Summaries as Centroids for Interpretable and Scalable Text Clustering

Jairo Diaz-Rodriguez

ICLR 2026

Department of Mathematics and Statistics, York University

Motivation

Two practical constraints in text clustering

- **Interpretability gap:** k-means prototypes are numeric averages in embedding space, hard to inspect and audit.
- **Scalability gap:** many LLM-based clustering methods rely on LLM calls that grow with dataset size (pairwise comparisons, iterative labeling, extensive refinement).

Goals

- Preserve k-means simplicity and objective-based updates.
- Produce semantically coherent, human-readable centroids.
- Keep LLM cost bounded, independent of dataset size.

Core idea

Summary-as-centroid

keep assignments in embedding space, but anchor each cluster prototype to a short summary that humans can read and audit.

Every few iterations:

- Assign documents by nearest centroid in embedding space
- Summarize each cluster into a short text
- Embed the summary and use it as the updated centroid

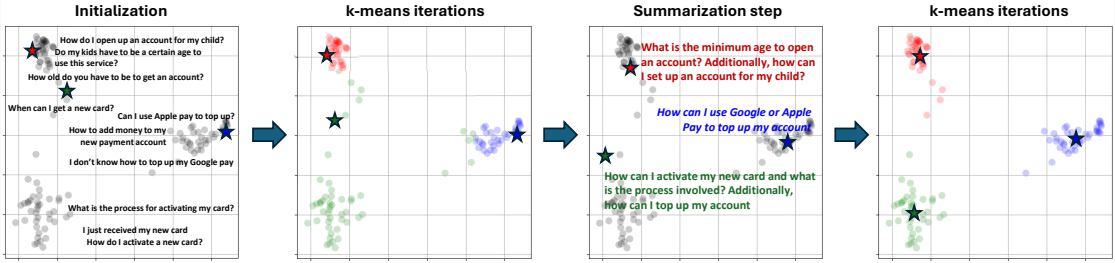

Figure 1: Summarization can redirect centroids toward more semantically coherent partitions.

Two variants

k-NLPmeans (zero-LLM)
lightweight, deterministic summarizers; offline, low-cost, stable.

k-LLMmeans (bounded LLM usage)
LLM-generated summaries as centroids; predictable cost.

Advantages

What you get over vanilla k-means

- **Interpretable centroids:** Better prototypes, same backbone. Each cluster becomes a concise description.
- **Easier debugging:** summaries reveal mixed-topic clusters and unstable merges quickly.
- **Still objective-driven:** assignments remain nearest-centroid in embedding space.

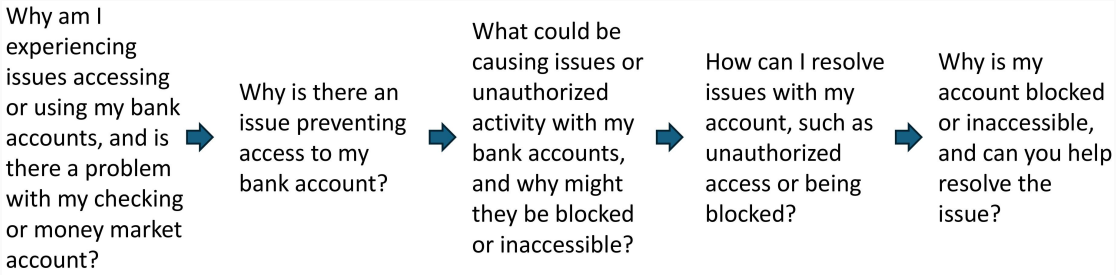

Figure 2: Summary centroids provide readable cluster prototypes for inspection and auditing.

Advantages

What you get over many LLM clustering pipelines

- Predictable cost (bounded LLM usage)
- No dataset-size scaling in LLM calls
- Works in both zero-LLM and LLM-assisted settings

Results: quality vs number of LLM calls

ACC/NMI and prompt counts on three benchmarks

Method	CLINC			GoEmo			MASSIVE (D)		
	prompts	ACC	NMI	prompts	ACC	NMI	prompts	ACC	NMI
k-means	0	73.8	90.8	0	22.0	22.0	0	58.4	63.7
k-NLPmeans	**0**	81.5	93.0	**0**	23.2	22.6	**0**	60.4	65.7
k-LLMmeans	**750**	81.4	93.0	**135**	24.2	24.3	**90**	62.3	65.9
LLMEdgeRefine (SOTA)	1350	86.8	94.9	895	34.8	29.7	892	63.1	68.7

Robustness

Stable across:

- **Embedding models:** DistilBERT, e5-large, S-BERT, and text-embedding-3-small(OpenAI).
- **Summarization strategies (for k-NLPmeans)** : TextRank, centroid-based, LSA-style.
- **Summarization prompts and LLM models (for k-LLMmeans):** GPT-4o, Llama-3.3, Claude-3.7 and DeepSeek-V3.

Streaming scalability: mini-batch variant

Mini-batch summary centroids

- Real-time clustering with bounded memory
- Evolving, readable summaries over time
- Bounded LLM usage by design

Resuts: quality and scalability

StackExchange stream (2020–2023)

205,943 posts clustered with no more than **3,850** LLM calls for mini-batch k-LLMmeans, improving over mini-batch baselines.

Method (ACC/NMI)	2020	2021	2022	2023
k-means (full data)	73.4/80.6	67.7/79.0	68.6/79.0	72.0/79.6
mini-batch k-means	67.0/78.2	67.7/77.4	67.5/77.6	67.2/77.0
mini-batch k-NLPmeans	68.0/79.5	67.9/78.5	69.0/78.9	71.6/78.8
mini-batch k-LLMmeans	75.4/81.6	73.5/80.2	72.8/80.1	72.7/80.1

Takeaways

Main takeaways

- **Interpretable prototypes:** summary centroids make clusters easy to inspect and audit.
- **Scalable cost:** k-LLMmeans uses a small, fixed summarization budget independent of dataset size.
- **Robust in practice:** stable across embeddings, summarizers, LLM generators, and mild hyperparameter changes.
- **Flexible deployment:** k-NLPmeans for zero-LLM, k-LLMmeans for higher accuracy with controlled cost.

Reproducible Research

https://github.com/jairoadiazr/summaryCentroids