Semantic Space Clustering and Controlled Generation of Synthetic Reviews for Mexican Magical Towns

Dulce Ramírez[1], Jose Ortiz-Bejar[2,*], Jesus Ortiz-Bejar[3] and Mario Graff[1]

[1]Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación (INFOTEC), Aguascalientes, Aguascalientes, México

[2]Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo (UMSNH), Morelia, Michoacán, México

[3]Facultad de Ciencias Físico Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo (UMSNH), Morelia, Michoacán, México

Abstract

This work presents a methodology for generating balanced synthetic data by applying K-Means clustering in semantic embedding spaces and structured prompting with large language models. Our proposal consists of selecting representative prototypes through K-Means (k = 200) over Sentence-BERT embeddings. The prototypes are used as a guide for controlled generation via 2-shot prompting. The system generates a balanced set of 3,000 samples (600 per polarity) using Gemini 3 Flash with a fixed temperature of $\tau = 0.7$, complemented by automated post-processing with Claude to ensure completeness and consistency across the generated samples. Evaluation in REST-MEX 2026 demonstrates competitive performance (5th place out of 18 teams, Track Score = 0.4933, Macro $f_1 = 0.5190$), outperforming the official baseline and standing out particularly in the classification of moderately negative sentiment (f_1 = 0.402), surpassing even ensemble-based approaches for this class. These results highlight the effectiveness of combining semantic clustering and controlled text generation for sentiment-oriented data augmentation tasks.

Keywords

Few-Shot Learning, Sentiment Analysis, Data Augmentation, Synthetic data generation, K-Means Clustering, Prompt Engineering.

1. Introduction

iberlef2026overview The REST-MEX 2026 challenge [1, 2], in contrast to previous editions [3, 4, 5, 6], proposes an innovative paradigm of indirect evaluation for synthetic data generation, in which each participant generates 3,000 balanced synthetic reviews, a text classifier is trained using the synthetic data, and the resulting model is evaluated on a hidden test set of real reviews. This framework assesses the ability of synthetic data to capture both the distributional properties and the semantic and linguistic characteristics of the target domain.

The dataset provided by the task organizers contains 208,051 reviews of tourist establishments in Mexico's Magic Towns, with the class imbalance characteristic of real-world data; approximately 60% have polarity 5 (very positive), while polarities 1–2 account for less than 10%. This imbalance introduces several technical challenges. First, training a robust classifier capable of generalizing to real data requires balancing the five polarity levels across the three establishment types (Hotel, Restaurant, and Attractive). Second, the 3,000 synthetic reviews must adequately reflect the diversity of aspects, styles, and vocabulary present in the 208,051 real reviews to avoid leaving large regions of the semantic embedding space underrepresented. Third, effective polarity control is needed to generate reviews that faithfully capture the intended sentiment without relying on explicit labels, since direct instructions such as "write a negative review" tend to produce artificial or stereotyped patterns. Finally, maintaining naturalness and coherence is crucial to avoid generic or repetitive outputs that a classifier could exploit without learning the true linguistic characteristics of the domain.

IberLEF 2026, September 2026, León, Spain

*Corresponding author.

dramirez@infotec.mx (D. Ramírez); jose.ortiz@umich.mx (. J. Ortiz-Bejar); jesus.ortiz@umich.mx (J. Ortiz-Bejar); mario.graff@infotec.mx (M. Graff)

This work addresses these challenges with a two-stage strategy: first, K-Means clustering is used to select a set of representative examples (prototypes), and the second step consists of performing few-shot prompting for controlled generation while maintaining a perfect balance by construction.

2. Methodology

2.1. Problem Formalization

Let the original dataset be defined as:

$$D = \{(r_i, p_i, t_i) : i = 1, \ldots, n\} \tag{1}$$

where r_i represents the i-th review text, $p_i \in \{1, 2, 3, 4, 5\}$ denotes its polarity score, and t_i is the review type.

The objective is to generate a synthetic dataset S of size $|S| = 3000$, consisting of synthetic reviews s, such that the synthetic data is perfectly balanced across all polarity levels. Mathematically, this requires:

$$|\{s \in S : \text{polarity}(s) = k\}| = 600, \quad \forall k \in \{1, 2, 3, 4, 5\} \tag{2}$$

$$|\{s \in S : \text{polarity}(s) = p \wedge \text{type}(s) = t\}| = 200, \qquad \forall (p, t) \tag{3}$$

This perfectly uniform distribution mitigates the class imbalance present in the original dataset, ensuring that the trained classifier has equal representation for all classes.

2.2. Vector Semantic Representation

Each review text is transformed into a dense vector through a pre-trained neural embedding function:

$$f_{\text{embed}} : \text{Text} \to \mathbb{R}^d \tag{4}$$

where $d = 384$ represents the dimensionality of the embedding space. Formally, for each review r_i within the subset $D_{(p,t)}$ (representing reviews of polarity p and type t), the embedding e_i is computed as:

$$e_i = f_{\text{embed}}(r_i) \tag{5}$$

To perform this mapping, we employ the *paraphrase-multilingual-MiniLM-L12-v2* model [7], a Sentence-BERT (SBERT) architecture pre-trained on multilingual semantic similarity tasks. This choice is justified by SBERT's ability to capture dense semantics at the sentence and paragraph level rather than treating isolated words. Consequently, the Euclidean distance $\|e_i - e_j\|_2$ in $\mathbb{R}^{384}$ reliably reflects the semantic similarity between reviews. This property is fundamental for the subsequent clustering stage, as it enables the grouping of semantically coherent texts.

In terms of computational efficiency, the $208{,}051$ embeddings were precomputed and stored on disk as a $208{,}051 \times 384$ matrix (occupying approximately 62 MB). This avoids redundant recomputation in downstream tasks. The initial embedding extraction required approximately 45 minutes of computation on an Apple M2 Ultra CPU.

2.3. K-Means Clustering and Representative Prototype Selection

For each (p, t) combination, the subset $D_{(p,t)}$ is partitioned using K-Means into $k = 200$ clusters. The objective of K-Means is to minimize the intra-cluster variance:

$$\min_{C,A} J(C, A) = \sum_{j=1}^{k} \sum_{i:A(i)=j} \|e_i - c_j\|_2^2 \tag{6}$$

where $C = \{c_1, \ldots, c_k\}$ denotes the set of centroids in $\mathbb{R}^d$, $A : \{1, \ldots, n_{(p,t)}\} \to \{1, \ldots, k\}$ is the cluster assignment function, e_i represents the review embeddings within $D_{(p,t)}$, and $\|\cdot\|_2$ denotes the Euclidean L_2 norm.

Iterative Optimization (Lloyd's Algorithm) The objective function is optimized iteratively following Lloyd's algorithm:

- **Initialization:** The initial centroid set
$$C^{(0)} = \left\{c_1^{(0)}, \ldots, c_k^{(0)}\right\} \tag{7}$$
is constructed by randomly selecting k distinct points from the embedding subset $\{e_1, \ldots, e_{n_{(p,t)}}\}$.
- **E-step (Assignment):** Each embedding is assigned to its nearest centroid:
$$A^{(t+1)}(i) = \underset{j\in\{1,\ldots,k\}}{\arg\min} \left\|e_i - c_j^{(t)}\right\|_2^2 \tag{8}$$
- **M-step (Update):** The centroids are updated by calculating the mean of all embeddings assigned to that cluster:
$$c_j^{(t+1)} = \frac{1}{|I_j|} \sum_{i\in I_j} e_i \tag{9}$$
where $I_j = \{\, i : A^{(t+1)}(i) = j \,\}$ represents the set of indices assigned to cluster j.
- **Stopping Criterion:** The iteration loops until the total shift of the centroids falls below a predefined threshold ε:
$$\sum_{j=1}^{k} \left\|c_j^{(t+1)} - c_j^{(t)}\right\|_2^2 < \varepsilon \tag{10}$$

The hyperparameter $k = 200$ is selected based on two primary considerations. First, this granularity is sufficient to capture the diverse array of aspects (e.g., location, cleanliness, service, and price), narrative styles (ranging from formal to colloquial), and domain-specific vocabulary characteristic of Mexican tourism reviews. Second, this choice ensures that each centroid represents a populated region of the embedding space; even within the smallest class subset (approximately 800 reviews for polarity 1), the partitioning yields an average of four reviews per cluster, thereby preventing over-segmentation and minimizing the risk of empty or statistically isolated clusters.

2.3.1. Prototype selection.

One prototype m_j is selected for each cluster j. Every prototype is defined as the review embedding within that cluster that lies closest to its respective cluster centroid c_j:

$$m_j = \underset{i\in I_j}{\arg\min} \|e_i - c_j\|_2^2 \tag{11}$$

Consequently, the complete set of prototypes for a given (p, t) combination is defined as:

$$M_{(p,t)} = \{m_1, m_2, \ldots, m_k\} \tag{12}$$

where $k = 200$.

The selection of an actual review from within each cluster is necessary because the centroids c_j are points in $\mathbb{R}^{384}$ that do not correspond to any real text. By selecting the nearest physical review, we ensure that the resulting exemplars are authentic instances from the domain. This preserves the natural syntax, vocabulary, and narrative structure of the original data, rather than introducing an artificial *a*verage text.

2.4. Few-Shot Prompting for Controlled Generation

For each representative prototype $m \in M_{(p,t)}$, a 2-shot prompt containing two reference exemplars is constructed to guide the language model. This framework forces the model to preserve target stylistic traits and domain characteristics without explicitly exposing polarity labels. The prompt structure is defined as:

$$\text{Prompt}(m, p, t, \text{town}) = [\text{Examples}] + [\text{Context}] + [\text{Instructions}] \tag{13}$$

where the component $[\text{Examples}] = \{s_1, s_2\}$ consists of two reference reviews:

$$s_1 = m \tag{14}$$

which acts as the structural prototype from the cluster, and

$$s_2 \sim \text{Uniform}\left(D_{(p,t)} \setminus M_{(p,t)}\right) \tag{15}$$

which is a review sampled uniformly at random from the remaining non-prototype elements of the same subclass to provide stylistic variance.

The conditioning variables within the [Context] block are formalized as follows:

$$\text{type} = \text{natural\_name}(t) \tag{16}$$

where $\text{natural\_name}(\cdot)$ maps the internal type identifier t to its corresponding natural language category (e.g., hotel, restaurant, or attraction). The destination variable,

$$\text{town} = \text{balanced\_round\_robin\_assignment}() \tag{17}$$

allocates a Mexican municipality to ensure spatial equilibrium across the generated corpus. Finally,

$$L_{\text{target}} = |s_1|_{\text{words}} \tag{18}$$

defines the target length constraint, matching the word count of the cluster prototype.

The [Instructions] block enforces the following operational constraints during generation:

- Maintain the semantic style, linguistic register, and tone exhibited in $\{s_1, s_2\}$.
- Generate completely original text without paraphrasing or duplicating the reference exemplars.
- Restrict the output length to approximately L_{target} words.
- Terminate the text cleanly with a period to guarantee structural completeness.
- Return raw plain text exclusively, omitting Markdown syntax or special formatting wrappers.

2.4.1. Design Rationales and Control Mechanisms.

The configuration of the prompt parameters and generation constraints is grounded in the following methodological justifications:

Justification for 2-Shot Learning. The inclusion of exactly two reference exemplars balances prototype fidelity with systemic variation. The first exemplar, s_1, acts as the semantic prototype representing the cluster center c_j; it captures the central thematic aspects, sentiment intensity, and domain-specific vocabulary of that dense region. Conversely, the second exemplar, s_2, is sampled dynamically to introduce lexical and structural diversity within the same (p, t) class. This multi-exemplar conditioning prevents the language model from becoming excessively anchored to a single syntactic pattern. Limiting the prompt to 2-shot learning, rather than an N-shot configuration where $N > 3$, follows established text generation benchmarks [8]. Higher values of N risk diluting the target stylistic pattern or introducing structural confusion due to the heterogeneity of the text.

Length Control Mechanism. To ensure that the synthetic reviews maintain a realistic length distribution relative to the original corpus, we enforce an explicit constraint of L_{target} words. This target is estimated by mapping the character length of the cluster prototype to an approximate word count using a standard heuristic ratio of 5 characters per word:

$$L_{\text{target}} = \left\lfloor \frac{|s_1|_{\text{chars}}}{5} \right\rfloor \tag{19}$$

Bounding the generation length prevents the model from drifting off-topic in excessively long outputs, while simultaneously avoiding overly brief generations that lack sufficient semantic density.

Implicit Polarity Control. The prompt strictly omits explicit structural metadata or hard labels such as "polarity 1" or "negative sentiment." Instead, the language model is forced to inductively infer the target tone, sentiment nuances, and register directly from the latent features embedded within $\{s_1, s_2\}$. This implicit conditioning strategy generates significantly more natural and diverse reviews, mitigating the risk of artificial, repetitive patterns or stylistic collapses often triggered by direct behavioral instructions (e.g., the over-generation of simplistic modifiers like "terrible," "awful," or "very bad"). An example of the prompt used can be seen at Appendix B.

2.5. Generative Model and Parameters

Generation is performed using Google Gemini 3 Flash Preview [9], configured to balance creativity with structural adherence to the reference examples. Formally, the language model operates as an autoregressive decoder. The probability of generating the next token w_t given the preceding context and the prompt is defined via a softmax activation over the vocabulary logits:

$$P(w_t = v \mid w_1, \ldots, w_{t-1}, \text{Prompt}) = \frac{\exp(z_{t,v}/\tau)}{\sum_{v' \in V} \exp(z_{t,v'}/\tau)} \tag{20}$$

where V is the model's vocabulary, $\tau = 0.7$ denotes the fixed sampling temperature, and the logit vector z_t is produced by the mapping function f_θ parameterized by the frozen weights θ of the model:

$$z_t = f_\theta\big(w_1, \ldots, w_{t-1}, \text{Prompt}\big) \tag{21}$$

Tokens are then iteratively sampled according to this calibrated distribution:

$$w_t \sim P(\cdot \mid w_1, \ldots, w_{t-1}, \text{Prompt}) \tag{22}$$

Hyperparameter Tuning and Temperature Dynamics. The sampling temperature τ scales the logit space prior to activation, fundamentally controlling the entropy of the output distribution:

- $\tau \to 0$ (Deterministic Decoding): The distribution collapses toward a greedy *argmax* selection. The model strictly emits the single most probable token sequence, eliminating variability and leading to highly repetitive, mechanical reviews within the same class.
- $\tau = 0.7$ (Empirically Validated Balance): This setting introduces sufficient stochasticity to promote lexical and structural diversity across generations while firmly preserving semantic and grammatical coherence.
- $\tau \to \infty$ (Uniform Sampling): The distribution approaches maximum entropy. Randomness is maximized, which drastically escalates the probability of syntactic degradation, ungrammatical formatting, and thematic drift.

The choice of a fixed $\tau = 0.7$ was justified through empirical validation across 15 pilot review generations (representing each (p, t) combination). In contrast, alternative experiments employing a dynamic temperature sampled from a uniform distribution, $\tau \sim \text{Uniform}(0.5, 0.8)$, introduced notable structural inconsistencies. These included fragmented sentences and abrupt syntactic errors, which were triggered when the temperature drifted toward either extreme of the interval.

Model Configuration Details. To complete the model environment setup, two final adjustments were made:

- The token length horizon parameter `max_output_tokens` was set to 4096. This boundary proved more than sufficient to guarantee text completion and was never saturated during execution.
- API responses containing internal chain-of-thought metadata, flagged via the `thought_signature` field, were programmatically intercepted and filtered out to retain only raw, user-facing plain text.

For details about implementation of the full pipeline see Appendix C.

3. Experimental Results

3.1. Dataset Distribution and Completeness

The data generation pipeline successfully constructed a balanced synthetic corpus, adhering strictly to the predefined stratification criteria. As detailed in Table 1, the final synthetic dataset comprises exactly 3,000 reviews, distributed uniformly with 600 instances per polarity level. By extension, this ensures an explicit density of 200 reviews for each unique (p, t) polarity and establishment type combination, satisfying the class-balancing objective by construction.

Regarding structural completeness, the automated quality-assurance filters verified that all 3,000 instances met the grammar and termination constraints after the post-processing phase. In terms of architectural performance, 77% of the final corpus was synthesized directly by the baseline model without requiring adjustments. The remaining 23% (approximately 690 reviews) failed the structural validation due to early truncation and were programmatically completed during the final phase using Claude.

Table 1
Targeted Class Distribution of the Synthesized Corpus

Polarity	Review Count	Percentage	Target Distribution
1	600	20.0%	20.0%
2	600	20.0%	20.0%
3	600	20.0%	20.0%
4	600	20.0%	20.0%
5	600	20.0%	20.0%

3.2. Statistical Length Analysis

To validate the behavioral integrity of the generation process, a comparative analysis of review lengths (measured in characters) was conducted between the original and synthetic datasets. The descriptive statistics for both corpora are summarized in Table 2.

Table 2
Comparative Length Statistics (Characters)

Dataset	Mean Length	Standard Deviation	Median Length
Original ($n = 208{,}051$)	356.9	288.9	259.0
Synthetic ($n = 3{,}000$)	458.4	301.7	391.0

Enforcing the explicit length heuristic constraint (L_{target}) aims to mitigate the systemic bias toward over-elaboration and excessively long outputs typically observed in unconstrained large language model generation.

3.3. Evaluation on REST-MEX 2026

To validate the downstream utility of the generated synthetic dataset, the corpus was evaluated within the framework of the official REST-MEX 2026 shared task. The evaluation protocol dictated training a standardized classifier exclusively on the $3{,}000$ synthetic reviews, which was subsequently evaluated on a hidden, real-world test set. The sentiment classification task spanned five ordinal classes ranging from 1 (very negative) to 5 (very positive). The competitive track drew 18 participating teams submitting a total of 63 system variants, with performance primarily benchmarked using a comprehensive Track Score alongside standard classification metrics.

The official performance metrics obtained by our submitted system (team identifier: *Tata-VQ_UMSNH*) are summarized in Table 3, alongside its final competitive standing.

Table 3
Official Global Performance and Competition Ranking

Evaluation Metric	Value	Competition Ranking Parameter	Value
Track Score	0.4933	Global Team Position	5 / 18
Macro F_1 (Polarity)	0.5190	Individual Submission Rank	23 / 63
Accuracy (Polarity)	58.86%	Team Percentile	72.2nd
Average Precision	0.5746	Official Baseline Track Score	0.4721
Average Recall	0.5155	Top-Ranked Score Relative Gap	-5.6%
Mean Absolute Error (MAE)	0.4620	Baseline Relative Improvement	$+4.5\%$

The proposed system demonstrated strong competitive performance, anchoring itself within the top quartile of the international challenge. The synthetic-trained classifier outperformed the official track baseline by 4.5% in Track Score and placed within a narrow 5.6% margin of the top-ranked individual approach. To further analyze the behavioral nuances of the model trained on synthetic data, Table 4 breaks down the Macro F_1 scores across each sentiment polarity level.

Table 4
Macro F_1 Performance Score Breakdown by Polarity Level

Polarity Level	Semantic Description	Macro F_1 Score
1	Very Negative	0.5898
2	Moderately Negative	0.4020
3	Neutral	0.4557
4	Moderately Positive	0.4343
5	Very Positive	0.7134

An analysis by polarity reveals that the classifier achieves its highest performance on the extreme classes, yielding F_1 scores of 0.590 for very negative and 0.713 for very positive sentiments. This behavior aligns with linguistic theory, as extreme polarity levels typically leverage highly distinctive vocabulary and clear lexical markers. Conversely, performance drops moderately for intermediate classes 3 and 4 (F_1 scores of 0.456 and 0.434, respectively), which are inherently characterized by semantic ambiguity and mixed-sentiment expressions.

Crucially, the model achieves a noteworthy F_1 score of 0.402 on the moderately negative class (polarity 2). In the context of the competition, this specific subclass represents the only instance where our individual model architecture outperformed more complex ensemble approaches. This suggests that the proposed combination of K-Means clustering and 2-shot prompting is uniquely robust at capturing and formalizing the subtle, non-trivial linguistic patterns underlying moderately negative domain data.

4. Conclusions and Future Work

This work introduces a systematic, mathematically grounded methodology for generating balanced synthetic reviews by integrating K-Means clustering into semantic embedding spaces and employing

structured few-shot prompting.

The primary contribution of this research is the development of a fully automated, reproducible, and highly parallelizable end-to-end pipeline capable of synthesizing $3{,}000$ balanced reviews in approximately three hours. To the best of our knowledge, this framework represents the first application of semantic clustering for the strategic selection of representative exemplars in a few-shot text generation context. By managing this selection programmatically, the methodology guarantees a perfectly uniform distribution, specifically 600 reviews per polarity level and 200 reviews per (p, t) combination, thereby mitigating the class imbalance inherent in the original source corpus by construction. Furthermore, we address a common operational limitation of large language models by introducing a robust post-processing stage, powered by a secondary model, that automatically enforces review completeness. The downstream utility of this pipeline was validated in the REST-MEX 2026 shared task, where a classifier trained on the synthetic corpus ranked 5th out of 18 teams. This performance placed our system in the top quartile, outperforming the official competition baseline and yielding a particularly competitive Macro F_1 score of 0.402 for the moderately negative class, which surpassed even complex ensemble approaches.

Several promising research directions emerge from these experiments. First, future work will explore ensemble data augmentation by pooling datasets from complementary techniques, such as paraphrasing, back-translation, and varied temperature configurations, across diverse language models. Second, to improve classifier performance on intermediate polarities, we will examine N-shot setups ($N > 2$) or use contrastive prompting to clarify semantic boundaries between neutral and mildly positive classes. Third, grid search or Bayesian optimization will be used to systematically map the isolated contributions of hyperparameters for cluster size k, temperature τ, and token limits. Fourth, we may refine semantic clustering by fine-tuning Sentence-SBERT on localized Mexican tourism corpora for domain-specific embeddings. Finally, we propose a classifier-in-the-loop generation framework, where an auxiliary sentiment model filters generated outputs and triggers regeneration when the target and predicted polarities do not match.

A. Appendices

B. Example of the Prompt Used

The following example shows an actual 2-shot prompt generated for polarity 1, establishment type Hotel, and the Magic Town of Taxco.

Read the following two reviews:

> The hotel is disappointing. The rooms are dirty and the service is terrible. I do not recommend it at all. The location is the only redeeming aspect, but it does not compensate for the poor experience.
>
> Very bad experience. The staff is rude and the facilities are in poor condition. I would definitely not return.

Prompt instruction:

> Write a complete review about a hotel in Taxco, Mexico. Use the same style and tone as the reference reviews, with approximately 18 words. Make sure to end the review properly with a period. Generate different content. Return plain text without special formatting.

Example of a 2-shot prompt for review generation (polarity 1, Hotel).

Prompt observations

- Both examples clearly establish a negative tone through natural vocabulary (‘disappointing”, ‘dirty”, ‘terrible”, ‘rude”).

- The target length (18 words) is automatically computed from s_1.
- Explicit instructions are provided regarding originality, completeness, and output format.
- The prompt does not explicitly mention ‘polarity 1” or ‘negative sentiment”; instead, the tone is inferred from the reference examples.

C. Implementation

The complete pipeline consists of four phases executed sequentially.

Phase 1: Embedding Computation

- Input: 208,051 reviews from the cleaned original dataset.
- Processing in batches of 1,000 reviews with a progress bar.
- Output: embedding matrix ($208{,}051 \times 384$) stored in pickle format.
- Execution time: approximately 45 minutes on CPU (Apple M2 Ultra, 24 cores).

Phase 2: Parallel Clustering

- For each (p, t) combination, K-Means clustering is performed with $k = 200$ using k-means++ initialization.
- Parallelization using 24 workers (*ThreadPoolExecutor*) across the 15 (p, t) combinations.
- Selection of the review nearest to each centroid.
- Output: $15 \times 200 = 3{,}000$ representative examples.
- Execution time: approximately 2 minutes.

Phase 3: Parallel Generation with Rate Limiting

- Construction of 3,000 2-shot prompts with balanced assignment of Magic Towns.
- Parallel generation using 34 workers while respecting the API limit of 34 requests per minute (RPM).
- Sliding-window rate limiter for precise request-rate control.
- Incremental thread-safe storage (each generated review is written immediately).
- Execution time: approximately 90 minutes (34 reviews per minute sustained).

Phase 4: Completeness Post-processing

- Automatic detection of truncated reviews (approximately 23
- Criterion: the review does not end with one of ., !, ? or has fewer than 50 characters.
- Correction using Claude Sonnet 4.5, which exhibited a higher completion rate than Gemini.
- Correction prompt: “Complete the review while preserving its tone and natural style.”
- Output: 3,000 complete and grammatically correct reviews.
- Execution time: approximately 15 minutes.

Total execution time

The complete pipeline required approximately 3 hours. The process is highly parallelizable, with overall execution time primarily constrained by API rate limits.

Automatic validation

Each generated review is validated according to the following quality criteria:

- **Completeness:** the review ends with a valid sentence-ending punctuation mark (., !, or ?).
- **Format:** the review contains no Markdown syntax (e.g., **, ##) and no meta-text such as “Here is” or “I present”.
- **Length:** between 50 and 1,500 characters, preventing excessively short or excessively long outputs.

Reviews that fail validation are processed during Phase 4 of the post-processing stage.

Acknowledgments

Authors acknowledge the support provided by SECIHTI through Project CF-2023-I-1174 within the framework of the *Ciencia de Frontera 2023.*

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly, Anthropic Claude, and Google Gemini in order to: Grammar, syntax, style correction, and coherence checking. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] M. Á. Álvarez-Carmona, M. Reyes-Sierra, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, L. Bustio-Martínez, V. Herrera-Semenets, Overview of rest-mex at iberlef 2026: Research on synthetic text modeling for mexican magical towns, Procesamiento del Lenguaje Natural 77 (2026).

[2] A. Bonet-Jover, J. Á. González-Barba, L. Chiruzzo, Overview of IberLEF 2026: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS. org, 2026.

[3] M. Á. Álvarez-Carmona, R. Aranda, S. Arce-Cárdenas, D. Fajardo-Delgado, R. Guerrero-Rodríguez, A. P. López-Monroy, J. Martínez-Miranda, H. Pérez-Espinosa, A. Y. Rodríguez-González, Overview of rest-mex at iberlef 2021: Recommendation system for text mexican tourism (2021).

[4] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, D. Fajardo-Delgado, R. Guerrero-Rodríguez, L. Bustio-Martínez, Overview of rest-mex at iberlef 2022: Recommendation system, sentiment analysis and covid semaphore prediction for mexican tourist texts, Procesamiento del Lenguaje Natural 69 (2022) 289–299.

[5] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, V. Muñiz-Sánchez, A. P. López-Monroy, F. Sánchez-Vega, L. Bustio-Martínez, Overview of rest-mex at iberlef 2023: Research on sentiment analysis task for mexican tourist texts, Procesamiento del Lenguaje Natural 71 (2023) 425–436.

[6] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, L. Bustio-Martínez, V. Herrera-Semenets, Overview of rest-mex at iberlef 2025: Researching sentiment evaluation in text for mexican magical towns, Procesamiento del Lenguaje Natural 75 (2025) 499–511.

[7] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3982–3992.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems (NeurIPS), volume 33, 2020, pp. 1877–1901.

[9] Google DeepMind, Gemini: A Family of Highly Capable Multimodal Models, Technical Report, Google DeepMind, 2024.