# TUKE Satira at the CLEF 2026 JOKER Track: Constant-Variable Optimization for English-French Pun Translation

Satira at CLEF 2026 JOKER Task 2

Ruslan Pylypiv[1,*], Maksym Kuznietsov[1] and Martina Szabóová[1,*]

[1]Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia

**Abstract**

Translating puns across languages requires preserving not only the meaning but also the wordplay itself, which typically relies on language-specific ambiguities and therefore resists direct translation. This paper describes the participation of the Satira team from the Technical University of Košice in Task 2 (Pun Translation) of the CLEF 2026 JOKER track, which asks systems to translate English punning jokes into French. We embed the Constant-Variable Optimization (CVO) prompting strategy, adapted from the Pun2Pun benchmark, into a retrieval-augmented, multi-candidate generate-and-judge pipeline, and extend it with a Delabastita-based pun-type classifier that routes each joke to a type-specialized decomposition prompt. By toggling retrieval, classification, and CVO we evaluate five system configurations—a $2 \times 2$ ablation over the retrieval pipeline plus a non-retrieval direct-CVO reference—on the official location-based metric over the JOKER 2026 test set of 4,061 puns. Our clearest finding is a negative one: the simplest configuration—direct CVO with no retrieval or classification—performs best (33.16, fifth place in the final official ranking), while each added pipeline component lowers the score. This suggests that, for positional pun preservation, a focused single-translation strategy that explicitly tracks the punning word outperforms candidate generation and selection.

**Keywords**

pun translation, wordplay, large language models, retrieval-augmented generation, CVO, JOKER, CLEF 2026

## 1. Introduction

Humour is deeply embedded in human communication, yet it remains one of the most challenging phenomena for natural language processing systems. Puns—wordplay that exploits multiple meanings or similar sounds of words—are particularly difficult to handle computationally because they rely on language-specific features such as polysemy, homophony, and cultural associations that often lack direct equivalents in other languages [3, 1].

The CLEF JOKER Lab, now in its fifth year, has established itself as the primary venue for shared tasks on computational humour, bringing together researchers from computational linguistics, translation studies, and artificial intelligence [5, 6, 7]. The 2026 edition [20] continues the wordplay translation task (Task 2) [21], which challenges participants to translate English puns into French in a way that preserves both the semantic content and the humorous wordplay effect. This year, the task has been expanded to also include Spanish as a target language.

Pun translation has long been considered a "mission impossible" in translation studies [9], particularly between languages with different phonetic and morphological systems. Traditional approaches resort to compromises such as replacing the pun with a non-pun equivalent, using a different rhetorical device, or simply omitting the wordplay entirely [2, 3]. The advent of large language models (LLMs) has opened new possibilities for creative text generation, but research shows that even state-of-the-art models struggle to consistently preserve wordplay across linguistic boundaries [12, 4, 8].

*CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany*

*Corresponding author.

ruslan.pylypiv@student.tuke.sk (R. Pylypiv); maksym.kuznietsov@student.tuke.sk (M. Kuznietsov); martina.szaboova@tuke.sk (M. Szabóová)

In this work, we build on the Constant-Variable Optimization (CVO) model, originally proposed by Zhao and An [15] in translation studies and recently adapted for computational use in the Pun2Pun benchmark [8]. The CVO framework provides a structured decomposition of a source pun into three semantic and pragmatic components (the *constants*), which are then systematically mapped onto target-language equivalents (the *variables*) through enumeration, reconstruction, and optimization stages. Where no direct cross-lingual equivalent exists, CVO encourages *mechanism interchange*—for instance, converting a homophonic pun into a homographic one—to preserve the humorous effect through alternative linguistic means.

Rather than treating CVO as a standalone prompting trick, we embed it in a retrieval-augmented generate-and-judge pipeline and study, in a controlled manner, how much each component contributes to English–French pun translation. Concretely, our system (i) retrieves semantically similar human pun translations from the JOKER training data and supplies them as in-context examples; (ii) prompts a 123B-parameter instruction-tuned generator to produce several candidate French translations through a structured decomposition; and (iii) uses a smaller LLM as a judge to score the candidates and select the best one. On top of this pipeline we introduce two orthogonal mechanisms: a *pun-type classifier* that, following Delabastita's typology [2, 3], routes each joke (lexical, phonetic, or idiomatic) to a type-specialized decomposition prompt; and the *CVO decomposition* described above. By independently switching these two mechanisms on and off, we obtain a $2 \times 2$ ablation over the retrieval-augmented pipeline, complemented by a non-retrieval direct-CVO reference system—five configurations in total. This design lets us isolate the effect of retrieval, pun-type routing, and CVO-guided reasoning rather than reporting a single end-to-end number.

The contributions of this paper are as follows:

- We present the first application, to our knowledge, of the CVO prompting strategy to the English–French pun translation task in a shared-task setting.
- We integrate CVO into a retrieval-augmented, multi-candidate generate-and-judge pipeline and extend it with a Delabastita-based pun-type classifier that selects a specialized translation strategy per pun type.
- We conduct a systematic ablation across five system configurations, disentangling the individual contributions of retrieval, pun-type classification, and CVO-guided decomposition.
- We evaluate all five configurations on the official location-based metric of the JOKER 2026 test set and report a controlled within-task comparison, rather than relying on reference-based metrics that are unavailable for the 2026 test data.

The remainder of this paper is organized as follows. Section 2 reviews related work on computational pun translation. Section 3 describes the task definition, dataset, and evaluation metrics. Section 4 presents our methodology and the five system configurations. Section 5 reports our experimental results, and Section 6 concludes with a discussion of findings and future directions.

## 2. Related Work

### 2.1. The CLEF JOKER Task

The JOKER Lab was established at CLEF 2022 as a workshop on automatic wordplay and humour translation [5]. Since then, it has grown into a multi-task evaluation campaign encompassing humour-aware information retrieval, pun detection and interpretation, and pun translation. The pun translation task (Task 2 in recent editions) has been a staple since the Lab's inception, consistently challenging participants to translate English puns into French while preserving both meaning and wordplay.

In CLEF 2025, nine teams submitted 52 runs for Task 2 [7]. The task corpus was expanded with 1,682 new source texts and 2,615 professional French reference translations. Evaluation combined standard automated metrics (BLEU, BERTScore), a novel location-based metric designed specifically for wordplay preservation, and manual human evaluation by expert annotators; a detailed description of these metrics

is provided in Section 3.3. Notably, the organizers found that BLEU and BERTScore rankings diverged substantially from human judgments, while the location-based metric correlated strongly with expert scores (Pearson $r = 0.84$), underscoring the need for humour-aware evaluation approaches.

## 2.2. Approaches in CLEF JOKER 2025

The participating teams in JOKER 2025 Task 2 employed a diverse range of methods.

**Fine-tuned LLMs.** The Skommarkhos team [16] fine-tuned Lucie-7B-Instruct and CroissantLLMChat using supervised fine-tuning (SFT) combined with Adaptive Rejection Preference Optimization (ARPO). Their SFT approach achieved the highest BLEU score of 43.20, though the authors noted that it tended to produce overly literal translations.

**Post-hoc pun detection filtering.** The University of Amsterdam team [17] developed pun detection classifiers for both English and French, generating multiple translation candidates and selecting the one with the highest pun score. This filtering approach showed promise for ensuring humour preservation, though with limited gains on automated metrics.

**Multi-agent LLM systems with phonetic-semantic embeddings.** The rdtaylorjr team [18] implemented a multi-agent approach using contrastive learning and phonetic-semantic embeddings. Their system used a generator LLM followed by a discriminator model to verify pun presence, iterating up to ten times per translation. Crucially, their guided chain-of-thought approach with phonetic-semantic embeddings outperformed all other teams in human evaluation, despite scoring near the bottom on BLEU, highlighting the fundamental tension between literalness (as measured by BLEU) and creative humour preservation.

Two of these ideas map directly onto components we study in this work. The University of Amsterdam's strategy of generating several candidates and keeping the one with the highest pun score corresponds to the judge-based selection step in our retrieval-augmented pipeline,while rdtaylorjr's generator-plus-discriminator loop corresponds to our generate-and-judge design. Our work differs in two respects: we ground generation in the linguistically motivated CVO decomposition rather than in contrastive embeddings, and we add an explicit pun-type classification step that routes each joke to a specialized translation strategy. By evaluating these components in a controlled ablation (Section 4), we aim to quantify how much each of them actually contributes, rather than reporting a single end-to-end system.

## 2.3. The Pun2Pun Benchmark and CVO

Ma et al. [8] introduced the Pun2Pun benchmark for evaluating cross-lingual pun translation between Chinese and English. A key contribution of their work is the adaptation of the Constant-Variable Optimization (CVO) model from translation studies [15, 14] into a computational prompting strategy. The CVO framework decomposes the source pun into three "constants"—the core punning word (SM1), the contextual framework (SM2), and the overall pragmatic effect (SPM)—and maps them to three target-language "variables" (TM1, TM2, TPM) through a structured process of enumeration, reconstruction, and optimization.

To illustrate, consider the English pun *"I used to be a banker but I lost interest."* The CVO decomposition proceeds as follows. The core punning word is SM1 = *interest*, which carries two meanings: a financial rate and personal enthusiasm. The contextual framework SM2 consists of the Anchor *banker* (which primes the financial reading) and the Bridge *lost* (which activates the second, emotional reading). The source pragmatic meaning SPM emerges from the interplay of both senses—losing financial interest and losing personal enthusiasm simultaneously—producing the humorous effect. In the target language (French), the CVO framework maps these constants to variables: TM1 = *intérêt*, which conveniently preserves the same dual meaning in French (financial interest and personal interest); TM2 maps the Anchor to *banquier* and the Bridge to *perdu*; and the target pragmatic meaning TPM reproduces the same humour mechanism, yielding *"J'ai été banquier mais j'en ai perdu tout l'intérêt."* This example represents an ideal case where the punning word has a direct cross-lingual equivalent. When no such equivalent

exists, the CVO framework instead encourages mechanism interchange—for instance, converting a homophonic pun into a homographic one—to preserve the humour effect through alternative linguistic means.

The experiments of Ma et al. across seven LLMs demonstrated that the CVO strategy generally improved translation quality compared to vanilla zero-shot and one-shot baselines, particularly for stronger models, and that the Overlap (Ovl) metric derived from the CVO framework provided a more nuanced assessment of translation quality than traditional metrics [8]. Importantly, their findings showed that effective pun translation often requires greater semantic divergence from the source text, and that mechanism interchange can be an effective cross-linguistic strategy.

We adopt the CVO prompting strategy from the Pun2Pun framework and apply it to the English–French language pair in JOKER 2026 Task 2. To our knowledge, this is the first application of the CVO approach to English–French pun translation in a shared-task setting, and the first to embed it inside a retrieval-augmented, classifier-routed generate-and-judge pipeline.

## 3. Task Description, Dataset, and Evaluation Metrics

### 3.1. Task Definition

JOKER 2026 Task 2 [21] requires participants to translate English punning jokes into French while preserving both the meaning and the wordplay of the original. The input is an English sentence containing a pun (homophonic or homographic), and the expected output is a French translation that maintains the humorous double meaning. Participants may convert a homophonic pun into a homographic pun, or vice versa, as long as the wordplay effect is preserved in French. The 2026 edition additionally introduces Spanish as a target language; in this work we focus exclusively on the English-to-French direction.

### 3.2. Dataset

We use the data released for JOKER 2026 Task 2, which builds on the corpus curated by the organizers across the previous editions of the shared task. The training data consists of **1,682** distinct English pun sentences paired with **2,615** professional French reference translations [7]. The number of French reference translations per English source pun varies from 1 to 9, as shown in Table 1, reflecting the inherently creative and open-ended nature of pun translation: there is no single "correct" answer, but rather a space of acceptable solutions that preserve the wordplay effect through different strategies. The majority of puns (1,252 out of 1,682) have only a single reference translation, while a smaller subset admits up to nine alternative renderings. Table 1 also reports the number of distinct pun *locations* across translations: most puns (1,382) have a single position at which the wordplay is realised, though some allow for multiple alternative wordplay positions in French.

**Table 1**
Distribution of the number of French reference translations and distinct pun locations per English source pun in the JOKER 2026 Task 2 training data [7].

| # per pun | References | Locations |
|---|---|---|
| 1 | 1,252 | 1,382 |
| 2 | 172 | 220 |
| 3 | 133 | 68 |
| 4 | 53 | 10 |
| 5 | 39 | 1 |
| 6 | 22 | 1 |
| 7 | 8 | — |
| 8 | 2 | — |
| 9 | 1 | — |

The puns in the dataset include both homophonic puns, which exploit words with similar or identical pronunciation but different meanings, and homographic puns, which exploit words with identical spelling but multiple semantic senses (e.g., "interest" meaning both a financial rate and personal enthusiasm). The reference translations were produced by professional French native-speaker translators across multiple editions of the shared task (2022–2025).

For evaluation, the organizers release a separate test set of **4,061** English source puns. The test set is provided *without* French reference translations, so reference-based metrics cannot be computed by participants on the test data; system quality on the test set is assessed by the organizers through the official location-based metric (Section 3.3) and, ultimately, by manual human evaluation. We further note that a substantial portion of the training puns also occur within the 4,061-pun test set, so for those items the retrieval-augmented configurations can surface a known reference translation, whereas the non-retrieval direct configuration receives no such signal; this is relevant when interpreting the results in Section 5.

### 3.3. Evaluation Metrics

Over its editions, JOKER Task 2 has employed several evaluation approaches: the reference-based automated metrics BLEU [10] and BERTScore [13], a wordplay-specific location-based metric, and manual human evaluation by expert annotators. The organizers have repeatedly noted that reference-based metrics are poorly suited to pun translation: BLEU rewards literal, surface-level overlap and systematically penalizes the creative lexical choices required to preserve wordplay, while BERTScore captures semantic similarity but not whether a pun mechanism has been preserved [7].

In this work we report results *solely* on the official location-based metric. This metric evaluates whether the pun-bearing word of the source is realised as an ambiguous (multi-meaning) word at an annotated position in the translation; it is computed by the organizers from manual pun-location annotations that are not released to participants, and it is the only score returned to us for the 2026 test set. We adopt it as our single evaluation criterion for two reasons. First, the 2026 test set carries no French references, so BLEU and BERTScore cannot be computed by participants on the official data. Second, the location-based metric was redefined between the 2025 and 2026 editions, which makes cross-year comparison with previously published scores unreliable; we therefore restrict all comparison to a single, within-2026 axis. We also recall that, in the 2025 edition, the location-based metric correlated strongly with manual human judgments (Pearson $r = 0.84$), markedly more so than BLEU or BERTScore, which supports its use as the primary indicator of wordplay-preservation quality [7]. The location-based scores reported in this paper are the final official results of the 2026 test phase; the organizers' manual human evaluation is still ongoing at the time of writing. Whether the retrieval-augmented configurations' emphasis on fluency and humorousness pays off will only become visible in the organizers' manual human evaluation, which we await with particular interest given the strong divergence between human and automated rankings observed in the 2025 edition.

## 4. Methodology

### 4.1. System Overview

All our systems are assembled from the same small set of building blocks arranged into a *retrieve–generate–judge* pipeline. Given an English pun, a system may optionally retrieve similar examples, optionally classify the pun type, generate one or more French candidate translations through a structured decomposition, and finally use a judge model to produce the output. We define each block in Section 4.2 and then describe the five concrete pipelines obtained by combining them (Section 4.3); the combinations are summarized in Table 2.

**Table 2**
The five system configurations evaluated in this work (a $2 \times 2$ ablation over the retrieval-augmented pipeline, plus a non-retrieval direct-CVO reference).

| Configuration | RAG | Pun-type classifier | CVO |
|---|---|---|---|
| Direct CVO (reference) | — | — | ✓ |
| RAG baseline | ✓ | — | — |
| RAG + CVO | ✓ | — | ✓ |
| RAG + classifier | ✓ | ✓ | — |
| RAG + classifier + CVO | ✓ | ✓ | ✓ |

## 4.2. Building Blocks

Retrieval is performed against an index that we build offline. Each training pun is encoded with the `all-MiniLM-L6-v2` Sentence-Transformer and stored alongside its French reference translations. At inference time the input pun is encoded with the same model, the three training puns with the smallest cosine distance are selected, and each is inserted into the generator prompt together with one of its French references. The retrieved pairs act as few-shot demonstrations, showing the generator how humans have previously adapted similar wordplay rather than asking it to translate from scratch.

Our baseline reasoning scheme, which we refer to as the *simple decomposition*, asks the generator to fill a fixed JSON structure before it commits to any sentence. It first explains the pun and names the ambiguous word, then states the two French meanings that the pun plays on, and then identifies an intersection—a single French word, idiom, or homophone able to carry both meanings at once. Only after this does it produce exactly five French translations, ranging from a literal rendering, where the pun happens to transfer directly, to a freely adapted one. The purpose of the structure is to make the model locate a French pivot for the wordplay before writing the joke.

The *CVO decomposition* keeps the same five-variant output but replaces the generic two-concept analysis with the linguistically grounded scheme of Ma et al. [8]. Here the model decomposes the source into its constants, namely the core punning word (SM1), an anchor that primes one reading and a bridge that activates the other (together SM2), and the pragmatic humour effect (SPM), and then maps them to the French variables TM1, TM2, and TPM. It is explicitly allowed to switch the pun mechanism between homophonic and homographic when no direct equivalent exists in French. Relative to the simple decomposition, CVO gives the model an explicit account of why the joke is funny and of which element must be preserved across the language boundary.

The pun-type classifier is a separate, lightweight model call placed before generation. Following Delabastita's typology [2], it labels the pun as lexical, where a single word carries two meanings, phonetic, where two words sound alike, or idiomatic, where a fixed expression is altered. The label then selects a type-specific generator prompt: the lexical prompt searches for a French word spanning both senses, the phonetic prompt enumerates the English soundalikes and looks for French homophones or paronyms such as *mer*/*mère*, and the idiomatic prompt finds a French equivalent idiom and alters one of its words. The motivation is that the three pun types call for different translation moves, which a single generic prompt handles unevenly.

Finally, the judge model is used in one of two modes. In its binary mode it receives a single translation and returns only a yes/no verdict on whether the French sentence contains a valid, naturally embedded pun, with parenthetical explanations disallowed. In its selection mode it receives the five generated candidates, scores each for pun preservation, fluency, and humour, and returns the index of the best one, which becomes the system's output.

## 4.3. The Five Pipelines

The direct-CVO reference skips retrieval entirely. The pun is sent straight to the generator under the CVO prompt with a single worked example, the generator returns its analysis and one French translation enclosed in `<FR>` tags, and the judge performs only a binary pun-presence check. Because it produces a

single committed translation and never selects among alternatives, this configuration isolates the effect of CVO on its own.

The RAG baseline introduces the full retrieve–generate–judge loop. It retrieves the three nearest training pairs, runs the simple decomposition to obtain five candidate translations, and lets the judge select the best one in selection mode. The RAG+CVO pipeline is identical except that the simple decomposition is replaced by the CVO decomposition, which lets us measure the contribution of CVO once retrieval is already in place.

The two classifier pipelines add pun-type routing on top of retrieval. In RAG+classifier the pun is first classified, the three nearest pairs are retrieved, and the type-specialized prompt for the predicted type generates the five candidates that the judge then ranks. The full system, RAG+classifier+CVO, behaves the same way but augments each type-specialized prompt with the CVO decomposition, so that pun-type routing and CVO reasoning operate together.

### 4.4. Implementation Details

All models are served locally through Ollama on the Perun HPC cluster of Technical university of Kosice, scheduled via SLURM on GPU nodes; the generator and judge are kept resident to avoid reload overhead. To make long batch runs robust, each model call uses retries with exponential back-off, generator outputs are parsed as JSON with a repair fallback for malformed responses.

## 5. Experiments and Results

### 5.1. Experimental Setup

We evaluate all five configurations on the official JOKER 2026 Task 2 test set of 4,061 English puns, using the location-based metric computed and returned by the organizers (Section 3.3). As the test set carries no French references, we report no reference-based scores; the location metric is our single criterion (higher is better). Every configuration is run once over the full test set with the same generator and judge models (Appendix B). The reported scores are the final official results of the closed 2026 test phase.

### 5.2. Results

Table 3 reports the official location-based scores for our five runs. Our best configuration—direct CVO with no retrieval and no pun-type routing—reaches 33.16 and ranks fifth in the final official standings. Adding retrieval drops the score to 29.27; adding the classifier or CVO individually on top of retrieval changes little (28.83 and 28.73, within run-to-run noise of one another and only about half a point below the baseline); and combining the classifier with CVO yields the lowest score (25.38). The metric thus ranks the configurations broadly in inverse order of pipeline complexity.

**Table 3**
Official JOKER 2026 Task 2 results on the location-based metric (higher is better). Configurations follow Table 2.

| Configuration (run) | RAG | Class. | CVO | Location |
|---|---|---|---|---|
| Direct CVO (`CVO_Satira`) | — | — | ✓ | **33.16** |
| RAG baseline (`baseline`) | ✓ | — | — | 29.27 |
| RAG + classifier (`with_classifier`) | ✓ | ✓ | — | 28.83 |
| RAG + CVO (`baseline_cvo`) | ✓ | — | ✓ | 28.73 |
| RAG + classifier + CVO (`classifier_cvo`) | ✓ | ✓ | ✓ | 25.38 |

### 5.3. Analysis

With all four retrieval cells available, the $2 \times 2$ ablation shows a clear pattern. Relative to the RAG baseline (29.27), adding the classifier alone (28.83) or CVO alone (28.73) has a negligible effect, but

adding both together (25.38) costs nearly four points—far more than the sum of their individual effects. The two mechanisms therefore interact negatively rather than combining constructively. More broadly, the single largest factor is retrieval itself: the non-retrieval direct-CVO system outperforms every retrieval-augmented variant.

We attribute this to a misalignment between the pipeline and the metric. The direct-CVO system commits to a single translation whose punning word is explicitly tracked through the CVO decomposition, which tends to keep it at a detectable location. The retrieval-augmented configurations instead generate several candidates and select among them with a judge whose rubric—pun presence, fluency, and humour—is not aligned with the location annotation, trading positional pun fidelity for other qualities. The classifier adds a further point of failure, since a misrouted pun receives an unsuitable strategy. Notably, because the training puns recur in the test set, the retrieval runs had access to a known reference for part of the test data, yet still scored below the non-retrieval system, which makes the strength of direct CVO more striking. This is consistent with the broader JOKER finding that no single automated signal fully captures wordplay quality; the location metric rewards positional pun preservation, which our simplest, most focused system optimises best. Final ranking awaits the closed test phase and the organizers' human evaluation.

## 6. Conclusion and Future Work

We described our participation in CLEF 2026 JOKER Task 2 on English-to-French wordplay translation. We embedded the Constant-Variable Optimization (CVO) prompting strategy in a retrieval-augmented, multi-candidate generate-and-judge pipeline, and extended it with a Delabastita-based pun-type classifier. By toggling retrieval, classification, and CVO we evaluated five configurations on the official location-based metric. Our clearest finding is that the simplest configuration—direct CVO prompting without retrieval or classification—scored best (33.16, fifth in the final official standings), while added pipeline complexity did not help: retrieval lowered the score, and the classifier and CVO interacted negatively when combined. This suggests that, at least under this metric, a focused single-translation strategy that explicitly tracks the punning word outperforms candidate generation and selection.

Several directions could be explored to build on these findings. A multi-role framework inspired by HOMER [11] could decompose translation into specialised extractor, imaginator, and generator roles supported by humour-database retrieval. The generate-and-judge loop could be made iterative, re-prompting only the candidates that fail the judge, with a selection criterion better aligned with the location-based objective—addressing the pipeline–metric misalignment identified in our analysis. Finally, the approach could be extended to the Spanish target language introduced in this edition of the task, testing whether the advantage of direct CVO holds across typologically different target languages.

## Acknowledgments

This research was funded by the Slovak Research and Development Agency under the contract No. APVV 22-0414. We thank the organizers of the CLEF 2026 JOKER track for preparing the task, the data, and the evaluation.

## Declaration on Generative AI

During the preparation of this work, the author(s) used a generative AI assistant (Anthropic Claude) in order to: assist with grammar and spelling checking and stylistic refinement of the text. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References


[1] D. Crystal, *The Cambridge Encyclopedia of the English Language*, 2nd ed. Cambridge University Press, 2006.

[2] D. Delabastita, "There's a double tongue: An investigation into the translation of puns," *Target*, vol. 5, no. 2, pp. 221–242, 1993.

[3] D. Delabastita, "Wordplay as a translation problem: A linguistic perspective," in *Übersetzung*, pp. 600–606, Walter de Gruyter, 2004.

[4] F. Dhanani, M. Ra, and M. A. Tahir, "Humour Translation with Transformers," 2023.

[5] L. Ermakova et al., "CLEF Workshop JOKER: Automatic Wordplay and Humour Translation," in *ECIR 2022*, LNCS, vol. 13186, pp. 355–363, Springer, 2022.

[6] L. Ermakova et al., "Overview of JOKER–CLEF 2023 Track on Automatic Wordplay Analysis," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, LNCS, vol. 14163, pp. 397–415, Springer, 2023.

[7] L. Ermakova, R. Campos, A.-G. Bosser, and T. Miller, "Overview of the CLEF 2025 JOKER Task 2: Wordplay Translation from English into French," in *Working Notes of CLEF 2025*, CEUR-WS.org, 2025.

[8] Y. R. Ma, S. Huang, Y. Xu, Z. Zhou, and Y. Wei, "Pun2Pun: Benchmarking LLMs on Textual-Visual Chinese-English Pun Translation via Pragmatics Model and Linguistic Reasoning," in *Proc. ACL 2025 Student Research Workshop*, pp. 331–354, 2025.

[9] C. T. Marina Ilari, "Translating humor is a serious business," ATA Chronicle, 2021.

[10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, pp. 311–318, 2002.

[11] W. Shang, Y. Sun, J. Ma, and X. Huang, "On the Wings of Imagination: Conflicting Script-Based Multi-Role Framework for Humor Caption Generation," in *Proc. ICLR 2026*, 2026.

[12] O. Weller and K. Seppi, "The rJokes Dataset: a Large Scale Humor Collection," in *Proc. LREC 2020*, pp. 6136–6141, 2020.

[13] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," in *Proc. ICLR 2020*, 2020.

[14] H. Zhao, "A quantitative model for pragmatic translation of puns," *Foreign Languages Research*, no. 5, pp. 72–76, 2012.

[15] H. Zhao and Y. An, "The meaning optimization of variables in translation of puns," *Foreign Language Research*, no. 6, pp. 92–98, 2020.

[16] I. Kuzmin, "CLEF 2025 JOKER track: No pun left behind," in: [19], 2025.

[17] A. Kreefft-Libiu, F. Helms, C. Selçuk, J. Bakker, and J. Kamps, "University of Amsterdam at the CLEF 2025 JOKER Track," in: [19], 2025.

[18] R. Taylor, B. Herbert, and M. Sana, "Pun Intended: Multi-Agent Translation of Wordplay with Contrastive Learning and Phonetic-Semantic Embeddings for CLEF JOKER 2025 Task 2," in: [19], 2025.

[19] G. Faggioli, N. Ferro, P. Rosso, and D. Spina (Eds.), *Working Notes of CLEF 2025: Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 4038, CEUR-WS.org, 2025.

[20] L. Ermakova et al., "Overview of the CLEF 2026 JOKER Track: Humor Detection, Search, and Translation," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026)*, LNCS, Springer, 2026.

[21] L. Ermakova et al., "Overview of the CLEF 2026 JOKER Task 2: Pun Translation from English to French," in: [22], 2026.

[22] E. Sánchez Salido, A. Barrón Cedeño, A. García Seco de Herrera, S. MacAvaney, and J. M. Struß (Eds.), *Working Notes of CLEF 2026: Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

# A. Prompt Design

This appendix lists the prompts used by our systems. All prompts are passed as system messages; the placeholder `{examples_block}` is filled at inference time with the retrieved few-shot pairs, and JSON schemas are shown with single braces for readability.

## A.1. Shared Pun Definition

The following definition is prepended to several prompts to establish a common vocabulary.

```
A homophonic pun relies on words that sound the same or similar but have different meanings.
    ↪
A homographic pun relies on words with the same spelling/form that have multiple meanings.
```

## A.2. Direct-CVO Generator Prompt (Pipeline 1)

```
[Shared Pun Definition]

You are an expert translator specializing in translating English puns into French.
Please apply the "Constant-Variable" (CVO) theory to optimize pun translation.

CVO Theory:
Constants (Source Meanings, SM) from the English original:
- SM1: The core word/phrase containing the pun (dual meaning).
- SM2: The contextual trigger consisting of an Anchor (basis) and Bridge (supporting word).
- SPM (Source Pragmatic Meaning): The humor effect formed by SM1 and Bridge.

Variables (Target Meanings, TM) in the French translation:
- TM1: The core word in French that reproduces the dual meaning.
- TM2: The contextual support in French.
- TPM: The reproduced humor effect in French.

Your task: Translate the following English pun into French. You MUST preserve the
pun effect. You are allowed to convert a homophonic pun into a homographic pun, or
vice versa, to make it work naturally in French.

--- EXAMPLE ---
English Pun: I used to be a banker but I lost interest
Analysis:
1. Constants (SM): SM1="interest" (financial rate / personal enthusiasm).
   Anchor="banker", Bridge="lost". SPM = humor from losing both.
2. Variables (TM): TM1="interet" (same double meaning in French).
   Anchor="banquier", Bridge="perdu". TPM = same humor effect in French.
3. Reconstruct: map the English structure to French smoothly.
Translation:
<FR>J'ai ete banquier mais j'en ai perdu tout l'interet.</FR>
--- END OF EXAMPLE ---

Let's think step by step. You MUST use the English indicator "Analysis:" for your
reasoning. After your analysis, you MUST wrap your final French translation inside
<FR> and </FR> xml tags!
```

## A.3. Binary Judge Prompt (Pipeline 1)

```
You are a strict translation expert and native French speaker evaluating
English-to-French pun translations.
[Shared Pun Definition]

Your task is to determine if the French translation contains a valid pun.
1. Check Fluency: Is the French natural? If not, answer "No".
2. Check Pun: Does the French sentence contain a word with similar/same pronunciation
   but different meanings (homophonic) OR a word with similar form but different
   meanings (homographic)?
3. Remember: We DO NOT allow explaining the pun in parentheses. The pun must be natural.

Evaluate the source English pun and the provided French translation.
End your response with exactly:
Final Answer: Yes
OR
Final Answer: No
```

## A.4. Simple Decomposition Generator Prompt (Pipelines 2 and 4)

```
You are an expert bilingual comedian and translator specializing in wordplay and puns.
Your task is to translate an English punning joke into French while preserving BOTH
the meaning and the wordplay (pun).

Here are some examples of successful translations:
{examples_block}

INSTRUCTIONS FOR DECOMPOSITION:
1. "analysis": Briefly explain the pun in English. Identify the ambiguous word.
2. "concept_1": Identify the first meaning/concept in French.
3. "concept_2": Identify the second meaning/concept in French.
4. "intersection": Find a French word, idiom, or homophone that can unite these two
   concepts, just like the original English pun.
5. "variants": Propose exactly 5 DIFFERENT French translations ranging from literal
   (if the pun works) to adapted (changing the joke slightly to fit a French pun).

Output STRICTLY as a valid JSON object:
{
  "analysis": "...",
  "concept_1": "...",
  "concept_2": "...",
  "intersection": "...",
  "variants": ["Variant 1", "Variant 2", "Variant 3", "Variant 4", "Variant 5"]
}
Do not include any other text outside the JSON object.
```

## A.5. CVO Decomposition Generator Prompt (Pipelines 3 and 5)

This prompt prepends the Shared Pun Definition and the CVO Theory block (as in the direct-CVO prompt above) and then replaces the two-concept analysis with a CVO analysis.

```
[Shared Pun Definition]
You are an expert bilingual comedian and translator specializing in wordplay and puns.
Your task is to translate an English punning joke into French while preserving BOTH
the meaning and the wordplay (pun).
[CVO Theory]

Here are some examples of successful translations:
{examples_block}

INSTRUCTIONS (apply CVO step by step):
1. "analysis": Identify the pun word and its dual meaning. Then apply CVO:
   SM1 (core pun word), Anchor (context basis), Bridge (supporting word), SPM (humor effect
    ↪ ).
2. "TM1": Find a French word that reproduces the dual meaning (the Variable).
3. "TM2": Identify the French contextual support (Anchor + Bridge in French).
4. "TPM": Describe the reproduced humor effect in French.
5. "variants": Propose exactly 5 DIFFERENT French translations, from literal
   (if the pun transfers directly) to adapted (changing context to fit a French pun).

Output STRICTLY as a valid JSON object:
{
  "analysis": "SM1=..., Anchor=..., Bridge=..., SPM=...",
  "TM1": "...",
  "TM2": "...",
  "TPM": "...",
  "variants": ["...", "...", "...", "...", "..."]
}
Do not include any other text outside the JSON object.
```

## A.6. Pun-Type Classifier Prompt (Pipelines 4 and 5)

```
You are an expert linguist classifying the type of wordplay (pun) in English jokes.
Based on Delabastita's typology, classify the pun into one of these 3 categories:

1. "lexical": Homonymy or polysemy (a single word has two different meanings).
   Example: "banker lost *interest*".
2. "phonetic": Homophony or paronymy (two different words that sound the same or very
   similar). Example: "Why is 6 afraid of 7? Because 7 *8* (ate) 9".
3. "idiomatic": Syntactic or phraseological (a well-known idiom or phrase is broken or
   altered). Example: "A baker's life is a *piece of cake*".

Output STRICTLY as a valid JSON object:
{
  "reason": "Briefly explain why",
  "pun_type": "lexical" | "phonetic" | "idiomatic"
}
```

©

## A.7. Type-Specialized Decomposition Instructions

Each predicted type selects a generator prompt that shares the structure above but differs in its decomposition instructions. The lexical prompt uses the simple decomposition of A.4. The phonetic and idiomatic instructions are as follows (in the classifier+CVO pipeline these blocks are additionally prefixed with the CVO Theory).

```
PHONETIC:
1. "analysis": Briefly explain the phonetic pun. Which words sound alike?
2. "english_soundalikes": List the English words that sound alike.
3. "french_homophones": Find French homophones or paronyms that recreate a similar
   phonetic joke (e.g., mer/mere, sans/sang, verre/vert, sept/cet).
4. "variants": 5 French translations adapting the joke to the French homophones found.

IDIOMATIC:
1. "analysis": Explain the pun. What is the original English idiom and how is it altered?
2. "french_equivalent": A similar French idiom or proverb.
3. "french_alteration": How to alter the French idiom (change a word) for a similar effect.
4. "variants": 5 French translations using the altered French idiom.
```

## A.8. Judge Selection Prompt (Pipelines 2–5)

```
You are an expert bilingual judge evaluating translations of English puns into French.
Your goal is to select the BEST translation among 5 variants.
A successful translation MUST preserve the wordplay (pun) in French, sound natural,
and be funny.

INSTRUCTIONS:
Evaluate each of the 5 variants from 1 to 10 on three criteria:
- Pun preservation (does it contain a double meaning or wordplay?)
- Fluency (is it natural French?)
- Humorousness (is it funny?)
Then pick the index (1 to 5) of the best variant.

Output STRICTLY as a valid JSON object:
{
  "evaluations": [
    {"variant_index": 1, "pun_score": 8, "fluency_score": 9, "humor_score": 8},
    {"variant_index": 2, "pun_score": ..., "fluency_score": ..., "humor_score": ...},
    {"variant_index": 3, "pun_score": ..., "fluency_score": ..., "humor_score": ...},
    {"variant_index": 4, "pun_score": ..., "fluency_score": ..., "humor_score": ...},
    {"variant_index": 5, "pun_score": ..., "fluency_score": ..., "humor_score": ...}
  ],
  "best_variant_index": 1
}
Do not include any other text outside the JSON object.
```

## B. Model Configuration

Table 4 lists the models and decoding parameters used in all configurations. The same generator and judge are used across every pipeline; the classifier reuses the judge model. All models are served locally through Ollama.

**Table 4**
Models and decoding parameters used in the pipeline.

| Setting | Generator | Judge / Classifier |
|---|---|---|
| Model | `mistral-large:123b` | `llama3.1:70b` |
| Parameters | 123B | 70B |
| Temperature | 0.8 | 0.1 |
| Top-$p$ | 0.95 | 0.9 |
| Top-$k$ | 40 | 10 |
| Context length | 8192 tokens | |
| Max new tokens | 2048 | |
| Repeat penalty | 1.2 | |

The retrieval encoder is `all-MiniLM-L6-v2`, with the top-3 nearest training puns used as few-shot examples (cosine distance). The generator is run with creative decoding to encourage diverse pun reconstructions, while the judge and classifier use near-deterministic decoding for stable verdicts.