Hybrid Markov-BERT Approach For Synthetic Comment Generation

José Adrián Rodríguez González[1,*]

[1]Universidad de Guanajuato, México.

Abstract

There are several problems that have to be addressed in text generation, and one of them is making the text as human-like as possible, while ensuring that the text can reflect emotions. Touristic reviews are one example of how the emotions can be reflected directly in the text and as proposed in the Rest-MEX 2026 challenge. It proposes a framework based on three models; a Conditioned Markov model, a BERT based model and a hybrid model between these models. The Hybrid model obtained the best score tracking from the 3 models, achieving a track score of 0.4747 in the challenge. This approach explores the connection between classic and modern natural language processing approaches and how both approaches can be connected.

Keywords

BERT, Markov, text generation, Large Language Models, LLM, hybrid model, Natural Language Processing

1. Introduction

There are several problems that have to be addressed in text generation, across different domains, including issues such as coherence, and the human-like text is one of the most relevant topics to date. Tasks such as summarizing, free generation and translation are the most relevant [1]. Touristic comments are another research area that has its own challenges, such as the topic discovering [2], analyzing sentiments in Spanish reviews [3, 4]. So, generating comments, from a certain base text it represents another challenge, that has been addressed since the beginning of generative pretraining models [5]. A task proposed at IberLEF [6], in contrast to previous editions [7, 8, 9, 10] , tries to bring together these multiple areas in a task which introduces a fully generative evaluation paradigm for tourism-related natural language processing in Spanish.

This work proposes a hybrid model to address the task proposed, using Markov chain models, BERT based models and Large Language Models

2. Dataset Review

The task has a train dataset [11] with 208051 samples, divided into 6 columns:

- Title
- Review
- Polarity
- Town
- Region
- Type

So, we have the following variables Polarity that indicates the sentiment from a review, this can be defined as $(Polarity)$ Region, the region where the review was written denoted as $(Review)$ Type,

IberLEF 2026, September 2026, León, Spain

*Corresponding author.

ja.rodriguez.g@ugto.mx (J. A. R. González)

https://github.com/JoseAdrianRodriguezGonzalez (J. A. R. González)

0009-0009-7068-8852 (J. A. R. González)

the places where it takes the review $(Type)$ Town, the location where the review was written denoted as $(Town)$.

There are 40 different magic towns, but the most representative are shown in the figure 1, where most of the comments are from Tulum, *Isla mujeres* and *San Cristobal de las Casas*, these cities are from the south of Mexico, with a warm and wet weather [12] The polarity distribution in the complete dataset is

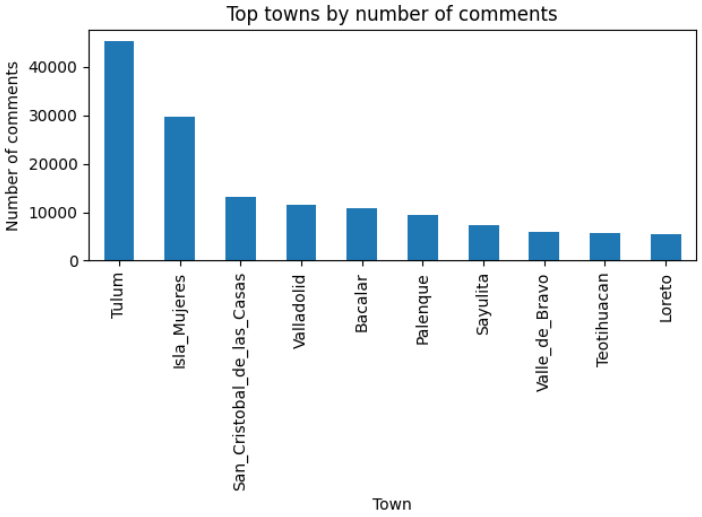

Figure 1: Top 10 towns with most reviews

shown in the figure 2, where the polarity 5, described as the most positive review, contains most of the comments, with the amount of 136561 comments, followed by the class 4.0. The rarest comments are the negatives. So the main challenge in review generation lies in these negative comments, because the distribution is clearly skewed into the value 5.0

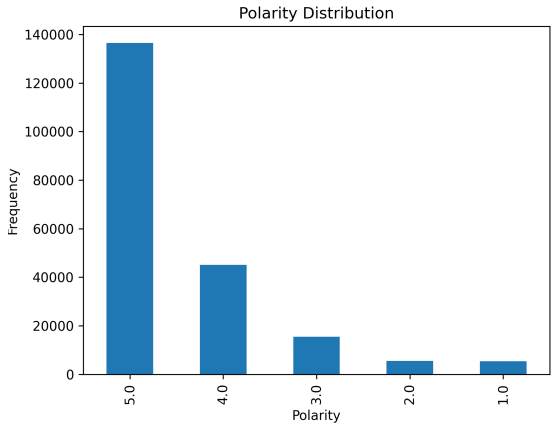

Figure 2: The bar-chart for the polarity distribution in the dataset

The most common regions where the comments belong are seen in the figure 3, where the 3 most common regions are Quintana Roo, Chiapas and Estado de Mexico, as also the next one are Yucatan, so the most common regions in the dataset are from the south region of Mexico and the less common are from the north

Let's analyze the mean token length given the polarity,defined as$\mathbb{E}[length|Polarity]$, this can be seen in the table 1

One observed pattern in the lengths is that the more positive the review is, fewer tokens are used. The variable given by type, is defined with three different elements, this refers to the main subject described

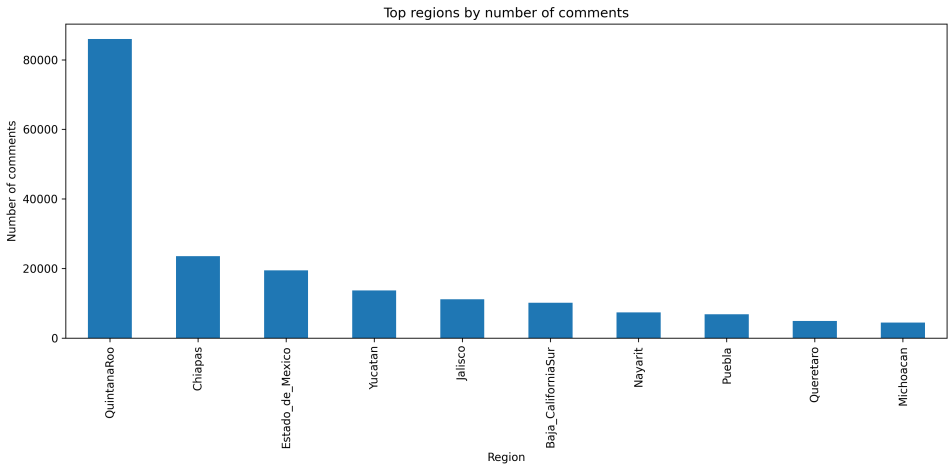

Figure 3: The barchart for the top 10 regions in the dataset

Polarity	length
1.0	80.54
2.0	81.10
3.0	70.62
4.0	64.42
5.0	57.87

Table 1
Table comparing the length given the polarity

Type	1.0	2.0	3.0	4.0	5.0
Attractive	868	1079	4895	16719	46360
Hotel	2086	2058	4622	11371	31273
Restaurant	2487	2359	6002	16944	58928

Table 2
Distribution of polarity scores by type

in the comment. In this case, there are 3 objects, the touristic attractive, this refers to monuments, the hotel and the restaurant. The restaurants according to the table 2 contains the most negative comments amongst the *Type* variable, meanwhile the attractive are the less negative comment. However the restaurants are also the type that has more positive comments, this indicates that this category of comments are the most polarized between the other two type of places. So, the problem can be seen that despite of the different variables, each polarity label has determined features from the other variables, these variables can help to generate more human reliable comments without losing the sense of the task.

3. Methodology

The main approach for the text generation can be seen in the figure 2. The previously described dataset is ingested into two models, where both outputs from both models are combined and refined into a Large Language Model (LLM) The classical model is instantiated as a Markov-based model. This follows a classic approach, where it has been proven into the text generation previously, such as texts in Russian [13], and the classic landmark proposed by Claude Shannon in 1949 [14]. The model is based on [15] where defines this type of Markov chain models.

On the other hand, the proposed BERT model based, is an embedding model that extracts features

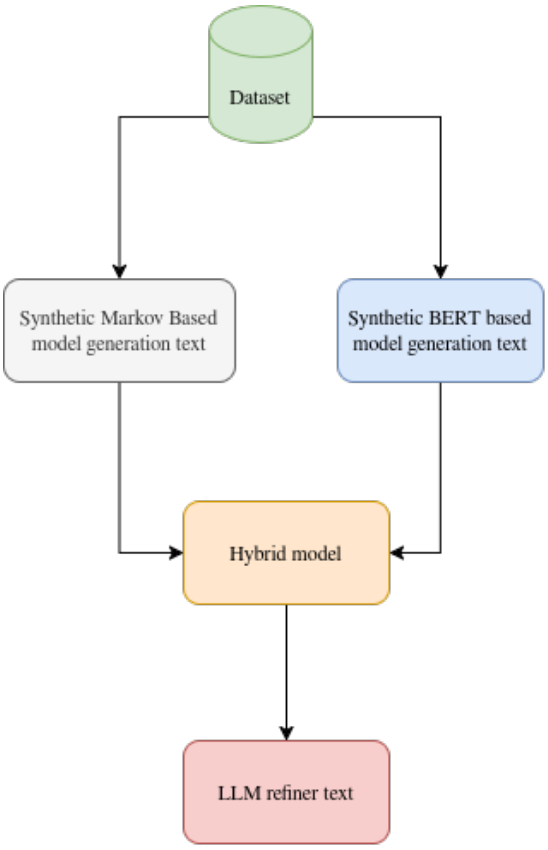

Figure 4: The main proposed pipeline of text generation

from its embeddings, and segments the space with unsupervised techniques, to find latent variables that could be helpful in the text generation.

3.1. Class Conditional Markov Text Generation Model

Let us define D as a set of reviews $D = \{d_1, d_2, \cdots, d_n\}$, where each document/review is associated with a polarity label defined as $y_i \in \mathcal{Y}$ defined previously as the following set $\{1, 2, 3, 4, 5\}$.

Each document is a sequence of words defined as the equation (1)

$$d_i = \{w_1, w_2, \cdots, w_T\} \tag{1}$$

Let us define a vocabulary as $\mathcal{V}$ where $w_T \in \mathcal{V}$

Initial Hypothesis The original objective can be expressed as a classic model such as the equation 2

$$P(Y \mid X) \tag{2}$$

Where Y represents the polarity labels and X the structured and unstructured variables. However, the text generation it is a high-dimensional problem that cannot be addressed solely with the structured variables from the original dataset, and also, not by features extracted from the reviews, so it requires more granularity to handle this problem, so it is necessary to reformulate the problem

Generative Reformulation So, the equation 3, is the baseline model:

$$P(d|y) = \prod_{t=1}^{T} P(w_t \mid w_{t-(n-1)}, \cdots, w_{t-1}, y) \tag{3}$$

The model calculates features defined as f, which are related to lemmas, dependencies, entities, length, sentence number, patterns, bi-grams, exclamations, questions, keywords, punctuation, negations, tokens, adjectives, verbs. And generates the vocabularies, in the equation 4

$$\mathcal{V} = \bigcup_{i=1}^{3} v_i \tag{4}$$

Where the $\mathcal{V}$ is the union of three vocabularies. These subsets are defined by certain type of words, for example, one vocabulary is for step-words, the second is for Part of Speech and the keywords.

For each polarity label will have frequency calculation, defined as it follows in the equation 5

$$P(w \mid y) = \frac{f_y(w)}{\sum_{w^T} f_y(w^T)} \tag{5}$$

But, the complete calculation of the n grams, are based on certain sequence denominated as $h \in \{w_1, \cdots, w_T\}$ where it filters the sequence depending on whether the raw tokens, or just the lemmas, and the n-gram model that will be used in the Markov model is calculated as in the equation 6

$$P(w \mid h, y) = \frac{C_y(h, w)}{C_y(h)} \tag{6}$$

Where C_y is the counting in a certain label. A feature vector is extracted, where it mixes the raw data, such as the Town, region and type of assessment reviewed and the semantic and lexical features extracted denominated as $c = \{\text{type}, \text{town}, \text{region}, \text{style}, \text{length}, \dots\}$. These features are used to generate another vector called context, where will help to the Markov model generator, what type of text to generate according to the previous features defined and 3 new ones, such as *negation strength* which describes how negative the comment is, the question ratio, where it says to the model, how many questions there are, and emotion level, where it says to the model, how many exclamations there are and how can influence the emotion in its review. The model will be like the equation 7

$$d \sim P(d \mid c, y) \tag{7}$$

To help the model to predict without struggling from scratch, with the help of n-grams and common narrative structures extracted from the text, it precalculates templates for the model, where it contains the backbone from the generation, these ones are based on how the language is structured in Spanish [16]. The equation 8 shows how a template could be chosen sampling it

$$T \sim P(T \mid c, \textbf{type}, y) \tag{8}$$

Markov Model Let that a sequence $d^{\mathcal{M}}$ is based on the histograms found in the equation 6. The model requires the count-based distributions that corresponds to the n-grams calculated in the equation 6. The equation 9 just described what type of counted-based distributions are based, such as uni-grams, bi-grams and tri-grams

$$C_y(w) \quad C_y(w_{t-1}, w_t) \quad C_y(w_{t-2}, w_{t-1}, w_t) \tag{9}$$

To control the randomness and diversity, a the temperature scaling is used, where it is viewed in the equation 10

$$\frac{C_y(h, w)^{1/T}}{\sum_{w^{total}} C_y(h, w^{total})^{1/T}} \tag{10}$$

To initialize the model it samples two initial words from the unigram distribution Finally the generation model can be seen as in the equation 11

$$d^{\mathcal{M}} = (w_1, w_2, \dots, w_T) \quad w_t \sim P(\cdot \mid h, y) \tag{11}$$

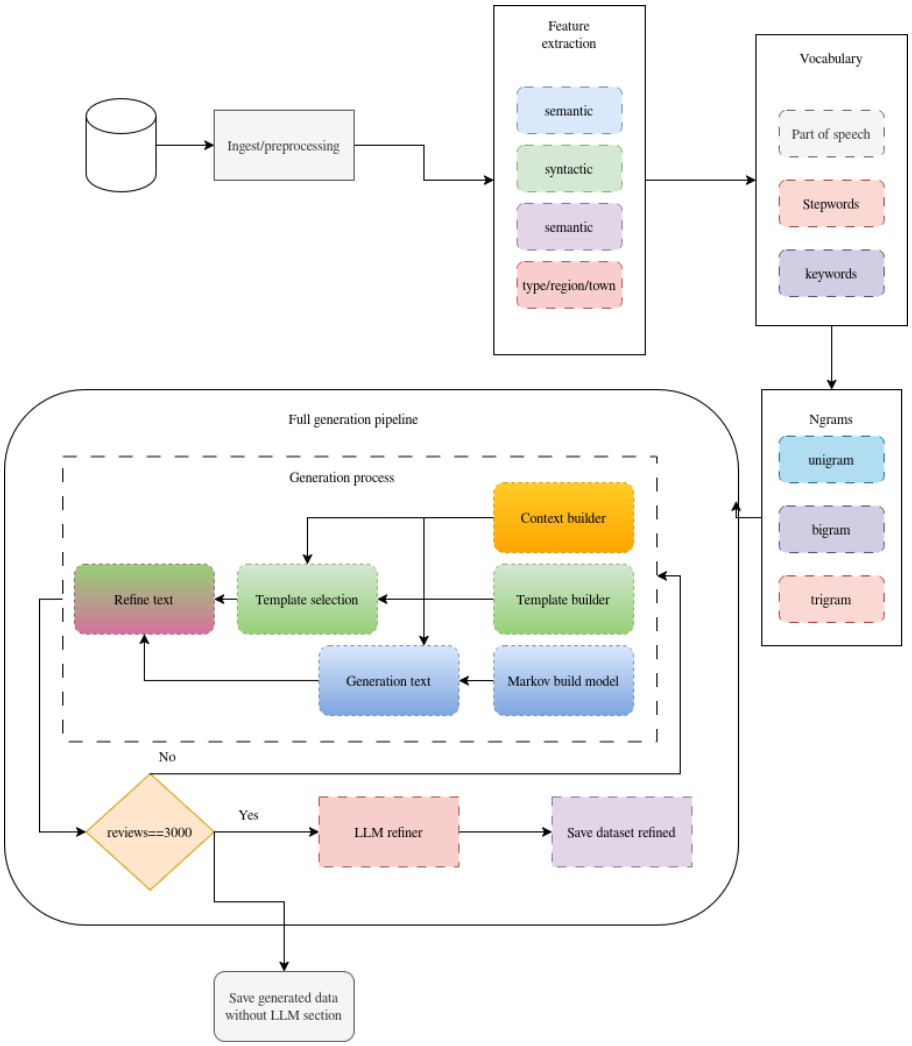

Figure 5: Full pipeline for Markov model

The initial Markov model text is refined into a function denominated as fill, where it grabs the template, the context vector and the text generated by Markov. Letting the generate document d_g as

$$d_g = \text{fill}(\text{template}, \text{context}, d^{\mathcal{M}}) \tag{12}$$

In the figure of the diagram 5, it shows the complete pipeline described before at its completeness. The ingest section it just joins the title with the review. The feature extraction comprehends the extraction of Lemmas, and other grammar rules. It is important to notice, that there are two paths at the final stages, until the desired number of reviews is reached . The first one its uses a LLM named Qwen 2.5 with 3 billions of parameters [17] for the refinement process and save the dataset in CSV format. But the alternative path it is to experiment it furthermore in the hybrid model.

3.2. BERT Model Proposal

The following model proposal is based on the encoder section from the transformer architecture, where it has been trained with a large corpus, BERT [18]. In the figure 6, the initial pipeline it follows a similar pipeline to the Markov model, however, instead of the semantic extraction by tokens, the first step is to use the BERT model with 4 sub-models. Each of them are

- paraphrase-multilingual-MiniLM-L12-v2,
- distiluse-base-multilingual-cased-v2",

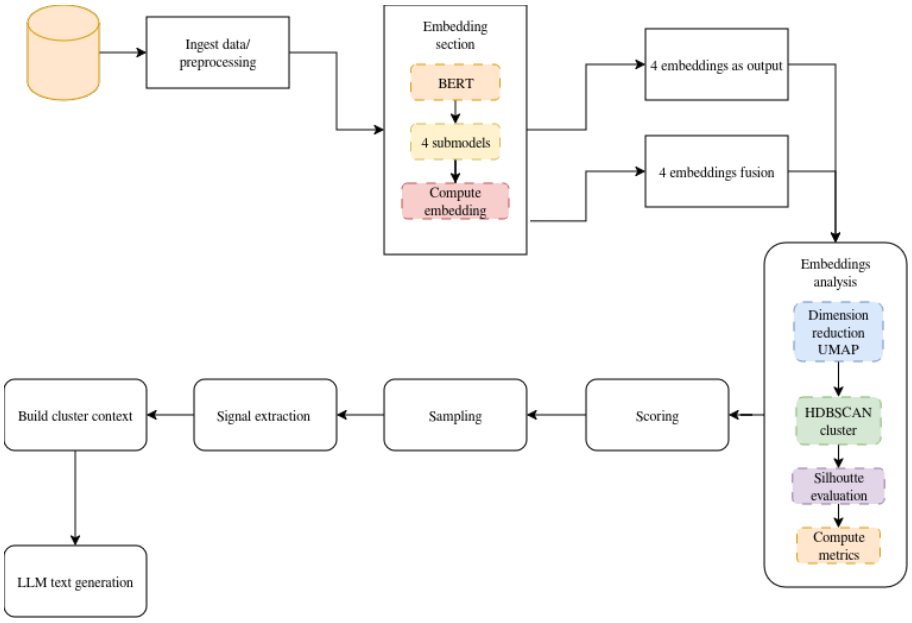

Figure 6: Full pipeline for BERT model

- all-mpnet-base-v2,
- hiiamsid/sentence_similarity_spanish_es,

Embedding These four models are based on three works, the sentence BERT embeddings [19], and the MP-Net model [20] and the BERT model [18]. All of these embeddings are stored in cache. Therefore, it fuses both matrices into a fused embedding map, and it is analyzed through four sub-tasks.

Embedding Analysis The dimension reduction with the UMAP technique [21], with the output reduced, it is analyzed with the unsupervised algorithm HDBSCAN [22]. This technique is applied to each embeddings output from the *embedding section* and it is evaluated with the Silhouette evaluation and the section defined as *compute metrics*, it calculates the size of the cluster, the amount of outliers, the noise ratio, the number of clusters, the intra and inter distance.

Scoring At the following stage, the scoring it evaluates with metrics from embeddings analysis, the equation 13, shows it as a linear combination of the three main elements extracted from the metrics. The silhouette represent how good the model was approximated into space data. And those variables were contains a value less than 0 multiplying the variable, where 0.5 has an equal weight between letting the noise and punishing it. Finally, the metric *penalty*, it refers to if HDBSCAN have not found any cluster or the amount of clusters is excessive, (more of 50 clusters)

$$\text{score} = \text{silhouette} - 0.5\text{noise\_ratio} - \mathit{penalty} \tag{13}$$

The score is ranked between the 5 embeddings, and it chooses the best embedding model and in based to the best one, the following stages will be applied to it.

Sampling The strategy is to perform semantically stratified sampling. It requires the three variables *region*, *Polarity* and the clusters obtained before, this groups the text into a certain context, this is modeled as $P(\text{region}, \text{Polarity}, \text{cluster})$. The sampling strategy performs local downsampling or for each group, this is because it can generate huge inner groups if it is not made. Furthermore, each group is concatenated and there is another sampling layer, where global sampling is performed. For clusters a reweighting is applied as $\frac{1}{|C|}$ where $\mid C \mid$ refers to number of samples with a certain cluster. If the

cluster number is bigger, the probability is less than if the cluster number is small. The whole strategy is similar to the approach proposed by Diaconis Persi [23], where it studies the cases where the data does not live in a Euclidean space, but in a manifold.

Signal Extraction After the embeddings were sampled, there are features that are extracted. The amount of clusters, the centroids of each cluster, the distance from each point to a certain cluster, the z score of each distance for each point to the center of the cluster. The size of the cluster. Also, as each point is related with a review, each review is associated with a certain region, so, it calculates the frequency for each region region in a cluster and the polarity cluster ratio.

The Builder Cluster Context In this section, it is calculated the most frequent region,polarity,the average distance and the average density points for each cluster

Lastly, these features and the embedding context it is passed through thee Large Language Model (LLM) and it generates the 3000 synthetic reviews.

3.3. Hybrid Model Proposal

The hybrid model connects the outputs from the Markov conditioned model generator, and the BERT embedding based model. The outputted text from the Markov model and the features from the BERT model are necessary to build the prompt and it generates the 3000 outputs.

3.4. Prompts for Each Model

The final stage of the proposed pipeline relies on a Large Language Model (LLM) to refine and generate synthetic reviews. Specifically, the model Qwen 2.5 (3B parameters) is used as a generative model conditioned on structured and unstructured inputs.

The prompts are designed to integrate three main sources of information:

- The Markov-generated text $d^{\mathcal{M}}$
- The structured context vector c (region, polarity, type, style, etc.)
- The semantic signals extracted from the embedding-based clustering

In the appendix, can be seen the complete prompt used for the hybrid model. Qwen by default, explains its responses so it is necessary to clarify that does not explain the review generated

4. Results

In the challenge [11], the final result were evaluated in a model to analyze if the reviews fits with every polarization well. The table 3 shows the global metrics for each result, where the Hybrid model proposal was the best one in every global metric except from the Mean Absolute Error, where the Markov model excels over BERT model and Hybrid. According to the track score from the challenge, the Markov model surpasses the BERT model in the track score, probably the result that obtained the Markov model from the Mean Absolute error which was responsible for boosting the result

Table 3
Global metrics for each model

Model	Track Score	Macro F1	Accuracy	Avg Precision	Avg Recall	MAE
Hybrid model	0.4747	0.5178	67.2611	0.5704	0.4877	0.3853
Markov model	0.4604	0.4844	54.4098	0.5712	0.4756	0.5249
BERT model	0.4464	0.4911	64.9519	0.5569	0.4706	0.3989

Each number in the variables noted as $(1), (2) \cdots (3)$ tables ,4,5 and ,6 refers to the polarity number. In the table 4, it can be seen the polarities 1 and 5 obtained the highest results in every model, but the polarity 2 obtained in every model the lowest f1 score. This is related to the unbalanced classes showed at the dataset review, and the model subsequently would generate few comments in this type of polarity,also the polarity 2 can be misplaced due to the shared similarities with the polarity 1 In the

Table 4
F1-score per class

Model	F1(1)	F1(2)	F1(3)	F1(4)	F1(5)
Hybrid model	0.5723	0.3149	0.4205	0.4669	0.8143
Markov model	0.5528	0.3809	0.4034	0.4233	0.6615
BERT model	0.5548	0.2256	0.4086	0.4822	0.7843

table 5 the pattern is the same, the highest results obtained were between the polarity 1 and 5 in hybrid model and BERT model. But in the Markov model shows the polarity 4 obtained 0.6340 of precision being the highest in the model. This result means that there are fewer false positives in the polarity 4 than in the polarity 5, meaning that the model could confuse the polarity 4 and 5 The table 6 shows the

Table 5
Precision per class

Model	P(1)	P(2)	P(3)	P(4)	P(5)
Hybrid model	0.6835	0.4453	0.4423	0.5242	0.7568
Markov model	0.5901	0.5858	0.5343	0.6340	0.5120
BERT model	0.8509	0.2250	0.3933	0.6170	0.6984

results for the recall metric and they follow the same pattern than the f1 score, where the higher values are in 5 and 1 polarity label.

Table 6
Recall per class

Model	R(1)	R(2)	R(3)	R(4)	R(5)
Hybrid model	0.4922	0.2436	0.4008	0.4208	0.8813
Markov model	0.5200	0.2822	0.3240	0.3177	0.9342
BERT model	0.4115	0.2261	0.4252	0.3958	0.8942

5. Discussion

These models struggled with polarity 2, and could be confused with polarity 1. The hybrid and Markov model had more difficulty with false negatives than with false positives, meaning that the review generated could be more likely to be another class and the model predicts it into another class, causing many real samples to be misclassified into other classes. With these metrics, it is clear that the three models have to be refined but the polarity 2 is a cases that has to be studied more, generate more reviews with this polarity and looking features that can discriminate better the nearest polarities. The polarities 3 and 4 have to be improved too, despite that both polarities followed similar metrics, that can lead the most of the confusions between both labels could result between these polarities.

6. Conclusions

The best model in the task was the hybrid model followed by the Markov model. However there is significant room for improvement, because the BERT model can be further refined, use another kind of transformer or by using BETO as an encoder. However, the Qwen model has few parameters, so that is another improvement and also, it could represent a bottleneck in refinement stage, so a model that could have more parameters can significantly improve the metrics as also, reviewing in more detail the labels 2,3 and 4. This approach bridges classical and modern natural language processing and how these models can compete and finally how the knowledge between both of them can be used in the text generation.

Declaration on Generative AI

During the preparation of this work, the author(s) used CHAT-GPT-5 in order to: Grammar and spelling check. 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] J. Becker, J. P. Wahle, B. Gipp, T. Ruas, Text generation: A systematic literature review of tasks, evaluation, and challenges, 2024. URL: https://arxiv.org/abs/2405.15604. arXiv:2405.15604.

[2] T. Zhang, V. T. Nguyen, T. Shimizu, Identifying tourist preferences from online reviews: An integrated bertopic-llm approach with bidirectional validation, Journal of Open Innovation: Technology, Market, and Complexity 12 (2026) 100757. URL: https://www.sciencedirect.com/science/article/pii/S2199853126000430. doi:https://doi.org/10.1016/j.joitmc.2026.100757.

[3] M. A. Carmona, A. Diaz-Pacheco, R. Aranda, A. Rodríguez González, V. Muniz, A. López-Monroy, F. Sánchez-Vega, L. Bustio-Martínez, Overview of rest-mex at iberlef 2023: Research on sentiment analysis task for mexican tourist texts (2023). doi:10.26342/2023-71-34.

[4] A. Díaz-Pacheco, A. B. García-Gutiérrez, A. Y. Rodríguez-González, R. Aranda, M. Á. Álvarez-Carmona, Balancing sentiment analysis datasets through representative-word-guided synthetic review generation: A case study on mexican spanish tourism reviews, Applied Sciences 16 (2026) 7398.

[5] A. Radford, K. Narasimhan, Improving language understanding by generative pre-training, 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[6] A. Bonet-Jover, J. A. Gonzalez-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.

[7] 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. Rodríguez-González, Overview of rest-mex at iberlef 2021: Recommendation system for text mexican tourism, Procesamiento del Lenguaje Natural 67 (2021). doi:https://doi.org/10.26342/2021-67-14.

[8] 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.

[9] 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.

[10] 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.
[11] M. A. Álvarez Carmona, A. Diaz-Pacheco, R. Aranda, M.-M. Reyes-Sierra, 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).
[12] J. Villaseñor, E. Ortiz, Biodiversidad de las plantas con flores (división magnoliophyta) en méxico, Revista Mexicana de Biodiversidad 85 (2013) S134–S142. doi:10.7550/rmb.31987.
[13] U. Yodgorov, Text generator based on markov chains, 2022.
[14] C. Shannon, W. Weaver, The Mathematical Theory of Communication, number v. 1 in Illini books, University of Illinois Press, 1949. URL: https://books.google.com.mx/books?id=dk0n_eGcqsUC.
[15] C. Bishop, Pattern Recognition and Machine Learning, Information Science and Statistics, Springer, 2006. URL: https://books.google.com.mx/books?id=kTNoQgAACAAJ.
[16] E. Cabrejas, Estructuras morfolÓgicas del idioma espaÑol: TeorÍa de los orÍgenes del espaÑol (primera parte), 2022. doi:10.13140/RG.2.2.35099.54560.
[17] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, Z. Qiu, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.
[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.
[19] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL: https://arxiv.org/abs/1908.10084. arXiv:1908.10084.
[20] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, 2020. URL: https://arxiv.org/abs/2004.09297. arXiv:2004.09297.
[21] L. McInnes, J. Healy, Umap: Uniform manifold approximation and projection for dimension reduction (2018). doi:10.48550/arXiv.1802.03426.
[22] C. Malzer, M. Baum, A hybrid approach to hierarchical density-based cluster selection, in: 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), IEEE, 2020, p. 223–228. URL: http://dx.doi.org/10.1109/MFI49285.2020.9235263. doi:10.1109/mfi49285.2020.9235263.
[23] P. Diaconis, S. Holmes, M. Shahshahani, Sampling from a manifold, 2012. URL: https://arxiv.org/abs/1206.6913. arXiv:1206.6913.

A. Hybrid Model Prompt

```
Eres un modelo experto en generación de reseñas realistas de usuarios.

Tu tarea es mejorar un comentario generado automáticamente,
haciéndolo más natural, coherente y humano.

### INSTRUCCIONES:

- Mantén el significado original del texto
- Respeta la polaridad (sentimiento): {data['polarity']}
- Usa el contexto semántico del cluster
- NO inventes información que no esté implícita
```

```
- Mejora fluidez, gramática y naturalidad
- El texto debe parecer escrito por una persona real

---

### TEXTO BASE (generado automáticamente):
{data['text']}

---

### CONTEXTO SEMÁNTICO DEL CLUSTER:
{data['cluster_summary']}

---

### SEÑALES SEMÁNTICAS:
- Densidad local: {data['cluster_features']['density']:.3f}
- Centralidad: {data['cluster_features']['centrality']:.3f}
- Z-score cluster: {data['cluster_features']['zscore']:.3f}

---

### OUTPUT:
Devuelve SOLO el texto final mejorado, sin explicaciones.
"""
```