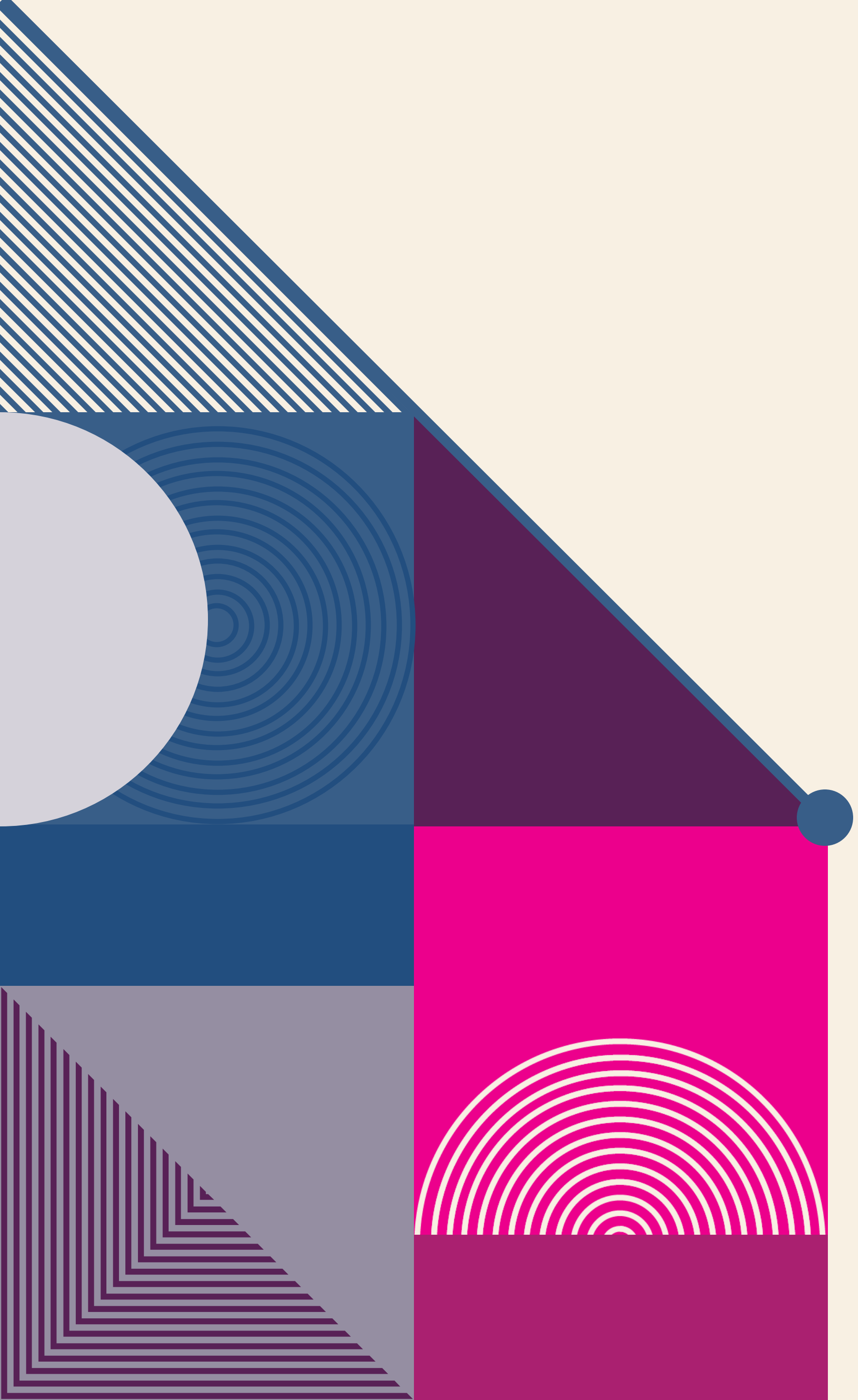
CLUSTERING TEXTUAL DATA

DUTT SALVEEN, PRUSAK PATRYK, TIURINA KARINA



AGENDA

Project Recap

Methodology

Results

Takeaways

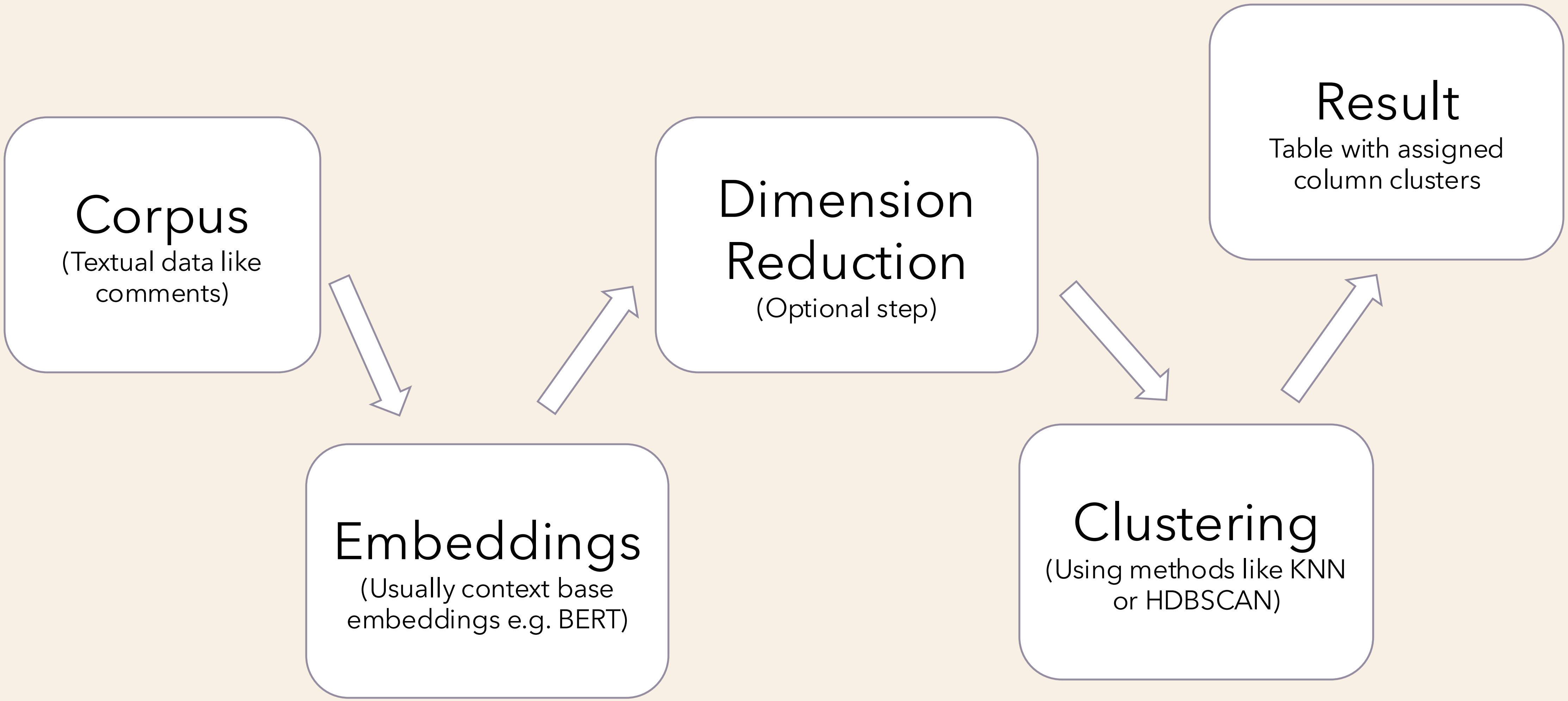
REMINDER ON POC

- Datasets: 20 newsgroups, Amazon Reviews;
- Tested pipelines:
  - Word2Vec embeddings clustering;
  - SBERT embeddings clustering;
  - SBERT embeddings with SVM;
- Best F1 score for SVM with non-reduced embeddings.

REMINDER ON POC

- Planned work from POC:
  - Test different embeddings, e.g. BERT and LLM embeddings;
  - Substitute clustering with classification approach;
  - Introduce LLMs for dataset labeling to aid in the supervised learning process.

CLUSTERING PIPELINE POC



Corpus
(Textual data like comments)

Embeddings
(Usually context base embeddings e.g. BERT)

Dimension Reduction
(Optional step)

Clustering
(Using methods like KNN or HDBSCAN)

Result
Table with assigned column clusters

NEW DATASET

- News BBC News Dataset
- https://www.kaggle.com/c/learn-ai-bbc
- 2225 articles labeled under 5 categories
- Dataset is designed to predict a label for previously unseen articles

WHAT DID WE DO?

- Pivoted to classification instead of clustering
- Implemented the paper "Text Clustering as Classification with LLMs"
- SVM vs LLMs in text classification analysis

CLASSIFICATION VS CLUSTERING

Pros:

- Proper classes make more sense in business scenarios.
- Easily understood accuracy when training the model

Cons:

- Classification needs labeled data

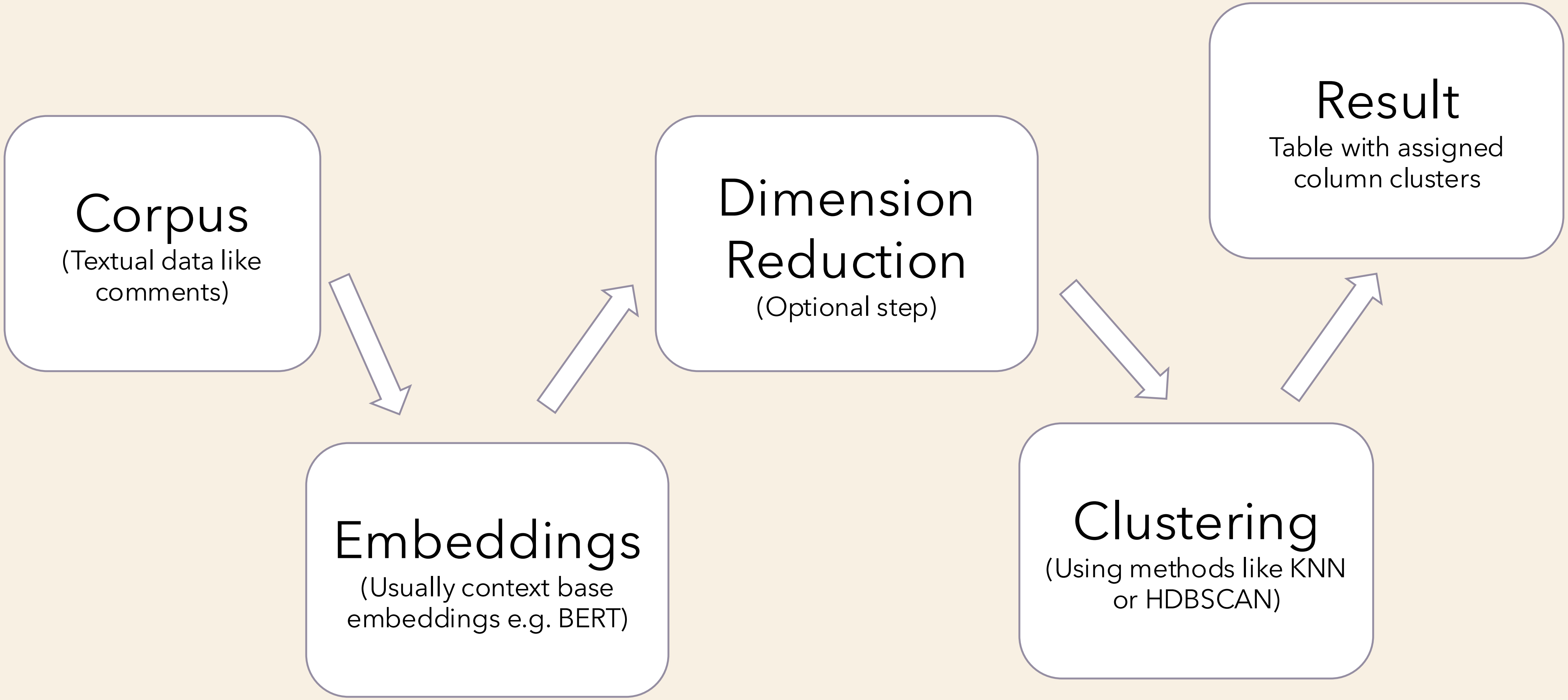
Pros:

- No data labeling is needed hence any textual data could be used.

Cons

- No way of defining clusters.
- This reduces the value for businesses

TRADITIONAL ML APPROACH



Corpus
(Textual data like comments)

Embeddings
(Usually context base embeddings e.g. BERT)

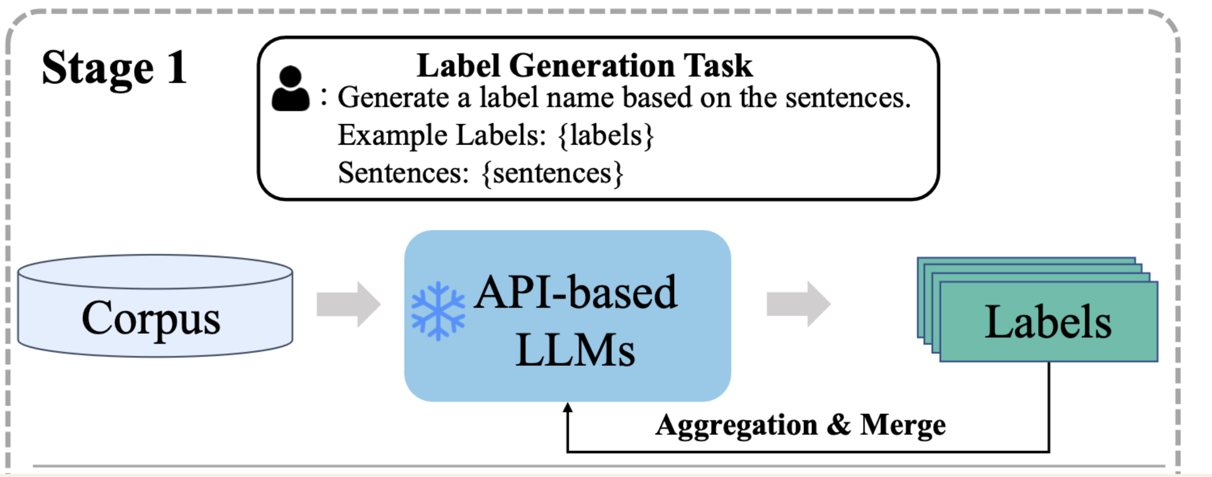
Dimension Reduction
(Optional step)

Clustering
(Using methods like KNN or HDBSCAN)

Result
Table with assigned column clusters

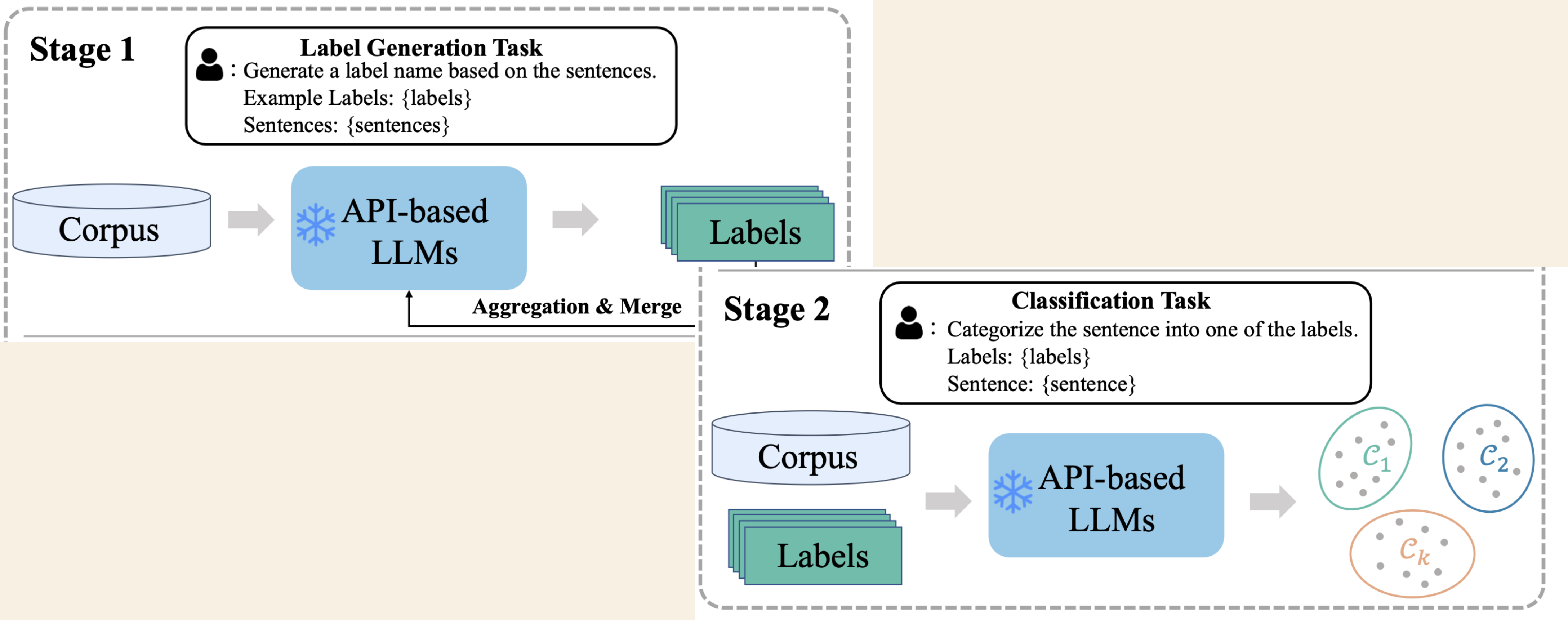
CLUSTERING AS CLASSIFICATION

USING LLMS - MAIN IDEA



CLUSTERING AS CLASSIFICATION

USING LLMS - MAIN IDEA



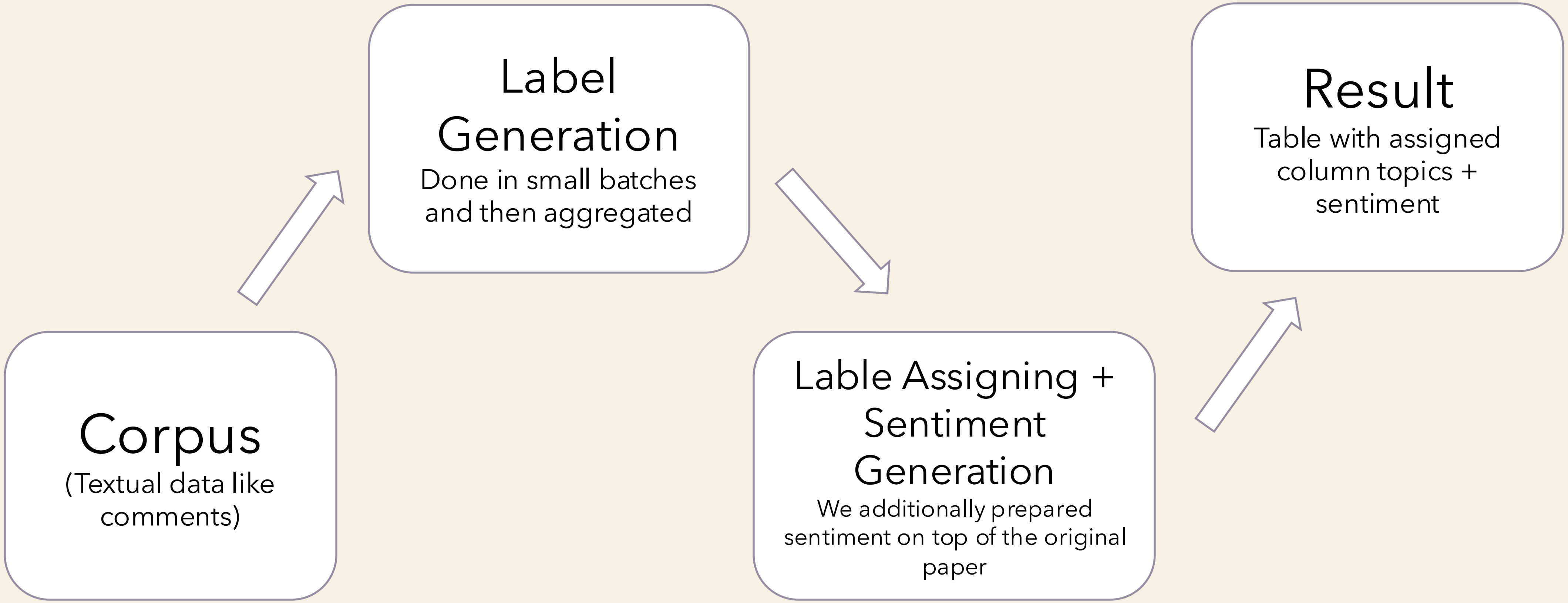
Label Generation Task

: Generate a label name based on the sentences.
Example Labels: {labels}
Sentences: {sentences}

Classification Task

: Categorize the sentence into one of the labels.
Labels: {labels}
Sentence: {sentence}

IMPLEMENTATION



Corpus
(Textual data like comments)

Label Generation
Done in small batches and then aggregated

Lable Assigning + Sentiment Generation
We additionally prepared sentiment on top of the original paper

Result
Table with assigned column topics + sentiment

IMPLEMENTATION FEATURES

- Our pipeline is **Open Source.** The paper used ChatGPT API calls while we used Gemini 9B which runs locally model.

- Sentiment Analysis – In addition to the topic, sentiment is being predicted by Gemini at almost **no cost.**

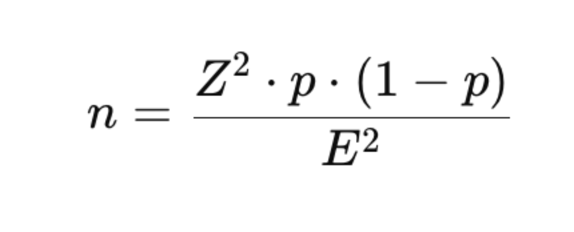
RESULTS

LLM FOR AMAZON REVIEWS

- We took sample of 10000 reviews from 1.5M from amazon
- ~10 minutes label generation + ~4 hour assignment runtime on M3 Pro chip

RESULTS

HOW TO CALCULATE ACCURACY?



$$n = \frac{Z^2 \cdot p \cdot (1 - p)}{E^2}$$

n = required sample size
Z = Z-score corresponding to the desired confidence level (e.g., 1.96 for 95% confidence)
p = estimated proportion of correct data (assumed to be 0.5 if unknown, as it maximizes the sample size)
E = margin of error (e.g., 0.05 for ±5%)

RESULTS

HOW TO CALCULATE ACCURACY?

$$n = \frac{Z^2 \cdot p \cdot (1 - p)}{E^2}$$

n = 369.98

n = required sample size
Z = Z-score corresponding to the desired confidence level (e.g., 1.96 for 95% confidence)
p = estimated proportion of correct data (assumed to be 0.5 if unknown, as it maximizes the sample size)
E = margin of error (e.g., 0.05 for ±5%)

RESULTS

LLM FOR AMAZON REVIEWS

- We took sample of 10000 reviews from 1.5M from amazon
- ~10 minutes label generation + ~4 hour assignment runtime on M3 Pro chip
- Accuracy: 97%

RESULTS

LLM FOR AMAZON REVIEWS

- We took sample of 10000 reviews from 1.5M from amazon

- ~4 hour runtime on M3 Pro chip

- Accuracy: 97%

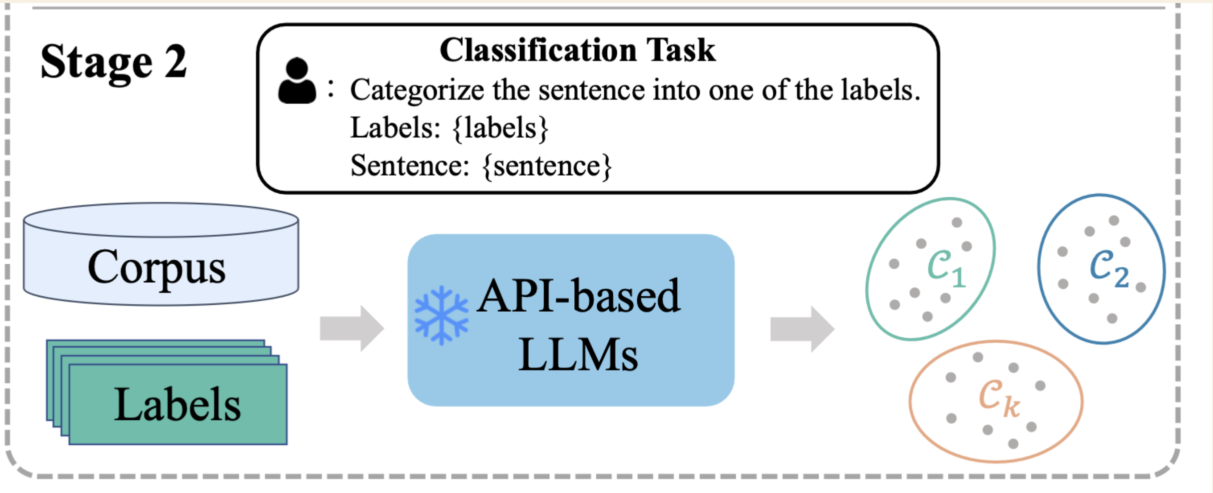
- Sentiment Accuracy:

Weighted Precision: 0.858
Weighted Recall: 0.875
Weighted F1 Score: 0.854
Weighted Accuracy: 0.903

CLASSIFICATION

LLMS VS SVM

- From PoC, SVM paired with RoBERTa embeddings yielded best results.
- To classify using LLMs, having already the labels we use stage 2 of pipeline given in paper “clustering as classification”



Classification Task

: Categorize the sentence into one of the labels.
Labels: {labels}
Sentence: {sentence}

CLASSIFICATION

USING BBC NEWS DATASET

- LLM Classification (Gemini 9B)
- Runtime ~45minutes

- Accuracy: 94.97%
- F1 Score: 94.93%
- Precision: 95.16%
- Recall: 94.97%

CLASSIFICATION

USING BBC NEWS DATASET

- LLM Classification (Gemini 9B)
- Runtime ~45minutes

- Accuracy: 94.97%
- F1 Score: 94.93%
- Precision: 95.16%
- Recall: 94.97%

- SVM with RoBERTa Embeddings
- Runtime ~3minutes

- Accuracy: 98.64%
- F1 Score: 98.64%
- Precision: 98.65%
- Recall: 98.64%

CLASSIFICATION

USING 20NEWSGROUP DATASET

- LLM Classification (Gemini 9B)
- Runtime (on 80% dataset) ~6hours

- Accuracy: 60.25%
- F1 Score: 64.69%
- Precision: 77.62%
- Recall: 60.25%

- SVM with RoBERTa Embeddings
- Runtime ~12minutes

- Accuracy: 64.17%
- F1 Score: 64.06%
- Precision: 64.43%
- Recall: 64.17%

TAKEAWAYS

- SVM with RoBERTa is surprisingly good compared to LLM
- Topics can be generated by LLMs, but it should be supervised.

FUTURE WORKS

- SVM classification of dataset labelled by LLM.

REFERENCES

- Chen Huang and Guoxiu He, "Text Clustering as Classification with LLMs," StatNLP Research Group, Singapore University of Technology and Design; School of Economics and Management, East China Normal University.
- Kaggle. "Learn AI with BBC." Available at: https://www.kaggle.com/c/learn-ai-bbc. Accessed: January 22, 2025.

THANK YOU