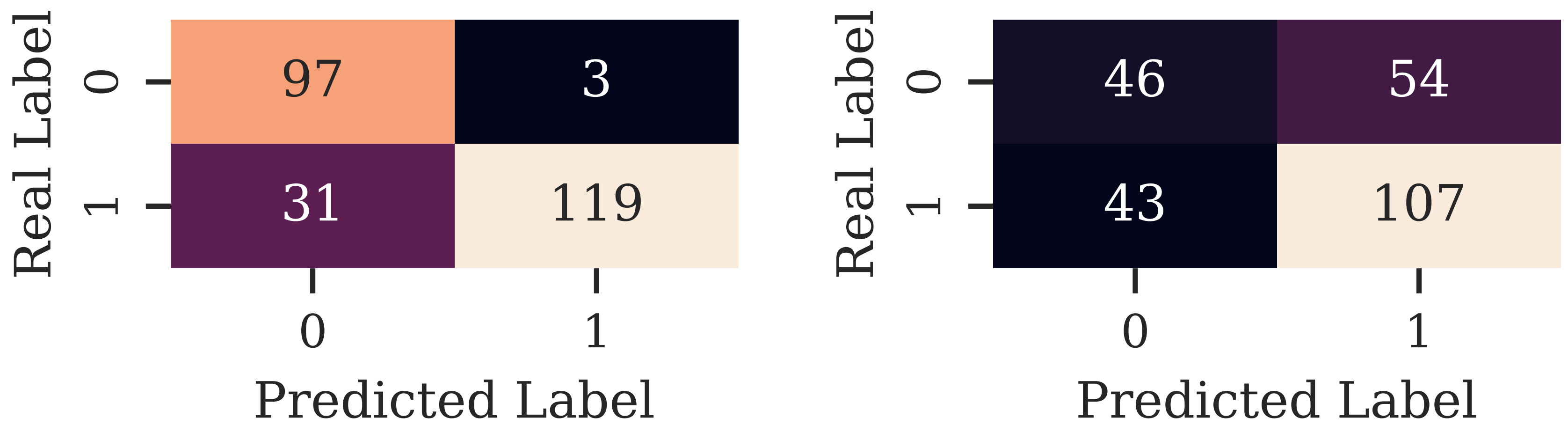


Embedding vector dimension is reduced from 768 to 64.

Results - Comparison with Baseline

Metrics	Standalone Queries			Contextual Queries	
	GPTCache	MeanCache (MPNet)	MeanCache (Albert)	GPTCache	MeanCache
F score	0.56	0.73	0.68	0.67	**0.93**
Precision	0.52	0.72	0.66	0.66	**0.98**
Recall	0.85	0.78	0.77	0.71	0.79
Accuracy	0.72	0.85	0.81	0.61	**0.86**

Table 1. MeanCache outperforms GPTCache on both standalone and contextual queries.



(a) MeanCache [1] (b) GPTCache (baseline) [2]

Figure 1. MeanCache only reports three false hits compared to 54 false hits by GPTCache. False hits are undesirable as they require the user to resend the query to the LLM service to obtain the correct response.

Conclusion

- MeanCache effectively reduces LLM inference costs by caching semantically similar queries locally.
- MeanCache ensures user privacy through federated learning and local caching.
- MeanCache outperforms the baseline (GPTCache), especially in handling contextual queries.

Future Work

- **Multimodal Caching:** Expand MeanCache to support multimodal caching, including vision and audio data.
- **Embedding Quantization:** Investigate quantization of embeddings to significantly speed up semantic caching and reduce memory and disk usage.
- **Optimizing Query Caching**: Assess caching benefits for different queries to optimize cache (e.g., excluding time-sensitive, grammatical, and long queries.)
- **Eviction Policy:** Investigate optimal eviction policies for semantic caching.
- **Benchmarking:** Create diverse benchmark dataset to evaluate semantic caching across domains (e.g., medical, software engineering).

MeanCache Key Features and Evaluation Highlights

- **Privacy-Preserving:** No user data stored on central servers.
- **Scalable:** Efficient local caching and federated model training.
- **Contextual Queries:** Only 3 false hits vs. 54 by GPTCache.
- **High Accuracy:** MeanCache surpasses baseline for contextual queries with 25% higher F-score and 32% better precision.
- **Storage & Speed:** Embedding compression reduces storage by 83% and speeds up matching by 11%.

References

[1] Gill, Waris, et al. "MeanCache: User-Centric Semantic Cache for Large Language Model Based Web Services." arXiv preprint arXiv:2403.02694 (2024).

[2] Bang, Fu. "GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings." Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023). 2023.