Fing-Udelar-Students Participation in TA1C at IberLEF 2025: Clickbait Detection and Spoiling Shared Task

Miguelángel Díaz Cerecetto, Gastón Paiva

Instituto de Computación (InCo), Facultad de Ingeniería, Universidad de la República, Uruguay

Abstract

We present two competitive solutions in the *Te Ahorré Un Click* (TA1C) competition for (i) **clickbait detection** in Spanish tweets and (ii) **clickbait spoiling** given a headline and an article body. For the detection task, multiple BERT encoders were benchmarked, obtaining a blind-test macro-F_1 of 0.80480, ranking **4th** on the blind test set. On the spoiler generation task, we designed a pipeline and tested multiple axes of variation. These included the use of information retrieval (top-k chunks) versus no retrieval (using the full article on the prompt), prompt language (Spanish vs. English), number of few-shot demonstrations (0–3), and whether to inject definitions of *clickbait* and *spoiler*. Supervised fine-tuning and multiple inference generation, ranked by a separate "judge" model, provided a significant upgrade over the Gemini-1.5-flash-002 model. The final spoiling pipeline achieved a BLEU metric of 0.43589 on the blind test set, ranking **1st** in spoiler generation.

Keywords

Large Language Models, Spanish NLP, Clickbait Detection, Spoiler Generation, Chain-of-Thought Prompting, Retrieval-Augmented Generation, Supervised Fine-Tuning

1. Introduction

Clickbait is a widespread practice in online journalism: headlines or teasers deliberately omit a key piece of information in order to trigger curiosity and force the reader to click. Curiosity arises when people notice a gap between what they know and what they want to know; clickbait exploits that gap to capture traffic. The *Te Ahorré Un Click* (TA1C) shared task [1] at IberLEF 2025 [2] is, to the best of our knowledge, the first evaluation campaign that tackles clickbait **detection** and **spoiling** in Spanish, covering twelve national varieties. Task 1 is Clickbait Detection. Given a tweet and the full news article it links to, systems must decide whether the tweet is clickbait. Evaluation follows standard classification metrics: precision, recall and their harmonic mean F_1, averaged per class to obtain macro-F_1, which is the official task 1 metric. Precision measures the proportion of predicted clickbait tweets that are truly clickbait, while recall measures the proportion of actual clickbait tweets that are correctly detected; their combination in F_1 balances both aspects. Task 2 is Clickbait Spoiling. Systems must generate a short Spanish sentence (280 characters) that reveals the missing fact, or state that no answer is present in the article. Because multiple valid spoilers may exist, automatic evaluation relies on three overlap-based metrics: BLEU[3], ROUGE-L [4], and BERTScore[5]. After the automatic phase, the top systems will undergo a human assessment of fluency, accuracy and conciseness as described in the official website.

In this paper, Section 2 introduces the two TA1C subtasks and the corpora on which they are evaluated. Section 3 details our methodology for both clickbait detection and spoiler generation, including model architectures, prompt design and experimental protocol. Section 4 reports and analyses the results obtained in each task. Finally, Section 5 summarises the main findings and sketches directions for future work.

IberLEF 2025, September 2025, Zaragoza, Spain

miguelangel.diaz@fing.edu.uy (M. Díaz Cerecetto); gaston.paiva@fing.edu.uy (G. Paiva)

https://github.com/DiazCerecetto (M. Díaz Cerecetto); https://gitlab.fing.edu.uy/gaston.paiva (G. Paiva)

2. Task and Data Overview

2.1. Clickbait Detection Task

The corpus comprises 4200 Spanish tweets posted between October 2020 and October 2021 by 18 major media outlets, spanning twelve national varieties [6]. Each instance stores the tweet, the target URL and the cleaned HTML of the article (headline, sub-headline, body, captions and embedded links). Tweets were manually labeled. The split is 2800 training and 700 development tweets; a hidden set of 700 tweets is kept for the final leaderboard, which is ranked by macro-F_1. Table 1 shows a typical tuple.

Table 1
Example instance from the Clickbait detection train corpus.

TweetID	Date	Country	Teaser	Clickbait
1302968016477589504	2020-09-07	Uruguay	#SegundaDivisión | La fortaleza del ataque...	No

2.2. Spoiler Generation Task

A manually curated subset of 500 tweets (300 train, 100 dev and 100 test examples) extends the detection corpus with human-written spoilers that close the information gap. The answers cover factoids, summaries, lists and "no-answer" cases. Table 2 presents an abbreviated example. The full article text is stored as JSON and omitted here for space.

Table 2
Example instance from the Spoiler Generation subset.

Field	Value
TweetID	1355225506040438784
Date	2021-01-29
Teaser	*Cómo inversores aficionados se enfrentaron a …*
Spoiler	La clave son las "ventas en corto", en las que un fondo apuesta a que el precio caerá.

3. Methodology and experimental setup

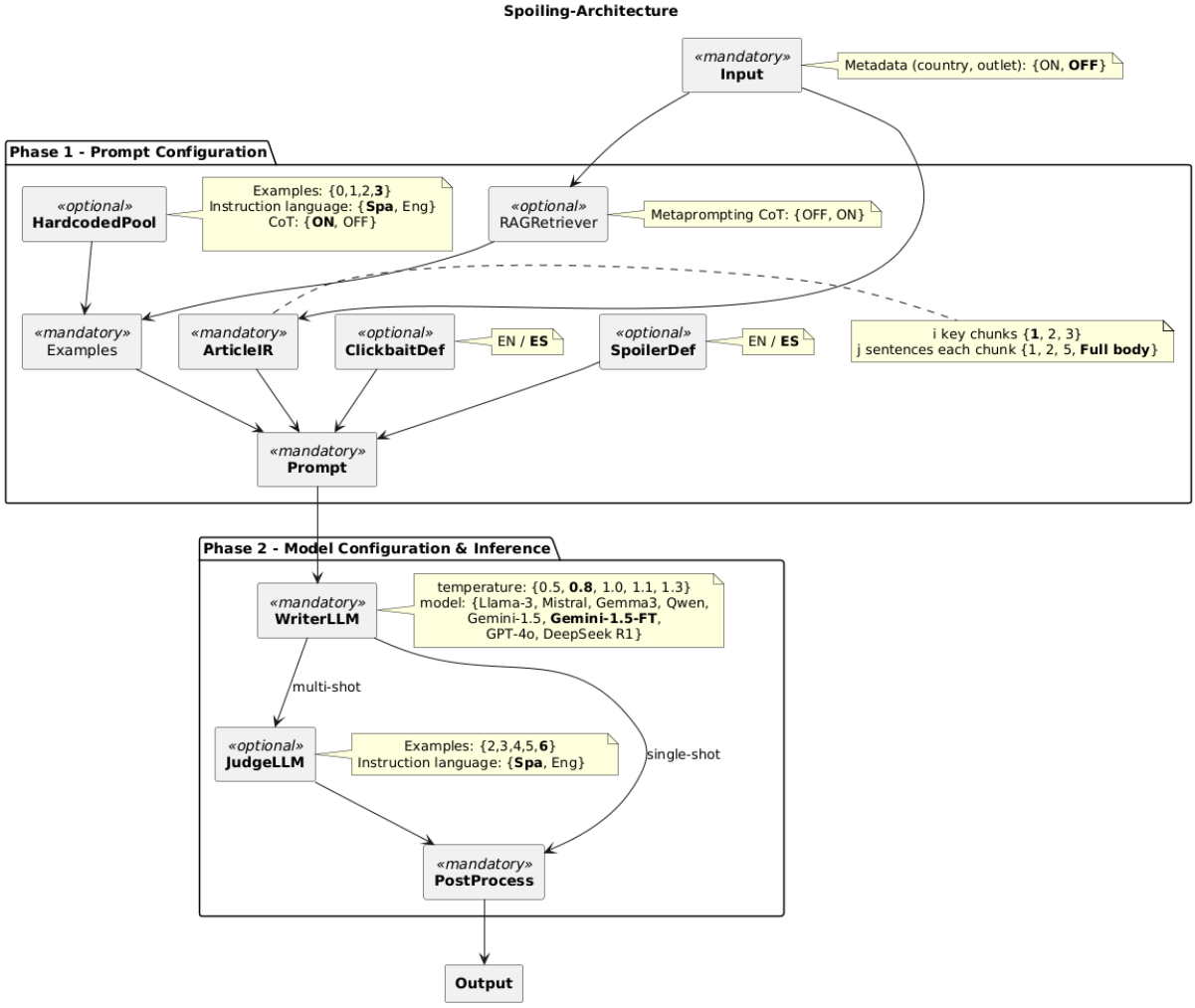
3.1. Clickbait Detection Methodology

Our detection pipeline is simple. We benchmarked a small pool of Spanish (and one multilingual) BERT-type encoders. The evaluated models were PlanTL-GOB-ES/roberta-large-bne[7], dccuchile/bert-base-spanish-wwm-cased[8], pln-udelar/rouberta-base-uy22-cased[9], and the multilingual xlm-roberta-base[10]. The same protocol was implemented for each model: batch size 16, max length 128, class-weighted cross-entropy, and five-fold cross-validation using the 2800 labeled training tweets. The model that obtained the highest macro-F_1 on the development fold was used for the blind-test submission.

3.2. Spoiler Generation Methodology

Figure 1 presents a detailed overview of our spoiler generation pipeline. The process begins with the user providing the input pair: a tweet and its linked article with optional metadata (country and outlet) that we choose to include in the article body if requested. This input is first preprocessed (e.g. truncation, format cleaning), after which the article is then split into i key chunks of j sentences each, using a semantic model, allowing the possibility of using the whole body in the prompt. Then, the system branches depending on whether KATE-style[11] retrieval is enabled. Unlike canonical RAG[12], which incorporates passages of external knowledge, KATE seeks relevant in-context examples.

Figure 1: Spoiling pipeline architecture, divided by two stages of testing, prompt configuration and Large Language Model (LLM) configuration and inference.



If KATE-style retrieval is active, the title is first embedded with all-MiniLM-L6-v2[13], and the k most similar training examples are retrieved via FAISS[14]. In the hard-coded path, 0-3 illustrative examples can be supplied, and each could be annotated with a handcrafted 5-step chain-of-thought to guide the model, whereas in the KATE path the retrieved instances are automatically enriched with meta-prompted CoT reasoning generated by the best local Large Language Model (LLM) benchmarked in evaluation phase 1.

Once the examples are selected, the system builds a complete prompt by concatenating a task-specific system message based in similar works[15], then the optional clickbait and spoiler definitions, also optional special case instructions (e.g., for tweets lacking an answer in the retrieved parts from the article), after that we include the selected examples, and finally the actual tweet–article pair to be spoiled. This full prompt, built as shown in Figure 2, is passed to a language model wrapper class, which can be configured to dispatch to any LLM provider services, local models or APIs.

In the second phase, we also introduced a two-stage generation and correction pipeline, and a postprocessing final step. At first, the chosen writer LLM produces distinct candidate spoilers. This generates M candidates that can present variations in their structure. Then, a dedicated judge LLM reviews all M outputs, choosing the best one according to a tiny prompt we wrote for this final step (Figure 3). We propose this fix as external to the final model, and present 4 final models for the evaluation phase, varying only this parameter. Finally, we use a postprocessing step to remove manually detected prompt-related inaccuracies generated by the models (e.g. "SALIDA ESPERADA:" message at the start).

Figure 2: Prompt skeleton, hardcoded version (spanish, k=3 examples, CoT, Clickbait definition, Spoiler definition).

```
--- SYSTEM ---
Eres un experto en redactar spoilers de clickbait para arruinar las noticias clickbait en español.

El clickbait es un método para generar titulares llamativos, especialmente en línea, que omite
deliberadamente parte de la información con el objetivo de generar curiosidad,
al crear una brecha de información, atrayendo así la atención de los lectores
y logrando que hagan clic.
Un spoiler de clickbait es una corta respuesta objetiva que acaba con la brecha
de información que el Tweet siembra y que, en el artículo, se resuelve.
La principal funcionalidad del spoiler es que el lector ya no necesite abrir la noticia.

Tu tarea es leer el titular y el artículo, y escribir una única oración en español
que revele claramente lo que el tweet insinúa. Busca en el artículo la frase o dato
que responda a la duda planteada, preferiblemente citando el texto original
entre comillas si se trata de una declaración. Sé conciso: redacta una sola
oración que contenga la idea central, sin repetir nombres, fechas, hashtags
ni frases o entidades ya presentes en el tweet.

Ten presente que pueden existir los siguientes casos particulares:
1. Que no exista respuesta en el conjunto. En ese caso debes responder
SOLAMENTE (sin comillas): No hay respuesta
2. Que la respuesta sea la nota completa. En ese caso debes responder
SOLAMENTE (sin comillas): La respuesta es la nota completa

--- EXAMPLES ---
Ejemplos (titular, artículo, frase reveladora):
TWEET: ...
NOTICIA ORIGINAL:

...

RAZONAMIENTO:
Paso 1: ¿Cuál es/son la(s) parte(s) oculta(s) a desvelar en el tuit?: ...
Paso 2: Por lo tanto, ¿qué pregunta(s) debemos responder?: ...
Paso 3: ¿Aparece(n) la(s) respuesta(s) en la noticia?
¿Dónde exactamente (cita breve o localiza el párrafo)?: ...
Paso 4: ¿Cómo resumirías esa(s) respuesta(s) sin repetir datos del tuit?: ...
Paso 5: Por lo tanto, la respuesta es: ...
SALIDA ESPERADA (debes escribir lo siguiente): ...

--- TASK ---
TWEET: {tweet}
NOTICIA ORIGINAL: {article}
SALIDA ESPERADA:
```

Figure 3: Judge prompt skeleton example, (Spanish, M=3).

```
Elige el mejor spoiler entre los 3 candidatos. Criterio:
Responde la incógnita del tuit de forma correcta, concisa y sin repetir literalmente el tuit.

Tweet:
¿Sabes qué jugador fue transferido por cifra récord?

Artículo (referencia):
El club confirmó que Juan Pérez fue traspasado al Real FC por 50 millones
de euros, estableciendo un nuevo récord en la liga...

Candidatos:
1. Fue Juan Pérez, vendido por 50 millones de euros.
2. El club no ha confirmado ninguna venta.
3. Se trata de Luis Gómez.

Respuesta:
```

3.2.1. Experimental setup

In the detection task, we evaluated a range of Spanish BERT variants in the clickbait detection task using a 5-fold cross-validation protocol with class-weighted loss, using both train and val datasets, training each model for three epochs at a learning rate of 2×10^{-5}, and a batch size of 16.

For the spoiling task, the evaluation was split into three phases. At the very start of the competition, we set aside 60 training instances and benchmarked five candidates' LLMs. Then, we chose the best local model and then began refining the prompt using this model. Prompt configuration included evaluation for article information retrieval, few-shot example building variants, clickbait and spoiler definitions, and prompt language. When prompt construction was done, multiple LLM temperatures were evaluated by a sweep, and supervised fine-tuning was also explored.

4. Results and discussion

4.1. Classification

The best base model PlanTL-GOB-ES/roberta-large-bne[7] achieved a macro-F_1 of 0.8928 ± 0.0096 in this search step. Table 3 summarizes the mean and standard deviation of macro-F_1 across the folds. We ultimately deployed the best-performing model from our evaluation phase, which delivered an F1 score of 0.80480 on the test dataset[1].

Table 3
5-fold cross-validation results (macro-F_1) for different BERT variants, using both train and val datasets concatenated.

Model	Mean Macro-F_1	Std. Dev.
xlm-roberta-base	0.8686	0.0109
dccuchile/bert-base-spanish-wwm-cased	0.8779	0.0076
pln-udelar/rouberta-base-uy22-cased	0.8754	0.0119
PlanTL-GOB-ES/roberta-large-bne	**0.8928**	0.0096

Table 4
Final detection model proposed and their performance on test set

Model	Precision	Recall	F1
PlanTL-GOB-ES/roberta-large-bne	**0.8072**	0.8024	**0.8048**
PlanTL-GOB-ES/roberta-base-bne	0.7039	**0.8683**	0.7775

4.2. Spoiling

4.2.1. Model selection

Table 5 shows how metrics behaved when we compared five candidate large language models (LLMs): Deepseek-R1[16] API, Gemma-3 12B[17], Deepseek-R1 14B, Phi-4[18], and ClickbaitFighter 10B[15]. While Deepseek-R1 achieved the highest BLEU and ROUGE-L scores, it is only available through a rate-limited API, which made iterative prompt tuning impractical. Consequently, we selected Gemma-3 12B as our base model since it was the best-performing locally executable model in Google Colab.

4.2.2. Prompt construction

We then refined the prompting strategy using Gemma-3 12B. Initially, we compared three article retrieval methods: using the full article versus using smaller fragments (five fragments of two sentences

[1] https://huggingface.co/dcere/ta1c-Clickbait-Detector-es-large

Table 5
Baseline performance on a 60-instance development slice.

Model	BLEU	ROUGE-L
Deepseek-R1 (API)	**0.1687**	**0.2034**
Gemma-3 12B	0.1399	0.1881
Deepseek-R1 14B	0.1363	0.1518
Phi-4	0.1224	0.1238
ClickbaitFighter 10B	0.0370	0.1205

or two fragments of two sentences each). As shown in 6, providing the model with the complete article consistently outperformed fragmented versions, which frequently omitted critical information, causing the model to mistakenly predict the fallback answer *"No hay respuesta"*.

Table 6
Impact of article fragmentation on validation metrics (60 instances).

Configuration	BLEU	ROUGE-L	$\text{BERTScore}_{\text{F1}}$
Full article (no IR)	**0.1323**	**0.2040**	**0.6856**
5 fragments, 2 sentences each	0.1180	0.1864	0.6773
2 fragments, 2 sentences each	0.1010	0.1690	0.6727

Next, we examined few-shot prompting strategies. The KATE-style example retrieval—MiniLM embeddings [13] with FAISS [14]— was compared against a simpler pool of manually curated examples. Although similarity-based retrieval chose tweets lexically closer to the test headline, these KATE prompts did not always capture diverse reasoning patterns. In contrast, manually curated examples augmented with an explicit five-step Chain-of-Thought (CoT) rationale achieved consistently higher scores, echoing the CoT-over-KATE outperforming reported by Fu et al. (2023) [19]. Consequently, we adopted a fixed trio of hardcoded CoT-enhanced examples for the final system.

We also assessed the impact of including explicit definitions of *clickbait* and *spoiler*. Although adding these definitions did not noticeably alter performance metrics, we included them in the final pipeline since they provided explicit and self-contained context, potentially aiding the model's generalization and interpretability. Lastly, we evaluated whether switching the prompt language from Spanish to English affected performance. This comparison revealed negligible differences (less than 0.1 points difference), leading us to retain Spanish for practical reasons, such as ease of human evaluation, reduced code-switching artifacts, and simplified prompt quality control.

4.2.3. Language model tuning

Once we had identified the best-performing prompt for Gemma-3, fine-tuning Gemma-3 was infeasible within the competition's tight timeline, we instead supervised-fine-tuned Gemini-1.5 with the winning prompt from the previous phase. The resulting model became the core of our system.

Using this fine-tuned Gemini-1.5, we conducted temperature sweeps and post-processing experiments. We found that $t = 0.80$ produced the best results, whereas the optimal temperature for base Gemma-3 had been $t = 1.0$. This upgrade lifted BLEU on the development set from 0.2978 (base Gemma-3 + best prompt) to 0.4201.

With the writer model finalized, we next generated N candidate outputs and re-ranked them with a second Gemini-1.5 acting as a judge. Generating six candidates and selecting the best one raised BLEU on the test set by +4.7 points compared with single-shot decoding. Manual probing showed that a high-quality spoiler often appeared only on the third, fourth, or even sixth attempt; the first answer frequently missed key facts or omitted information. By letting the writer explore a small diversity of outputs and allowing the judge to pick the most concise and faithful candidate, we systematically increased the chance of capturing the correct answer gap.

Our final spoiler-generation model used a Gemini-1.5-flash-002 base, fine-tuned in supervised mode for 10 steps, with a learning-rate multiplier of 1 and an adapter size of 8, without intermediate checkpoints. A simplified code implementation with the winning model is available in a companion notebook[2].

Table 7
Final spoiling models proposed and their performance on test set

Model	BLEU	ROUGE-L	BERTScore
NoIR-Hardcode-Defs-GeminiFT-NoJudge	0.3886	0.7517	0.7247
NoIR-Hardcode-Defs-GeminiFT-Judge3	0.4254	**0.7606**	**0.7410**
NoIR-Hardcode-Defs-GeminiFT-Judge5	0.4261	0.7527	0.7354
NoIR-Hardcode-Defs-GeminiFT-Judge6	**0.4359**	0.7497	0.7374

5. Conclusions and Future Work

This study introduced a simple approach to Spanish clickbait detection and a competitive architecture for spoiler generation, ranking fourth in classification and first in spoiling. Supplying the language model with the full, cleaned article consistently outperformed any information-retrieval strategy that restricted context, because crucial details often appear late in the body and a truncated window pushed the model toward spurious "No hay respuesta" replies. A small, manually balanced set of Spanish examples, each enriched with an explicit chain of thought, guided the model more effectively than retrieval-based prompts whose similarity was measured only at the lexical level; the benefit stems from showcasing diverse reasoning patterns rather than surface overlap. Finally, replacing the base Gemma-3 with a supervised fine-tuned Gemini-1.5 and re-ranking multiple candidate outputs with a second "judge" instance yielded a cumulative gain of more than 13 BLEU points. This shows the value of supervised fine-tuning.

Future research could explore multimodal LLMs: some training tweets contained spoilers available only through an embedded image (see Table 8), we removed them from the train and val dataset before performing the fine-tuning. Injecting an automatic caption of that image obtained with a multi-modal LLM into the textual context could unlock currently unsolved cases. Most prompt-engineering choices and decoding hyper-parameters (temperature, number of candidates, prompt language) were tuned on the intermediate Gemma-3 model and only quick-checked once the pipeline migrated to the fine-tuned Gemini-1.5 writer + judge. A systematic, model-specific search (e.g., grid search) directly on Gemini—could reveal combinations better aligned with the newer model's behavior and yield additional gains in both automatic scores and human-rated fluency.

Table 8
Example instance whose spoiler is only an image.

Field	**Value**
TweetID	1333698909747748865
Date	2020-12-01
Teaser	*Maradona eterno en Boca: el mural que empezó ...*
Spoiler	<an URL leading to an image>

6. Declaration on Generative AI

During the preparation of this work, the authors used two generative-AI tools—OpenAI GPT-4o and OpenAI o3— for copy-editing tasks (grammar and spelling). All AI-suggested edits were reviewed

[2]https://colab.research.google.com/drive/1qnfduoTNdUTE-Tnb8kCZfFGR-ujwWiOW?usp=sharing

and, where necessary, modified by the authors, who accept full responsibility for every aspect of this publication.

References

[1] G. Mordecki, L. Chiruzzo, R. Laguna, J. Prada, A. Rosá, I. Sastre, G. Moncecchi, Overview of TA1C at IberLEF 2025: Detecting and Spoiling Clickbait in Spanish-Language News, Procesamiento del Lenguaje Natural 75 (2025).

[2] J. Á. González-Barba, L. Chiruzzo, S. M. Jiménez-Zafra, Overview of IberLEF 2025: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.

[3] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, PA, 2002, pp. 311–318.

[4] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out (Post-ACL Workshop), Barcelona, Spain, 2004, pp. 74–81.

[5] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: International Conference on Learning Representations (ICLR 2020), 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[6] G. Mordecki, G. Moncecchi, J. Couto, Te ahorré un click: A revised definition of clickbait and detection in spanish news, in: L. Correia, A. Rosá, F. Garijo (Eds.), Advances in Artificial Intelligence – IBERAMIA 2024, Springer Nature Switzerland, Cham, 2025, pp. 387–399.

[7] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. Rodríguez-Penagos, A. Gonzalez-Agirre, M. Villegas, MarIA: Spanish language models, Procesamiento del Lenguaje Natural 68 (2022) 39–60.

[8] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: PML4DC at ICLR 2020, 2020.

[9] J. P. Filevich, G. Marco, S. Castro, L. Chiruzzo, A. Rosá, A language model trained on uruguayan spanish news text, in: Proceedings of the Second International Workshop Towards Digital Language Equality (TDLE): Focusing on Sustainability@ LREC-COLING 2024, 2024, pp. 53–60.

[10] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, CoRR abs/1911.02116 (2019). URL: http://arxiv.org/abs/1911.02116. arXiv:1911.02116.

[11] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, W. Chen, What makes good in-context examples for gpt-3?, 2021. URL: https://arxiv.org/abs/2101.06804. arXiv:2101.06804.

[12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.

[13] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. URL: https://arxiv.org/abs/2002.10957. arXiv:2002.10957.

[14] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library (2024). arXiv:2401.08281.

[15] I. García-Ferrero, B. Altuna, Noticia: A clickbait article summarization dataset in spanish, 2024. arXiv:2404.07611.

[16] DeepSeek-AI, Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL: https://arxiv.org/abs/2501.12948. arXiv:2501.12948.

[17] G. Team, Gemma 3 (2025). URL: https://goo.gle/Gemma3Report.

[18] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, J. R. Lee, Y. T. Lee, Y. Li, W. Liu, C. C. T. Mendes, A. Nguyen, E. Price,

G. de Rosa, O. Saarikivi, A. Salim, S. Shah, X. Wang, R. Ward, Y. Wu, D. Yu, C. Zhang, Y. Zhang, Phi-4 technical report, 2024. URL: https://arxiv.org/abs/2412.08905. arXiv:2412.08905.

[19] P. Fu, Y. Zhang, H. Wang, W. Qiu, J. Zhao, Revisiting the knowledge injection frameworks, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 10983–10997. URL: https://aclanthology.org/2023.emnlp-main.677/. doi:10.18653/v1/2023.emnlp-main.677.