

# DUTH at CLEF JOKER 2026: Conservative Hybridisation for Humor-Aware Retrieval, Pun Translation, Onomastic Wordplay Translation, and Multilingual Humour Generation

Georgios Arampatzis<sup>1,\*</sup>, Avi Arampatzis<sup>1</sup>

*Democritus University of Thrace, Department of Electrical and Computer Engineering, Xanthi, Greece*

## Abstract

This paper presents the participation of Team DUTH in the CLEF JOKER 2026 track, covering four tasks on computational humour and wordplay: humor-aware information retrieval, English–French pun translation, English–French onomastic wordplay translation, and guided multilingual humour generation. Although the tasks differ in input format and evaluation setting, they share a common challenge: systems must process creative language while preserving lexical, semantic, cultural, and humorous constraints. Across all tasks, our methodology follows a conservative hybridisation strategy. Rather than relying on unconstrained neural generation, we combine strong lexical or translation baselines, supervised reranking, rank fusion, official and mined lexical resources, retrieval-based candidates, large language model candidate selection, and rule-based filtering.

For Task 1, we develop a hybrid retrieval and reranking framework for English and Hinglish that combines TF–IDF, BM25, supervised learning-to-rank, tree-based ensembles, neural reranking, rank fusion, and artifact-aware analysis. Our best submitted runs achieved 0.7582 MAP for English and 0.6648 MAP for Hinglish. For Task 2, we use Google Translate as a strong base system and improve it through block-level LLM-guided candidate selection, reaching an official score of 32.0783. For Task 3, we formulate onomastic wordplay translation as a franchise-aware name-translation problem and combine training-data reuse, official bilingual lexicons, description mining, external lookup, and conservative candidate filtering, obtaining 0.4537 exact match. For Task 4, we treat humour generation as controlled creative generation and combine deterministic templates, training-data pattern mining, retrieval-based candidates, LLM-generated alternatives, and language-specific patching, achieving best submitted scores of 10.68 for English, 8.34 for Spanish, and 10.83 for French.

Overall, our results show that controlled hybrid systems are effective for computational humour tasks where fluency alone is insufficient and where exact lexical, cultural, or reference-oriented constraints strongly affect evaluation.

## Keywords

Computational Humour, Wordplay Processing, Humor-Aware Retrieval, Pun Translation, Onomastic Wordplay Translation, Multilingual Humour Generation, Learning to Rank, Rank Fusion, Large Language Models, Candidate Selection, Conservative Hybridisation, CLEF JOKER

## 1. Introduction

Humour and wordplay remain difficult phenomena for computational language processing, because they depend on more than literal semantic interpretation. Humorous language often exploits ambiguity, incongruity, phonological similarity, morphology, cultural knowledge, and pragmatic context in order to produce a compact communicative effect [1, 2]. These properties make wordplay challenging for several NLP tasks. In retrieval, a system must identify texts that are not only topically relevant to a query but also humorous or wordplay-bearing. In translation, it must preserve meaning while also transferring or recreating the humorous mechanism in the target language. In generation, it must produce fluent and natural text while satisfying lexical, semantic, and cultural constraints.

---

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

\*Corresponding author.

✉ geoaramp@ee.duth.gr (G. Arampatzis); avi@ee.duth.gr (A. Arampatzis)

🆔 0009-0003-3840-4537 (G. Arampatzis); 0000-0003-2415-4592 (A. Arampatzis)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The CLEF JOKER track provides a shared benchmark for computational humour, wordplay, and creative language processing across multilingual settings [3]. The 2026 edition includes tasks that cover different aspects of humour-aware processing, including retrieval, pun translation, onomastic wordplay translation, and multilingual humour generation. Although these tasks differ in input format, output format, and evaluation protocol, they share a common underlying difficulty: systems must handle playful language under strict task-specific constraints. A fluent output is not sufficient if it fails to preserve the intended wordplay, ignores the required lexical item, retrieves non-humorous content, or produces a translation that differs from the expected reference form.

This paper presents the participation of Team DUTH in the CLEF JOKER 2026 track. We describe our systems for four tasks: Task 1 on humor-aware information retrieval in English and Hinglish, Task 2 on English–French pun translation, Task 3 on English–French onomastic wordplay translation, and Task 4 on guided multilingual humour generation. Across these tasks, our central methodological principle is conservative hybridisation. Rather than relying on a single model or on unconstrained large language model generation, we combine strong lexical or translation baselines with controlled selection, rule-based filtering, retrieval, rank fusion, lexical resources, and task-specific validation.

For Task 1, we address humor-aware retrieval as a ranking problem in which documents must satisfy both topical relevance and humorous intent. Our system combines word-level and character-level TF-IDF, BM25-style lexical retrieval, supervised learning-to-rank, tree-based ensembles, neural reranking, and reciprocal rank fusion [4, 5]. This design is motivated by the fact that wordplay often depends on surface form, partial string overlap, spelling variation, and character-level similarity, while humour-aware relevance also requires broader topical matching. For Hinglish, code-mixing and non-standard spelling further increase the importance of hybrid lexical and character-level evidence.

For Task 2, we address English–French pun translation using a multi-candidate translation and block-level fusion strategy. Pun translation is difficult because literal machine translation may preserve the surface meaning while losing the humorous ambiguity or wordplay mechanism [6]. Our approach uses Google Translate as a strong base system and generates alternative candidates using several machine translation and large language model components. Instead of applying direct LLM translation or full-system replacement, we use instruction-tuned models as candidate selectors over predefined translation alternatives. This allows the system to exploit the semantic and pragmatic knowledge of LLMs while reducing the risk of hallucination, excessive rewriting, or reduced reference agreement.

For Task 3, we formulate onomastic wordplay translation as a reference-oriented and franchise-aware name translation problem. Humour-bearing proper names often occur in fictional universes such as comics, animation, games, and children’s literature. Some names should be copied, some have official French equivalents, and others require creative adaptation. Since exact-match evaluation rewards agreement with hidden reference translations, broad unconstrained generation is risky. Our system therefore combines high-precision training-data reuse, safeguards against unsafe identifier-based matches, franchise-specific bilingual lexicons, description-based extraction, external knowledge lookup, and conservative candidate filtering. The goal is to recover official or reference-compatible French forms whenever there is sufficient evidence, while avoiding noisy substitutions.

For Task 4, we treat multilingual humour generation as controlled creative generation. The task requires systems to generate short humorous texts from a target pun word and sense descriptions in English, Spanish, and French. Large language models can produce fluent candidates, but they may ignore the required pun word, over-explain the joke, or drift away from the intended senses. Conversely, rule-based generation preserves constraints more reliably but may produce weaker humour. We therefore combine deterministic templates, training-data pattern mining, retrieval-based candidates, LLM-generated alternatives, and language-specific conservative patching. Candidate changes are applied only when they are likely to improve or preserve the current best run, which is especially important in a setting where automatic humour evaluation is sensitive to lexical and semantic alignment.

The main contribution of this paper is a unified account of our participation across the full JOKER track. First, we show that humor-aware retrieval benefits from combining lexical, character-level, supervised, neural, and rank-fusion signals. Second, we show that LLMs can be useful for pun translation when used as controlled candidate selectors rather than as unconstrained translators. Third, we show that

onomastic wordplay translation benefits from franchise-aware lexicon mining and high-confidence official-name recovery. Fourth, we show that multilingual humour generation can be approached through controlled candidate construction and conservative patching rather than broad open-ended generation.

Overall, our results support the view that computational humour tasks require systems that are both creative and constrained. Across retrieval, translation, name translation, and generation, the strongest systems were neither purely neural nor purely rule-based. Instead, they combined complementary sources of evidence and applied task-specific controls before accepting final outputs. This suggests that conservative hybridisation is a practical strategy for shared tasks in which humour, wordplay, cultural knowledge, and reference-oriented evaluation interact.

The remainder of this paper is organized as follows. Section 2 provides an overview of our participation across the CLEF JOKER 2026 track and summarizes the common methodological principles behind our systems. Section 3 discusses related work on computational humour, wordplay processing, humour-aware retrieval, pun translation, onomastic wordplay translation, and humour generation. Section 4 describes our humour-aware information retrieval system for English and Hinglish. Section 5 presents our block-level LLM-guided fusion approach for English–French pun translation. Section 6 describes our franchise-aware approach to English–French onomastic wordplay translation. Section 7 presents our controlled multilingual humour-generation pipeline for English, Spanish, and French. Section 8 discusses the main findings across tasks, including the role of conservative hybridisation, lexical control, candidate selection, and benchmark-specific constraints. Finally, Section 9 concludes the paper and outlines directions for future work in multilingual humour and wordplay processing.

## 2. Overview of the DUTH Participation

Team DUTH participated in four tasks of the CLEF JOKER 2026 track: humor-aware retrieval, English–French pun translation, English–French onomastic wordplay translation, and guided multilingual humour generation. Although these tasks differ in input format, output type, language setting, and evaluation protocol, they share a common requirement: systems must process creative language while respecting lexical, semantic, cultural, and task-specific constraints. Task 1 requires the retrieval of short humorous texts that are both topically relevant and wordplay-bearing. Task 2 requires English–French pun translation while preserving meaning, fluency, and, when possible, the wordplay mechanism. Task 3 focuses on humour-bearing proper names, often requiring the recovery of official or conventional French forms. Task 4 requires multilingual humour generation from a target pun word and sense descriptions.

Across the track, our submissions follow a conservative hybridisation principle. Rather than using large language models as unconstrained generators or translators, we use them selectively as candidate generators, auxiliary selectors, post-editing assistants, or reranking signals. Final outputs are controlled through lexical matching, ranking, rule-based filtering, official and mined lexicons, validation scripts, and ablation-driven selection. Table 1 summarizes the four tasks, the main languages, the core methodological components, and the best submitted results of our systems.

The four systems can be viewed as two broader methodological families. The first family concerns retrieval and selection. Task 1 combines lexical, character-level, supervised, neural, and rank-fusion signals to rank humour-bearing documents in English and Hinglish. Task 2 treats machine-translation and LLM-based outputs as candidate translations and applies block-level selection only where alternatives improve a strong baseline. In both cases, the system avoids relying on a single model output and instead combines complementary evidence conservatively.

The second family concerns controlled lexical and creative generation. Task 3 aims to recover reference-compatible French forms for humour-bearing proper names using training-data reuse, franchise-aware lexicons, description mining, external lookup, and conservative filtering. Task 4 generates short humorous texts from a target pun word and sense descriptions using deterministic templates, training-data patterns, retrieval-based candidates, LLM-generated alternatives, and language-

Table 1: Overview of the DUTH participation in CLEF JOKER 2026.

| Task   | Setting                                                  | Main approach                                                                                                             | Metric                   | Best result                           |
|--------|----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|--------------------------|---------------------------------------|
| Task 1 | Humor-aware retrieval in English and Hinglish            | re- TF-IDF, BM25, supervised LTR, tree ensembles, neural reranking, rank fusion, and artifact-aware analysis              | MAP                      | 0.7582 (EN)<br>0.6648 (HI)            |
| Task 2 | English–French translation                               | pun Google Translate baseline, multi-system candidate generation, block probing, and LLM-guided candidate selection       | Official automatic score | 32.0783                               |
| Task 3 | English–French onomastic wordplay translation            | ono- Training-data reuse, franchise-aware lexicons, description mining, external lookup, and conservative filtering       | Exact Match              | 0.4537                                |
| Task 4 | Guided humour generation in English, Spanish, and French | Deterministic templates, training-pattern mining, retrieval-based candidates, LLM alternatives, and conservative patching | Official automatic score | 10.68 (EN)<br>8.34 (ES)<br>10.83 (FR) |

specific patching. In both tasks, broad open-ended generation is risky because fluent outputs may violate lexical constraints, miss official or reference-compatible forms, or drift away from the intended sense structure.

Overall, the common lesson across the four tasks is that computational humour systems must balance creativity and control. Lexical and rule-based components provide stability and constraint preservation, while neural and LLM-based components provide candidate diversity, semantic judgement, and selective improvements. The following sections describe each task-specific system in detail. Section 4 presents the humor-aware retrieval system for English and Hinglish. Section 5 describes the block-level LLM-guided fusion approach for English–French pun translation. Section 6 presents the franchise-aware system for onomastic wordplay translation. Section 7 describes the controlled multilingual humour-generation pipeline.

### 3. Related Work

Computational humour and wordplay have long been challenging topics for natural language processing, because they require models to handle ambiguity, incongruity, linguistic form, cultural knowledge, and pragmatic interpretation rather than only literal meaning [1, 2]. Puns are particularly difficult because their humorous effect often depends on the simultaneous activation of multiple meanings, sound similarities, or form-based associations. These properties affect retrieval, translation, and generation: systems must identify, preserve, or recreate playful effects while remaining compatible with the intended meaning and the constraints of each task.

Shared tasks have made humour and wordplay processing more measurable and comparable. SemEval-2017 Task 7 introduced benchmark subtasks for pun detection, location, and interpretation, highlighting the computational difficulty of lexical ambiguity [7]. The CLEF JOKER track extended this direction to multilingual humour and wordplay processing, covering humor-aware retrieval, pun translation, onomastic wordplay translation, and creative language generation [8, 9, 10]. These benchmarks show that humour-related NLP systems must account for both semantic content and linguistic form, especially when the target phenomenon depends on ambiguity, cultural references, or creative lexical manipulation.

Humour-aware information retrieval differs from standard ad hoc retrieval because topical relevance alone is not sufficient. Retrieved documents must also express humorous intent or contain wordplay. Classical lexical retrieval methods such as TF-IDF and BM25 remain strong baselines for short-text retrieval, especially when query terms or related forms occur directly in the document [4]. However, wordplay may also depend on surface similarity, spelling variation, affixation, punctuation, and partial string overlap. This motivates hybrid retrieval methods that combine word-level matching, character-level evidence, supervised learning-to-rank, and rank fusion. Reciprocal Rank Fusion is useful in such settings because it can combine heterogeneous rankings without requiring score calibration [5].

Wordplay translation is a related but distinct challenge. In pun translation, the target text must preserve not only propositional meaning, but also ambiguity, humorous effect, and, when possible, the wordplay mechanism. Translation studies have long noted that wordplay is difficult to translate literally, because the relevant ambiguity or sound pattern may not exist in the target language [6]. In shared-task settings, this difficulty is amplified by reference-based evaluation: a creative translation may be valid but still receive a low automatic score if it differs from the available reference. Automatic metrics such as BLEU provide reproducible comparison signals, but they cannot fully measure humour preservation, pragmatic effect, or creative adequacy [11].

Onomastic wordplay translation introduces an additional entity-level and cultural dimension. Humour-bearing proper names often occur in fictional universes such as literature, comics, animation, games, and popular media. Some names are copied unchanged, others have official target-language equivalents, and others require creative adaptation. Exact-match evaluation makes this task particularly sensitive to whether the system recovers the expected conventional or reference-compatible form. Multilingual knowledge resources such as Wikidata can provide useful cross-lingual labels and aliases for entity-level translation and disambiguation [12]. However, such resources can be incomplete, ambiguous, or contextually misleading, making high-precision candidate selection important.

Humour generation requires a different form of control. A system must produce fluent and concise humorous text while satisfying lexical and semantic constraints, such as a required pun word and sense descriptions. Large language models have improved open-ended text generation and can produce fluent candidates, but they may ignore constraints, over-explain the joke, generate generic templates, or drift away from the intended ambiguity [13, 14]. Rule-based methods, in contrast, can preserve constraints more reliably but may produce less natural or less surprising humour. Hybrid generation pipelines therefore combine templates, retrieval, neural candidates, and post-generation filtering in order to balance creativity and control.

The present work is also related to previous DUTH participation in shared tasks on humour, wordplay, retrieval, simplification, and multilingual NLP [15, 16, 17, 18, 19, 20, 21, 22]. Additional DUTH work in TREC explored ranking, sparse-neural fusion, retrieval-augmented reasoning, and candidate selection [23, 24, 25, 26, 27]. These lines of work provide general background for shared-task system design, but the task-specific methods used in CLEF JOKER 2026 are described separately in the following sections.

Overall, prior work suggests that humour and wordplay tasks benefit from systems that combine lexical precision, semantic flexibility, and task-specific constraints. This motivates the conservative hybridisation perspective adopted in this paper, while the concrete retrieval, translation, name-recovery, and generation pipelines are presented in Sections 4–7.

## **4. Task 1: Artifact-Aware Hybrid Retrieval and Rank Fusion for Humor-Aware Search**

### **4.1. Dataset**

We used the official datasets released for CLEF JOKER 2026 Task 1: Humor-Aware Information Retrieval [28]. The task consists of retrieving short humorous texts from a document collection given a query. A retrieved document is considered relevant when it is both related to the query and expresses humour, wordplay, or humorous intent. This makes the task different from standard ad hoc retrieval, because topical relevance alone is not sufficient.

The task includes two language settings: English and Hinglish. The English collection contains 96,603 documents and 972 test queries, while the Hinglish collection contains 32,652 documents and 485 test queries. For both languages, the organizers released a small set of training queries with relevance judgments, which we used for supervised reranking. The English training set contains 10 queries and 25 positive relevance judgments, whereas the Hinglish training set contains 10 queries and 60 positive relevance judgments.

Each document in the corpus consists of a document identifier and a short humorous text. Queries

are represented by a query identifier and a query string. Training relevance judgments follow the standard query–document format, where a positive label indicates that a document is relevant to the query under the humor-aware definition of the task. The test relevance judgments are hidden, and evaluation is performed through the official CodaBench leaderboard.

Table 2: Dataset statistics for CLEF JOKER 2026 Task 1.

| Language | Documents | Train queries | Test queries | Train positives |
|----------|-----------|---------------|--------------|-----------------|
| English  | 96,603    | 10            | 972          | 25              |
| Hinglish | 32,652    | 10            | 485          | 60              |

The limited number of training queries makes the task particularly challenging for supervised learning. For this reason, our approach relies on a combination of robust lexical retrieval features, supervised reranking, neural reranking, and rank fusion rather than on large-scale task-specific fine-tuning.

## 4.2. Models Used

Our Task 1 system combines lexical retrieval, supervised reranking, tree-based ensembles, neural reranking, rank fusion, and artifact-aware analysis. As lexical baselines, we use word-level TF–IDF with unigram and bigram features and a BM25-style retrieval component. TF–IDF captures direct query–document term overlap, while BM25 adds probabilistic term weighting with document-length normalization [29, 4]. These lexical signals are particularly useful because many humorous or wordplay-related documents preserve surface overlap with the query.

We also use character-level TF–IDF features based on character  $n$ -grams. This component targets surface-form phenomena that are central to wordplay, including affixation, partial string overlap, spelling variation, and creative morphology. Character-level evidence is especially important for Hinglish, where code-mixing, transliteration, and non-standard spelling make word-level matching less reliable [30, 31].

The main supervised component is a learning-to-rank reranker trained on the released relevance judgments. The initial LTR model is a logistic regression reranker using lexical scores, character-level scores, rank positions, exact-match indicators, substring overlap, compact string overlap, and surface-level humour indicators. Learning-to-rank is suitable for this setting because it allows heterogeneous retrieval signals to be combined into a single ranking model [32].

For the strongest standard systems, we train tree-based LTR models, including Random Forests, Extra Trees, Gradient Boosting, and Histogram-based Gradient Boosting. These models capture non-linear interactions between lexical, character-level, rank-based, and surface-form features. The English results show that tree-based systems form a strong standard baseline, while the best Hinglish result is obtained by an Extra Trees and Gradient Boosting ensemble.

For English, we further experiment with neural reranking and fusion using DeBERTa, MS MARCO MiniLM L6/L12 cross-encoders, MonoT5, and a train-qrels selector with expanded candidates [33, 34, 35]. These models provide additional semantic ranking signals and are compared with the lexical and tree-based configurations. Finally, for English only, we analyze artifact-aware block-ranking variants. These runs exploit an observed corpus-order regularity and are therefore treated separately from the transferable retrieval models.

For Hinglish, exact-match and block-offset rescue strategies were also tested, but they did not improve over the strongest supervised tree ensemble. The final Hinglish results therefore support feature-based supervised reranking rather than artifact-aware reordering for the code-mixed setting.

## 4.3. Methodology

Our methodology consists of five stages: candidate retrieval, feature extraction, supervised reranking, rank fusion, and artifact-aware analysis.

In the first stage, we retrieve candidate documents using several lexical views: word-level TF-IDF, character-level TF-IDF, a hybrid TF-IDF combination, and BM25. For each query, cosine similarity is computed between the query vector and all document vectors in the TF-IDF spaces. The hybrid score combines word-level and character-level similarities, while BM25 provides an additional lexical ranking signal based on probabilistic term weighting and document-length normalization [4].

In the second stage, we construct query-document feature vectors for the retrieved candidates. The features include word-level TF-IDF similarity, character-level TF-IDF similarity, hybrid similarity, BM25 score, reciprocal ranks from the different retrieval views, exact query occurrence, substring occurrence, compact string overlap, token overlap, query length, document length, and punctuation-based indicators. The punctuation indicators capture question marks, exclamation marks, quotation marks, and hyphens, since short humorous texts often use punctuation to signal punchlines, contrast, dialogue, or wordplay.

In the third stage, we train supervised rerankers using the released training relevance judgments. Positive examples correspond to query-document pairs marked as relevant by the organizers, while negative examples are sampled from the retrieved candidate pools. Since the number of positive relevance judgments is small, we use class-balanced training where applicable. The logistic regression reranker provides a linear learning-to-rank baseline, while tree-based rerankers capture non-linear interactions among lexical, character-level, rank-based, and surface-form features.

In the fourth stage, we combine ranked lists using rank-based fusion. For English, we combine tree-based rerankers with neural and selector-based ranking signals, including DeBERTa, MS MARCO MiniLM L6/L12, MonoT5, and a train-qrels selector with expanded candidates. This reduces reliance on a single model and allows complementary lexical, neural, and supervised signals to contribute to the final ranking. For Hinglish, we also test weighted Reciprocal Rank Fusion over the supervised LTR ranking, the hybrid lexical ranking, and the BM25 ranking. RRF is useful because it combines ranked lists without requiring score calibration [5]. Given a document  $d$  and a set of rankings  $R$ , the RRF score is computed as:

$$\text{RRF}(d) = \sum_{r \in R} \frac{w_r}{k + \text{rank}_r(d)} \quad (1)$$

where  $w_r$  is the weight assigned to ranking  $r$ ,  $k$  is a smoothing constant, and  $\text{rank}_r(d)$  is the rank of document  $d$  in ranking  $r$ . The best Hinglish RRF configuration assigns the largest weight to the supervised LTR ranking and smaller complementary weights to the hybrid lexical and BM25 rankings. However, the final strongest Hinglish result is obtained by a tree-based ensemble, showing that supervised feature interactions are more effective than rank fusion alone in the code-mixed setting.

Finally, the top 100 documents per query are exported in the official JSON format required by the CodaBench evaluator. Each prediction contains the run identifier, query identifier, document identifier, rank, and score. The final submission is packaged as a ZIP archive containing a single prediction file.

A separate part of the English analysis concerns artifact-aware ranking. After standard lexical, tree-based, neural, and selector-based systems reached a performance plateau around 0.607 MAP, we observed a strong corpus-order regularity in the English benchmark. We therefore tested block-aware ranking variants. These runs are reported separately from the transferable retrieval and reranking systems, because they reveal a benchmark-specific artifact rather than a generally reusable semantic retrieval strategy.

#### 4.4. Evaluation Metrics

Systems were evaluated through the official CodaBench leaderboard using standard information retrieval metrics. The primary metric was Mean Average Precision (MAP), which measures ranking quality across the full retrieved list by rewarding systems that retrieve relevant documents early and consistently [36]. We also report normalized Discounted Cumulative Gain at cutoffs 5, 10, and 100 (nDCG@5, nDCG@10, and nDCG@100), which evaluates ranking quality while giving higher importance to relevant documents appearing near the top of the ranking [37]. Mean Reciprocal Rank (MRR) was used to measure how early the first relevant document is retrieved, while Precision at rank 10 (P@10)

captures the proportion of relevant documents among the top ten retrieved results [38]. The official evaluator also reports the number of evaluated queries, the number of retrieved relevant documents, and the total number of relevant documents.

#### 4.5. Implementation and Environment

All Task 1 systems were implemented in Python using a modular retrieval and reranking pipeline. We used scikit-learn for TF-IDF vectorization, logistic regression, Random Forests, Extra Trees, Gradient Boosting, and Histogram-based Gradient Boosting. Sparse matrix operations were implemented with SciPy, while NumPy was used for numerical processing, score normalization, and rank manipulation. Batch processing and progress monitoring were handled with tqdm.

The lexical retrieval components were implemented with sparse vector representations in order to support efficient scoring over the full English and Hinglish document collections. Word-level TF-IDF used unigram and bigram features, whereas character-level TF-IDF used character  $n$ -grams to capture surface-form variation. Query and document vectors were normalized before cosine-similarity computation.

The BM25 component was implemented as a sparse transformation over term-count features. Document-frequency statistics were used to estimate inverse document frequency values, and term-frequency normalization followed the standard BM25 formulation [4]. This produced an additional lexical ranking signal for all queries and documents.

All supervised rerankers were trained using only the released training queries and relevance judgments. Positive instances corresponded to the official training qrels, while negative instances were sampled from retrieved candidate sets. Tree-based rerankers were trained with parallel execution where supported. For English, the final submitted ranking combined standard retrieval and reranking outputs with artifact-aware block ranking. For Hinglish, the final submitted ranking was produced by a tree-based ensemble over supervised query-document features.

Submission files were generated automatically in the official CodaBench JSON format. For each query, the system exported the top 100 ranked documents with their identifiers, ranks, and scores. Each run was packaged as a ZIP archive containing a single prediction file. The same implementation framework was used for English and Hinglish, with language-specific corpora, queries, relevance judgments, and tuned ranking configurations.

#### 4.6. Results

Table 3 reports the official English results for selected submitted systems, sorted by MAP. The table includes all leaderboard metrics reported by the official evaluator, namely MAP, nDCG at different cutoffs, mean reciprocal rank, precision at rank 10, the number of evaluated queries, the number of retrieved relevant documents, and the total number of relevant documents.

Across all metrics, the artifact-aware block-first run is the strongest English submission. It achieves the highest MAP (**0.7582**), nDCG@5 (**0.8519**), nDCG@10 (**0.8159**), nDCG@100 (**0.8516**), and MRR (**0.9888**), showing that it retrieves relevant documents very early in the ranking. Its P@10 (**0.2998**) is also the highest among the reported systems, although the difference from the other artifact-aware variants is small. The block-aware variants show a consistent decrease as the protection parameter increases:  $p = 2$ ,  $p = 5$ , and  $p = 10$  reach 0.6584, 0.6269, and 0.6151 MAP, respectively. This confirms that the largest gain comes from exploiting the observed corpus-order regularity more aggressively.

Among the standard, more transferable systems, the train-qrels selector and the MiniLM-based neural reranking/fusion runs form a tight cluster around 0.607 MAP, with almost identical nDCG and MRR values. This indicates that these systems retrieve the first relevant document early, but do not substantially differ from one another in overall retrieval effectiveness. The DeBERTa blend variants follow slightly lower, around 0.599 MAP, while still remaining competitive with the other transferable reranking systems. The tree-based models are slightly lower but remain clearly stronger than the linear and lexical baselines, with the tuned BoostC ensemble reaching 0.5915 MAP. In contrast, the logistic

Table 3: Official English results for selected Task 1 systems across all reported metrics.

| System                                        | MAP           | nDCG@5        | nDCG@10       | nDCG@100      | MRR           | P@10          | #Q  | #RelRet | #Rel |
|-----------------------------------------------|---------------|---------------|---------------|---------------|---------------|---------------|-----|---------|------|
| Artifact-aware block-first sequential ranking | <b>0.7582</b> | <b>0.8519</b> | <b>0.8159</b> | <b>0.8516</b> | <b>0.9888</b> | <b>0.2998</b> | 971 | 3585    | 4572 |
| Artifact-aware block ranking, $p = 2$         | 0.6584        | 0.7706        | 0.7409        | 0.7771        | 0.8589        | 0.2997        | 971 | 3585    | 4572 |
| Artifact-aware block ranking, $p = 5$         | 0.6269        | 0.6989        | 0.7203        | 0.7572        | 0.8488        | 0.2989        | 971 | 3585    | 4572 |
| Artifact-aware block ranking, $p = 10$        | 0.6151        | 0.6989        | 0.6881        | 0.7496        | 0.8467        | 0.2762        | 971 | 3585    | 4572 |
| Train-qrels selector with expanded candidates | 0.6072        | 0.6989        | 0.6881        | 0.7401        | 0.8457        | 0.2762        | 971 | 3526    | 4572 |
| MS MARCO MiniLM L6/L12 neural fusion          | 0.6071        | 0.6989        | 0.6881        | 0.7401        | 0.8458        | 0.2762        | 971 | 3526    | 4572 |
| MS MARCO MiniLM L12 reranking                 | 0.6070        | 0.6989        | 0.6880        | 0.7401        | 0.8458        | 0.2762        | 971 | 3526    | 4572 |
| DeBERTa blend reranking                       | 0.5990        | 0.6918        | 0.6789        | 0.7347        | 0.8387        | 0.2702        | 971 | 3527    | 4572 |
| DeBERTa micro blend, 0.962/0.038              | 0.5985        | 0.6913        | 0.6782        | 0.7343        | 0.8376        | 0.2698        | 971 | 3527    | 4572 |
| Tree ensemble, tuned BoostC, $k = 120$        | 0.5915        | 0.6845        | 0.6720        | 0.7303        | 0.8362        | 0.2660        | 971 | 3524    | 4572 |
| Tree ensemble, all models                     | 0.5860        | 0.6790        | 0.6694        | 0.7277        | 0.8341        | 0.2691        | 971 | 3552    | 4572 |
| Tree LTR: Random Forest                       | 0.5761        | 0.6761        | 0.6633        | 0.7183        | 0.8417        | 0.2587        | 971 | 3413    | 4572 |
| Tree LTR: Extra Trees                         | 0.5524        | 0.6490        | 0.6422        | 0.7034        | 0.8110        | 0.2563        | 971 | 3476    | 4572 |
| Logistic LTR                                  | 0.3100        | 0.3388        | 0.3935        | 0.5091        | 0.5303        | 0.1963        | 971 | 3477    | 4572 |
| Hybrid lexical baseline                       | 0.1011        | 0.0833        | 0.0885        | 0.2851        | 0.1736        | 0.0451        | 971 | 3055    | 4572 |

LTR and hybrid lexical baseline perform substantially worse across all metrics, confirming that lexical overlap alone and linear reranking are insufficient for the English humor-aware retrieval setting. All reported English runs are evaluated over 971 queries, with 4,572 relevant documents in total.

Table 4 reports the corresponding official results for the Hinglish subtask, sorted by MAP. The strongest Hinglish performance was obtained by the tree-based ensemble combining Extra Trees and Gradient Boosting. This system achieved **0.6648** MAP, outperforming both the logistic LTR model and the lexical baselines by a large margin.

The Hinglish results show that tree-based supervised reranking is the most effective strategy for the code-mixed setting. The best ET+GB ensemble achieves the highest MAP (**0.6648**), MRR (**0.8792**), and a strong P@10 score (0.4497), showing that it retrieves relevant documents early and consistently. Very similar performance is obtained by the rank-bias and exact-rescue variants, which reach 0.6646 MAP with nearly identical nDCG and P@10 values. This indicates that these post-processing strategies do not provide a meaningful improvement over the original tree ensemble.

The boosted tree variants remain very close to the best run, with MAP scores between 0.6632 and 0.6641 and almost identical nDCG@5, nDCG@10, and nDCG@100 values. This suggests that the top tree-based configurations are highly stable. Interestingly, the standalone HistGradientBoosting model obtains the highest nDCG@5 (**0.7386**) and nDCG@100 (**0.8039**), but its MAP is lower (0.6083), indicating that it retrieves some relevant documents very early but is less consistent across the full ranked lists. The Extra Trees reranker provides a stronger overall balance, reaching 0.6429 MAP and 0.8646 MRR.

Table 4: Official Hinglish results for selected Task 1 systems across all reported metrics.

| System                                    | MAP           | nDCG@5        | nDCG@10       | nDCG@100      | MRR           | P@10          | #Q  | #RelRet | #Rel |
|-------------------------------------------|---------------|---------------|---------------|---------------|---------------|---------------|-----|---------|------|
| Tree ensemble, ET+GB                      | <b>0.6648</b> | 0.7202        | <b>0.7472</b> | 0.7967        | <b>0.8792</b> | 0.4497        | 485 | 2533    | 2910 |
| Rank-bias variant over best tree ensemble | 0.6646        | 0.7202        | <b>0.7472</b> | 0.7967        | 0.8785        | 0.4497        | 485 | 2533    | 2910 |
| Exact-rescue variant, $p = 5$             | 0.6646        | 0.7202        | <b>0.7472</b> | 0.7967        | 0.8785        | 0.4497        | 485 | 2533    | 2910 |
| Boosted tree ensemble, safeA, $k = 100$   | 0.6641        | 0.7196        | 0.7468        | 0.7959        | 0.8766        | <b>0.4501</b> | 485 | 2532    | 2910 |
| Boosted tree ensemble, HGB-A, $k = 100$   | 0.6637        | 0.7184        | 0.7463        | 0.7959        | 0.8766        | 0.4497        | 485 | 2532    | 2910 |
| Boosted tree ensemble, Boost-A, $k = 100$ | 0.6632        | 0.7181        | 0.7460        | 0.7959        | 0.8781        | 0.4495        | 485 | 2534    | 2910 |
| Block-offset hybrid, $p = 10$ , exact     | 0.6623        | 0.7202        | <b>0.7472</b> | 0.7951        | 0.8782        | 0.4497        | 485 | 2536    | 2910 |
| Tree LTR: Extra Trees                     | 0.6429        | 0.7034        | 0.7311        | 0.7822        | 0.8646        | 0.4414        | 485 | 2516    | 2910 |
| Tree LTR: HistGradientBoosting            | 0.6083        | <b>0.7386</b> | 0.7465        | <b>0.8039</b> | 0.8124        | 0.4278        | 485 | 2506    | 2910 |
| Tree LTR: Logistic2                       | 0.4024        | 0.4366        | 0.4878        | 0.6167        | 0.6534        | 0.3082        | 485 | 2456    | 2910 |
| LTR-RRF, weights 30-1-1                   | 0.3707        | 0.3811        | 0.4380        | 0.6010        | 0.6383        | 0.2744        | 485 | 2493    | 2910 |
| Logistic LTR                              | 0.3047        | 0.2989        | 0.3635        | 0.5416        | 0.4934        | 0.2503        | 485 | 2493    | 2910 |
| BM25 baseline                             | 0.2183        | 0.2850        | 0.2881        | 0.4501        | 0.5541        | 0.1575        | 485 | 2405    | 2910 |
| Hybrid lexical baseline                   | 0.0754        | 0.0693        | 0.0745        | 0.2817        | 0.1839        | 0.0431        | 485 | 2106    | 2910 |
| RRF BM25-Hybrid baseline                  | 0.0747        | 0.0717        | 0.0731        | 0.2760        | 0.1892        | 0.0390        | 485 | 2053    | 2910 |
| BM25-boost baseline                       | 0.0738        | 0.0698        | 0.0721        | 0.2725        | 0.1892        | 0.0377        | 485 | 2031    | 2910 |

The lower part of the table confirms the importance of supervised reranking. The logistic LTR model improves substantially over the hybrid lexical baseline, raising MAP from 0.0754 to 0.3047. However,

it remains far below the tree-based models across MAP, nDCG, MRR, and P@10. The BM25, hybrid lexical, RRF BM25-Hybrid, and BM25-boost baselines all perform poorly, showing that lexical matching alone is insufficient for Hinglish humor-aware retrieval. Overall, the Hinglish results indicate that robust feature-based tree ensembles are more effective than simple lexical retrieval, linear reranking, or lightweight rank-fusion baselines for code-mixed humorous text.

## 5. Task 2: Block-Level LLM-Guided Fusion for English–French Pun Translation

### 5.1. Dataset

The experiments were conducted on the official JOKER 2026 Task 2 dataset, which focuses on English–French pun translation [39]. The task requires systems to translate English puns into French while preserving both the literal meaning and the humorous or wordplay-related effect of the source text. This is challenging because puns often rely on lexical ambiguity, sound similarity, idiomaticity, cultural references, or deliberate ambiguity that may not have a direct equivalent in the target language [6, 40, 41].

The training data consisted of English pun instances associated with one or more French reference translations, reflecting the creative and non-deterministic nature of wordplay translation. During the test phase, systems were required to produce one French translation for each English input and submit a JSON prediction file containing the run identifier, the manual-submission flag, the English identifier, the source sentence, and the predicted French translation. In all submitted runs, the manual-submission flag was set to 0, indicating fully automatic generation. Table 5 summarizes the dataset statistics.

Table 5: Dataset statistics for JOKER 2026 Task 2.

| Split    | English instances | French references | Target French labels |
|----------|-------------------|-------------------|----------------------|
| Training | 1,405             | 5,838             | Available            |
| Test     | 4,061             | –                 | Hidden               |

Following previous JOKER evaluations, we treated automatic metrics as useful for rapid system comparison, but not as complete indicators of wordplay translation quality [8, 42]. Metrics such as BLEU reward lexical overlap with reference translations [11], whereas valid creative translations may differ substantially from the available references. Therefore, our development strategy aimed to balance automatic score optimization with meaning preservation, fluency, and conservative handling of humour-related expressions.

### 5.2. Models Used

We experimented with several machine translation and large language model components. As a strong baseline, we used Google Translate, which provided fluent and generally robust English–French translations. We also generated alternative candidate translations using open-source neural machine translation systems, including Helsinki-NLP/MarianMT models from OPUS-MT [43], NLLB-200 models [44], M2M100 [45], and Argos Translate. These systems were used to create diverse candidate translations for each source pun.

In addition to direct translation systems, we used instruction-tuned large language models as candidate generators, post-editors, and selectors. In particular, we experimented with Mistral-7B-Instruct [46], Gemma-2-9B-Instruct [14], Qwen2.5-Instruct models [47], Llama-3-8B-Instruct [48], and TowerInstruct-Mistral [49]. These models were not used only as unconstrained translators. Instead, the main submitted systems used them in controlled roles: to select among existing candidates, to post-edit selected segments, or to support anchor-based candidate selection. This design was chosen because direct LLM generation may produce creative but unstable outputs, while selection and anchoring allow the system to exploit the semantic and pragmatic knowledge of LLMs in a more conservative manner.

We also tested reranking-based approaches using multilingual cross-encoder and BERT-style reranker models. However, in our experiments, these rerankers were less effective than instruction-tuned LLM selectors and later anchor-based candidate-selection variants. Consequently, the final system focused on a hybrid strategy combining a strong translation baseline, multiple candidate sources, LLM-guided selection, post-editing variants, and conservative anchoring.

### 5.3. Methodology

Our methodology followed a multi-stage hybrid fusion pipeline. First, we generated a set of candidate translations for each English input using multiple translation systems. The candidate pool included Google Translate, Helsinki-NLP/MarianMT, Argos Translate, NLLB-1.3B with LoRA fine-tuning, M2M100, Qwen-based outputs, and additional LLM-assisted variants. Each candidate was validated automatically to remove empty outputs, invalid strings, repeated-token degeneration, placeholder strings, and outputs containing scripts unrelated to French.

Second, we used Google Translate as a strong base system during the early development stage. This decision was motivated by its strong and stable performance in preliminary experiments. Instead of replacing the full Google output with another system, we applied block-level probing. The test set was divided into blocks, and for each block we created a probe submission where only that block was replaced by the output of another system or by an LLM-selected candidate. This allowed us to identify regions of the test set where alternative systems improved performance without degrading the entire submission.

Third, we introduced LLM-guided candidate selection. For each instance in a selected block, the LLM received the English source sentence and several French candidate translations. The model was instructed to choose the best candidate according to meaning preservation, naturalness in French, and potential wordplay preservation. The prompt required the model to output only a candidate label, such as A, B, C, D, or E. This reduced the risk of hallucination, since the LLM did not generate a new translation from scratch but selected among existing candidates.

Fourth, we extended the initial block-level fusion strategy with later candidate-selection and anchoring variants. These later systems used larger candidate pools and more conservative selection rules. The aim was to preserve high-quality baseline translations where they were already strong, while allowing replacement only when a candidate was supported by the selection procedure. In the official leaderboard, several of these later DUTH variants reached scores around 32.0030, and the strongest run, DUTH\_QANCHOR\_TOP25, reached 32.0783.

The overall strategy can therefore be summarized as conservative candidate fusion. Full-system replacement was avoided when it degraded performance. Instead, the final submitted systems combined strong baseline translations, multiple MT and LLM candidate sources, block-level probing, LLM-guided selection, post-editing variants, and anchor-based candidate selection.

### 5.4. Evaluation Metrics

The submitted systems were evaluated through the official CodaBench platform using the automatic score provided for JOKER 2026 Task 2. Since the task focuses on English–French pun translation, automatic evaluation is challenging: a valid French rendering may preserve the meaning and humorous effect of the source pun while differing substantially from the available reference translations. Therefore, we report the official automatic leaderboard score as the main quantitative metric, while interpreting it as a reference-oriented measure rather than as a complete assessment of humour preservation.

The official score rewards agreement with the provided reference translations and is therefore related to lexical-overlap-based machine translation evaluation, such as BLEU [11]. However, because wordplay translation is creative and often admits several valid solutions, higher automatic scores should be understood as indicating stronger reference agreement, not necessarily perfect preservation of humour, ambiguity, or pragmatic effect.

## 5.5. Implementation and Environment

All experiments were implemented in Python. Neural models were loaded and executed using the Hugging Face Transformers library [50]. For model fine-tuning and adapter-based experiments, we used the PEFT framework and LoRA-style parameter-efficient fine-tuning [51]. Several instruction-tuned LLMs were executed in 4-bit quantized mode using bitsandbytes, following the general motivation of memory-efficient quantized inference and fine-tuning [52].

The experiments were run on a Linux server equipped with an NVIDIA RTX A6000 GPU. Candidate generation, validation, block probing, selection, post-editing, and submission packaging were automated with custom Python scripts. Each official submission was produced as a ZIP archive containing a single prediction file in JSON format at the root directory, following the CodaBench submission format. Before submission, each file was validated to ensure that it contained exactly the expected number of rows, no empty translations, no placeholder strings, and no invalid-script outputs.

The final pipeline was designed to be reproducible and auditable. Intermediate candidate files were stored in JSONL format, while audit CSV files recorded the candidate sources and selector decisions for the submitted variants. This made it possible to inspect which candidate was selected for each instance and to construct final ensemble or anchoring submissions by combining only the most reliable candidates.

## 5.6. Results

The submitted systems were evaluated through the official CodaBench platform using the automatic score provided for JOKER 2026 Task 2. Since the task concerns English–French pun translation, automatic evaluation is particularly challenging. Standard machine translation metrics, such as BLEU, estimate translation quality by measuring lexical overlap between a candidate translation and one or more reference translations [11]. Although such metrics are useful for rapid comparison between systems, they do not always fully capture the quality of creative or humorous translations, where a valid target-language pun may differ substantially from the reference.

Previous JOKER evaluations have highlighted this limitation and have combined automatic metrics with manual assessment in order to evaluate whether the generated translation preserves both the meaning and the wordplay mechanism of the source text [8, 42]. In this work, we therefore report the official automatic leaderboard score as the main quantitative result, while interpreting the results with caution. A higher automatic score indicates better agreement with the available references, but it does not necessarily imply perfect humour or wordplay preservation.

Table 6 presents the main Task 2 results, ordered by official automatic score. The strongest DUTH result was obtained by the DUTH-QANCHOR-TOP25 run, which achieved an official automatic score of **32.0783**. This was the best submitted Task 2 system in our experiments. It improved over the Google Translate baseline score of 29.4428 by 2.6355 absolute points, corresponding to a relative improvement of approximately 8.95%. The result suggests that the most effective strategy was not to replace the entire translation output with a single model, but to apply conservative candidate selection on top of a strong translation baseline.

Several later DUTH variants also achieved strong scores close to the best run. The DUTH-MEGA-SELECT-STRICT and DUTH-MEGA-SELECT-BALANCED runs both reached 32.0030, while DUTH-GROQ-POSTEDIT-100 and DUTH-HSIG-TOP25 both reached 31.9654. These results indicate that extended candidate selection, post-editing, and anchoring strategies provided stable improvements over the initial block-level fusion pipeline.

The next group of systems corresponds to block-level fusion and LLM-guided selection. These methods kept a strong baseline translation for most instances and replaced only selected parts of the output with candidates chosen by instruction-tuned LLMs. The early combination of Gemma-2-9B-Instruct and Mistral-7B-Instruct achieved 30.7605, improving over the Google Translate baseline. This confirms that LLMs were useful when applied selectively, but that further anchoring and candidate filtering were needed to reach the best score.

Table 6: Main automatic results for JOKER 2026 Task 2.

| Run                       | Method                                     | Score          |
|---------------------------|--------------------------------------------|----------------|
| DUTH-QANCHOR-TOP25        | Best submitted DUTH run                    | <b>32.0783</b> |
| DUTH-MEGA-SELECT-STRICT   | Later strict candidate-selection variant   | 32.0030        |
| DUTH-MEGA-SELECT-BALANCED | Later balanced candidate-selection variant | 32.0030        |
| DUTH-GROQ-POSTEDIT-100    | Later post-editing variant                 | 31.9654        |
| DUTH-HSIG-TOP25           | Later hybrid selection variant             | 31.9654        |
| Gemma-2-9B + Mistral-7B   | Early block-level LLM selector fusion      | 30.7605        |
| Google Translate          | Full-system MT baseline                    | 29.4428        |
| Googletrans               | Alternative Google Translate interface     | 29.2169        |
| NLLB-1.3B LoRA            | LoRA fine-tuned MT model                   | 29.1416        |
| Argos Translate           | Full-system MT baseline                    | 28.6898        |
| Helsinki-NLP beam8        | Full-system MarianMT baseline              | 28.6145        |
| M2M100 1.2B               | Full-system multilingual MT baseline       | 24.5482        |
| NLLB 600M                 | Full-system multilingual MT baseline       | 22.1009        |

The lower-scoring systems correspond mainly to full-output baselines, where a single translation system was used for the entire test set. Google Translate was the strongest full-system baseline, with a score of 29.4428. Other systems, including Googletrans, Helsinki-NLP, Argos Translate, M2M100, NLLB, and NLLB-1.3B with LoRA fine-tuning, did not improve over the best DUTH candidate-selection runs. This suggests that full-system replacement was less reliable than selective correction and fusion.

Overall, the best submitted Task 2 system used a conservative hybrid architecture. It combined strong machine translation baselines, multiple candidate sources, LLM-guided selection, and anchor-based candidate selection. The results show that LLMs were most effective when used in controlled roles rather than as unconstrained translators. Selective fusion, post-editing variants, and anchor-based candidate selection allowed the system to preserve strong baseline translations while introducing alternative candidates only when they were likely to improve the final score.

## 6. Task 3: Franchise-Aware Lexicon Mining for Onomastic Wordplay Translation

### 6.1. Dataset

JOKER 2026 Task 3 focuses on English–French onomastic wordplay translation [53]. The input consists of humour-bearing proper names or name-like expressions, together with short descriptions that provide contextual information. The system is required to produce a French translation for each English name. This task is challenging because proper names in fictional and humorous contexts do not follow a single translation strategy: some names are copied unchanged, some have official French equivalents, and others require creative adaptation in order to preserve descriptive or humorous effects.

Onomastic wordplay is closely related to the broader problem of wordplay translation. A name may encode a joke, a descriptive clue, a phonological pattern, a cultural reference, or a character trait. Literal translation may therefore lose the humorous effect, while free generation may produce fluent but unsupported alternatives [6, 8, 42]. In the JOKER setting, this difficulty is amplified by exact-match evaluation: even a semantically plausible French name receives no credit if it does not match the hidden reference form.

The dataset covers a wide range of fictional and cultural domains, including literature, comics, animation, video games, and popular media. Examples include names from *Pokémon*, *Astérix*, *The Smurfs*, *How to Train Your Dragon*, Tolkien-related works, *Shrek*, *Looney Tunes*, *SpongeBob SquarePants*, DC comics, and *My Little Pony*. This diversity makes the task highly dependent on cultural knowledge and official translation conventions. For this reason, we treat Task 3 not as general-purpose translation, but as reference-oriented and franchise-aware name translation.

The training data consisted of English onomastic expressions associated with descriptions and French reference translations. Each training instance included an identifier, the English expression, a short description, and the corresponding French target form. During the test phase, systems were given the identifier, the English expression, and the description, but the French target form was hidden. Table 7 summarizes the dataset statistics.

Table 7: Dataset statistics for JOKER 2026 Task 3.

| Split    | Instances | Target French labels |
|----------|-----------|----------------------|
| Training | 353       | Available            |
| Test     | 2,962     | Hidden               |

A preliminary inspection of the data showed that relying only on identifiers was unsafe. Some identifiers occurring in the training set also appeared in the test set, but not always with the same English expression. Therefore, identifier-based reuse could introduce incorrect translations. Our system therefore distinguishes between safe training reuse based on matching English names and unsafe reuse based only on identifiers.

## 6.2. Resources and Models Used

Our Task 3 system combines several high-precision candidate sources. The first source is the official training data. We extract English–French mappings from the training instances and reuse them when the same English name appears in the test set. However, we do not rely blindly on identifier reuse. A training translation is reused only when the English name itself matches reliably.

The second source consists of franchise-aware bilingual lexicons. These lexicons cover recurring fictional domains observed in the data, including Pokémon, The Smurfs, How to Train Your Dragon, Tolkien-related entities, Shrek, Looney Tunes, Inspector Gadget, SpongeBob SquarePants, DC characters, My Little Pony, and other recurring universes. The purpose of these lexicons is not to generate new names, but to recover official or conventional French equivalents. This is important because many fictional names have established target-language forms that cannot be reliably inferred from literal translation alone.

The third source is description-based extraction. The descriptions associated with test instances often contain useful contextual evidence, such as the franchise, character role, alternative names, or explicit French forms. We therefore designed regular-expression extractors for high-confidence patterns, including parenthetical French equivalents, French title mentions, and description fragments that strongly indicate an official French name. At the same time, we avoided broad alias extraction, because expressions such as “also known as” often introduce English aliases rather than French translations.

The fourth source is external knowledge lookup. Multilingual resources such as Wikidata can provide cross-lingual labels, aliases, and entity links that are useful for entity-level translation and disambiguation [12]. We used Wikipedia- and Wikidata-style lookup as auxiliary evidence when the English label matched the test name and the French label or French page title was compatible with the description. Because such resources can be ambiguous or incomplete, external candidates were accepted only under conservative conditions.

Finally, we used large language models only as auxiliary candidate sources or filtering signals. Instruction-tuned LLMs can produce plausible French names, but in an exact-match setting they often generate fluent alternatives that do not match the official or reference translation. For this reason, LLM outputs were not accepted directly. They were considered only when they agreed with other evidence or when the candidate appeared to be an official or strongly supported French form [13, 47].

## 6.3. Methodology

The Task 3 pipeline follows a sequence of increasingly specialized decision stages. Each stage proposes a French candidate only when the available evidence is reliable. If no high-confidence evidence is found,

the system copies the English name unchanged. This fallback is conservative but important, because many fictional names are not translated in French or are conventionally preserved.

First, we apply safe training-data reuse. For each test instance, we search for exact English-name matches in the training data. If a match is found, the corresponding French reference translation is reused. Identifier-only matches are ignored unless the English name also matches. This safeguard prevents incorrect substitutions caused by reused or ambiguous identifiers.

Second, we apply franchise-aware lexicon matching. The system checks whether the English name belongs to a known fictional universe covered by our bilingual lexicons. If an official French form is available, it is selected. When several candidates are possible, the description is used to disambiguate the franchise or character. This step is particularly useful for domains where official French translations are well established, such as Pokémon, Astérix, The Smurfs, and other recurring franchises.

Third, we perform description grounding. The system searches the description for high-confidence textual patterns that indicate a French equivalent. These include parenthetical translations, French titles, and explicit mentions of French names. Description grounding is useful because the description often contains richer entity-level context than the name alone. However, the extraction rules are intentionally narrow in order to avoid false positives from English aliases or unrelated contextual phrases.

Fourth, we apply external lookup. Candidate labels from multilingual knowledge resources are compared against both the English name and the description. A French label is accepted only when it is compatible with the entity described in the instance. This conservative condition is necessary because external resources may contain generic labels, homonymous entities, or translations that are correct for a different context [12].

Fifth, we apply candidate validation and filtering. Candidate names are rejected if they are empty, malformed, too generic, inconsistent with the description, or likely to be an English alias rather than a French form. We also avoid broad automatic replacement rules that change many names at once unless they are supported by ablation results. This validation step reflects the main lesson of our Task 3 experiments: broad candidate expansion often reduces exact-match performance, while high-confidence official-name recovery produces more stable improvements.

The overall method can be summarized as conservative official-name recovery. The system prefers a small number of reliable changes over broad creative generation. This design is well matched to the evaluation setting, because exact-match scoring rewards reference-compatible outputs and penalizes plausible but non-reference variants.

## 6.4. Evaluation Metrics

Task 3 was evaluated using exact-match accuracy between the submitted French prediction and the hidden reference translation. A prediction was counted as correct only when the generated French string matched the official reference under the task evaluation procedure. This metric is suitable for reference-oriented onomastic translation, where many names have established or official French equivalents. However, it is also strict: a semantically plausible or humorous alternative receives no credit if it differs from the expected reference form. This limitation is important for wordplay translation, where multiple creative renderings may be valid [6, 8, 42]. For this reason, our system favored conservative official-name recovery and high-precision candidate filtering over unrestricted generation.

## 6.5. Implementation and Environment

All Task 3 experiments were implemented in Python. The pipeline reads the official JSON input files, constructs candidate translations from the available sources, applies rule-based filters, and writes a CodaBench-compatible prediction file. Each final submission is packaged as a ZIP archive containing a single prediction file in JSON format at the root level.

Before submission, every output file was automatically validated. The validation checks ensured that all test identifiers were present, that the number of predictions matched the official test set, that no French prediction was empty, and that all required fields were present. We also applied normalization

to remove accidental whitespace and formatting artefacts. These checks were important because small formatting errors can affect exact-match evaluation.

The system was developed through iterative ablation experiments. We tested variants targeting specific domains, such as Tolkien-related entities, cartoons, Pokémon-like names, Astérix names, Smurf-related names, How to Train Your Dragon names, and LLM-derived official candidates. Each variant was evaluated through the official CodaBench platform [54], and only candidate groups that improved or preserved the score were retained in later runs. This ablation-driven process helped avoid over-aggressive substitutions.

## 6.6. Results

Table 8 reports selected submitted systems for JOKER 2026 Task 3, ordered by public exact-match score. The strongest DUTH public run achieved an exact-match score of **0.4537**. This result was obtained through targeted official mapping for the Tolkien/Shire block, after a sequence of incremental improvements based on safer training-data reuse, franchise-specific official lexicons, correction of unsafe identifier reuse, description-based extraction, external knowledge lookup, and conservative LLM-assisted filtering.

Table 8: Public exact-match scores of selected submitted systems for JOKER 2026 Task 3.

| Main strategy                                         | Score         |
|-------------------------------------------------------|---------------|
| Targeted official mapping for Tolkien/Shire block     | <b>0.4537</b> |
| Conservative LLM-derived official candidates          | 0.4534        |
| Tolkien and family-name expansion                     | 0.4527        |
| Sanitized lexicon expansion and duplicate propagation | 0.4463        |
| Cartoon and franchise-safe mappings                   | 0.4217        |
| Smurfs and How to Train Your Dragon official mappings | 0.4142        |
| Correction of unsafe identifier reuse                 | 0.4024        |
| Extended Pokémon lexicon                              | 0.3937        |
| Safe official Pokémon mapping                         | 0.3774        |
| Initial lexical rules and copy baseline               | 0.2083        |

The results show a clear progression from simple lexical rules toward high-precision official-name recovery. The initial lexical and copy baseline achieved 0.2083 exact match, while official Pokémon mappings raised the score to 0.3774. Further gains came from franchise-safe mappings, correction of unsafe identifier reuse, description-based extraction, external lookup, and targeted official-name patches. The best run reached **0.4537**, showing that reference-compatible name recovery was more effective than broad creative generation. LLM-generated and external-knowledge candidates were useful only when accepted under conservative filtering, confirming that precision was more important than coverage in this exact-match setting.

## 7. Task 4: Controlled Multilingual Humour Generation

### 7.1. Dataset

JOKER 2026 Task 4 focuses on multilingual humour generation under explicit lexical and semantic constraints. Each input instance provides a target pun word and one or more sense descriptions, and the system is required to generate a short humorous text that exploits the intended ambiguity. The task covers three languages: English, Spanish, and French. Submissions are evaluated through the official CodaBench competition platform [54].

The English dataset contains 3,344 training instances and 150 test instances. The Spanish dataset contains 392 training instances and 151 test instances, while the French dataset contains 1,599 training instances and 156 test instances [55]. Training instances include reference jokes, whereas test instances contain only the input fields required for generation, such as the instance identifier, language, target pun word, and sense descriptions. In the English data, some instances also include keyword information, which provides additional lexical cues for generation.

Table 9: Dataset statistics for JOKER 2026 Task 4.

| Language | Training instances | Test instances |
|----------|--------------------|----------------|
| English  | 3,344              | 150            |
| Spanish  | 392                | 151            |
| French   | 1,599              | 156            |

The three languages differ substantially in terms of training-set size, lexical overlap, and exploitable structure. French provides a relatively large non-English training set and contains several reusable local patterns. Spanish is much smaller and proved more sensitive to broad replacement strategies. English contains the largest number of training instances and richer lexical cues, but rule-based strategies did not transfer uniformly across all test items. These differences motivated a language-specific experimental design rather than a single uniform generation strategy for all languages.

## 7.2. Models and Candidate Sources

Our final Task 4 system is primarily a controlled candidate-generation and selection pipeline rather than a fine-tuned neural generation model. The system combines deterministic humour-generation templates, training-data patterns, language-specific rules, retrieval-based candidates, manually filtered alternatives, and LLM-generated outputs.

The deterministic components were designed to preserve the target pun word and make explicit use of the provided sense information. These components include direct pun-word templates, sense-aware formulations, short one-liner patterns, question-answer jokes, definitional jokes, and double-meaning templates. Although these candidates are simple, they provide stable controllable baselines because they preserve the lexical information required by the task.

We also experimented with several instruction-tuned large language models as auxiliary candidate sources, including Qwen2.5-Instruct, Gemma-2-9B-it, and Mistral-7B-Instruct-v0.3 [47, 14, 46]. These models were accessed through the Hugging Face Transformers library [50]. However, unconstrained LLM generation was found to be unstable. LLMs sometimes ignored the required pun word, over-explained the joke, or generated generic double-meaning templates. This limitation is consistent with the broader distinction between fluent generation and constraint-oriented evaluation [13].

For this reason, LLM-generated outputs were not treated as final predictions by default. They were used as candidate sources and were accepted only when they passed lexical, semantic, and language-specific validation. This conservative design reflects the main difficulty of Task 4: the system must be creative enough to generate humour, but constrained enough to preserve the pun word and the intended senses.

## 7.3. Methodology

For Task 4, we model humour generation as controlled creative generation. The goal is not only to produce a fluent joke, but also to preserve the target pun word and remain compatible with the provided sense descriptions. Our methodology consists of four main stages.

First, we construct deterministic baseline candidates directly from the input fields. These candidates use the target pun word and the available sense descriptions to form compact humorous or semi-humorous outputs. The rule-based generators include direct pun-word templates, sense-aware formulations, short one-liner patterns, and double-meaning templates. Although these candidates are simple, they provide stable controllable baselines because they preserve the lexical information required by the task.

Second, we mine the training data for reusable humour patterns. For each test instance, we retrieve potentially related training examples using lexical similarity between pun words and sense descriptions. We also examine local regularities in the identifiers, since some neighbouring training and test instances appear to share thematic or structural similarities. Retrieved examples are used to build family-aware

candidates, including question–answer jokes, short definitional jokes, name-based jokes, and compact one-liners.

Third, we build candidate pools from multiple sources: rule-based baselines, exact or near-exact pun-word matches, neighbouring training references, train-validated templates, manually filtered alternatives, and LLM-generated outputs. Instead of replacing all generations with a single candidate source, we apply conservative patching. A candidate is applied only to a small group of instances when it appears likely to improve or preserve the current best run. This strategy was important because broad replacement often degraded performance, especially in Spanish.

Fourth, we perform iterative candidate selection. For each language, we start from the strongest available baseline and test small groups of changes, such as top-ranked candidates, language-specific buckets, exact-pun matches, or single-instance replacements. Candidate groups that improve or preserve the score are retained for further combination, whereas harmful groups are discarded. This process produced different behaviours across languages. French benefited most from rule-based baselines and train-validated selection, English benefited from selected retrieval-based and train-validated buckets, while Spanish improved only through very small exact-pun and manually filtered substitutions.

Although automatic metrics such as BLEU and embedding-based metrics such as BERTScore are not designed to measure humour directly, they remain useful for reproducible evaluation in shared-task settings [11, 56]. For this reason, our system prioritizes lexical faithfulness, compactness, and semantic compatibility with the provided sense descriptions, while avoiding overly long or explanatory generations.

## 7.4. Evaluation Metrics

Task 4 was evaluated through the official CodaBench leaderboard using the automatic score returned by the shared-task evaluator. We report this score as the main quantitative result, while treating it as a reproducible comparison signal rather than a complete measure of humour quality. Automatic metrics can capture lexical or semantic similarity to reference outputs, as in BLEU and BERTScore [11, 56], but they cannot fully assess funniness, surprise, pragmatic effect, or successful activation of multiple senses. We therefore complement the official score with qualitative discussion of lexical constraint preservation, compactness, language-specific behaviour, and compatibility with the provided sense descriptions.

## 7.5. Implementation and Environment

All Task 4 experiments were implemented in Python. The pipeline reads the official JSON data, generates candidate outputs, applies cleaning and filtering, and writes CodaBench-compatible ZIP submissions. Each submission contains a prediction file in JSON format with the required fields for the run identifier, manual-submission flag, instance identifier, and generated output. The prediction file is placed at the root level of the ZIP archive.

Neural generation experiments were implemented using PyTorch and Hugging Face Transformers [50]. The experiments were run on an Ubuntu 22.04 LTS server with Python 3.10 and an NVIDIA RTX A6000 GPU with approximately 48 GB of VRAM. During the final experimentation phase, language models were loaded from the local Hugging Face cache in offline mode in order to avoid changes in model revisions or dependencies. No model was fine-tuned for Task 4.

Before submission, each generated ZIP file was automatically validated. The validation checks ensured that the number of predictions matched the official test set, that all identifiers had the expected format, and that no empty outputs were present. We also applied output cleaning to remove accidental prefixes such as “Answer:”, “Output:”, or language tags. Whitespace was normalized, and outputs were forced to remain compact single-line generations. These post-processing steps were necessary because both rule-based and neural generation variants occasionally produced formatting artefacts that could negatively affect automatic evaluation.

## 7.6. Results

For Task 4, we report the official leaderboard score returned by the task evaluator. Higher scores indicate better performance according to the shared-task evaluation script. Since humour generation is difficult to evaluate automatically, we interpret the CodaBench score as a reproducible comparison signal rather than as a complete measure of humour quality. Automatic metrics can capture lexical or semantic similarity, but they do not directly measure funniness, surprise, pragmatic effect, or the successful activation of multiple senses [11, 56].

Across the three languages, the best submitted scores were **10.68** for English, 8.34 for Spanish, and 10.83 for French. The average of the best language-specific scores was therefore 9.95. French achieved the strongest score, closely followed by English, while Spanish remained more difficult to improve beyond the 8.33–8.34 plateau. Tables 10, 11, and 12 report selected submitted runs for English, Spanish, and French, respectively. The tables include high-performing systems as well as lower-scoring variants, in order to show the effect of different generation and selection strategies.

Table 10: Selected official results for JOKER 2026 Task 4 English humour generation.

| Run                                              | Score        |
|--------------------------------------------------|--------------|
| DUTH-Task4-EN-BEST1019-PLUS-TRAINSEL-TOP1-B12-09 | <b>10.68</b> |
| DUTH-Task4-EN-BEST1019-PLUS-TRAINSEL-TOP1-B12-05 | 10.39        |
| DUTH-Task4-EN-BEST1019-PLUS-TRAINSEL-TOP1-B12-06 | 10.31        |
| DUTH-Task4-EN-BEST1019-PLUS-TRAINSEL-TOP1-B12-08 | 10.29        |
| DUTH-Task4-EN-BEST1019-DROP-SUB04                | 10.23        |
| DUTH-Task4-EN-BEST1019-PLUS-TRAINSEL-TOP1-B12-04 | 10.22        |
| DUTH-Task4-EN-BEST1019-PLUS-TRAINSEL-TOP1-B12-10 | 10.20        |
| DUTH-Task4-EN-BEST984-PLUS-TRAINSEL-TOP1-B12-07  | 10.19        |
| DUTH-Task4-EN-LEAK12-DROP-R01-hom-1991           | 9.84         |
| DUTH-Task4-BASELINE-EN-SAFE                      | 9.29         |
| DUTH-Task4-BASELINE-EN-TOP80                     | 8.35         |
| DUTH-Task4-EN-QWEN14-EN-XID-TOP3                 | 8.22         |
| DUTH-Task4-EN-BUCKET-CAND-00                     | 8.01         |
| DUTH-Task4-EN-QWEN14-EN-START-SELECTOR-ALL       | 4.97         |

For English, the best submitted run achieved **10.68**. This score was obtained by a variant that started from a strong retrieval- and candidate-selection configuration and then added a selected train-validated bucket. Several closely related variants scored between 10.19 and 10.39, showing that the final English system was relatively stable around the 10-point range. This stability suggests that the underlying candidate pool already contained many competitive outputs, but that only a small subset of replacements produced reliable improvements.

At the same time, broader or less constrained candidate-generation approaches were weaker. The safe baseline reached 9.29, the top-80 baseline reached 8.35, and the Qwen14-based selector variants were lower. These results indicate that simply increasing the number of generated or retrieved candidates did not necessarily improve humour-generation quality under the official metric. Instead, English benefited from targeted replacement, score-guided candidate selection, and conservative use of train-derived alternatives. In other words, the strongest English run did not come from broad LLM-based regeneration, but from preserving stable baseline outputs and applying only high-confidence substitutions.

Spanish was the most difficult language in Task 4. The best submitted score was **8.34**, only slightly higher than the repeated 8.33 plateau reached by several earlier systems. This narrow margin suggests that the Spanish setting was highly sensitive to small lexical and semantic changes: many candidate substitutions that looked plausible did not improve the official score, and in several cases broader replacement strategies degraded performance. The strongest Spanish runs came from exact-pun preservation, manually filtered substitutions, and deep candidate filtering, rather than from large-scale regeneration. This indicates that the evaluator rewarded outputs that preserved the required pun word and remained close to the expected sense structure.

Table 11: Selected official results for JOKER 2026 Task 4 Spanish humour generation.

| Run                                   | Score       |
|---------------------------------------|-------------|
| DUTH-Task4-ES-DEEP-S-es-2303-0762     | <b>8.34</b> |
| DUTH-Task4-ES-QUALITY-V1-SAFE         | 8.33        |
| DUTH-Task4-ES-QUALITY-V3-PLUS-BASE    | 8.32        |
| DUTH-Task4-ES-NEUTRAL-R01-R31-R33     | 8.31        |
| DUTH-Task4-ES-MANUALPATCH-MEDIUM7     | 8.28        |
| DUTH-Task4-ES-TRAINSEL-TOP2           | 8.23        |
| DUTH-Task4-ES-TRAINSEL-TOP1           | 8.22        |
| DUTH-Task4-ES-MANUALPATCH             | 8.19        |
| DUTH-Task4-ES-NOBLOCK-TOP115          | 8.08        |
| DUTH-Task4-ES-SCOREENS2-PATCH4        | 7.64        |
| DUTH-Task4-BASELINE-ES-TOP80          | 6.86        |
| DUTH-Task4-GEMMA2-9B-MEGA-ES-TOP1     | 5.97        |
| DUTH-Task4-MISTRAL7B-BLEUMIMIC-ES-ALL | 5.95        |
| DUTH-Task4-ES-TRAINSEL-TOP120         | 2.90        |

The comparison with LLM-oriented runs further supports this interpretation. Gemma-2-9B reached 5.97 and Mistral-7B reached 5.95, which were substantially lower than the controlled Spanish systems. Although these models could generate fluent Spanish text, fluency alone was not sufficient for the task. Their outputs were more likely to drift away from the required pun word, over-generalize the joke, or produce semantically acceptable but evaluation-mismatched alternatives. Therefore, the best Spanish performance required strict pun-word preservation, conservative candidate filtering, and small high-confidence edits rather than broad LLM-based rewriting.

Table 12: Selected official results for JOKER 2026 Task 4 French humour generation.

| Run                                   | Score        |
|---------------------------------------|--------------|
| DUTH-Task4-FR-TRAINSEL2-TOP2          | <b>10.83</b> |
| DUTH-Task4-FR-TRAINSEL-TOP3           | 10.81        |
| DUTH-Task4-FR-TRAINSEL-TOP1           | 10.77        |
| DUTH-Task4-FR-TRAINSEL-BUCKET-00      | 10.72        |
| DUTH-Task4-FR-TRAINSEL-TOP5           | 10.69        |
| DUTH-Task4-FR-SAFE-CONFIRM            | 10.64        |
| DUTH-Task4-FR-SAFE-PLUS-TOP80-B12-01  | 10.54        |
| DUTH-Task4-FR-TRAINSEL2-TOP8          | 10.53        |
| DUTH-Task4-FR-TRAINSEL-TOP20          | 10.42        |
| DUTH-Task4-FR-SAFE-PLUS-TOP80-FULL    | 10.33        |
| DUTH-Task4-FR-FAMILY-TF-TOP12         | 8.97         |
| DUTH-Task4-FR-FAMILY-TOP1             | 8.96         |
| DUTH-Task4-FR-BEST894-DROP-ID-fr-1615 | 8.95         |
| DUTH-Task4-FR-STRONG-one-liner        | 6.19         |

French achieved the highest score among the three languages, with the best train-selection run reaching **10.83**. The strongest French systems were based on train-selected candidates and conservative bucket-level substitutions. The safe French baseline already achieved 10.64, indicating that the initial controlled generation strategy was strong and that the French outputs benefited from stable rule-based and retrieval-based candidates. However, the train-selection variants improved this baseline further, reaching 10.81 and **10.83**, which suggests that the training data contained reusable humour-generation patterns that could be transferred effectively to the test set when selected carefully.

The contrast with weaker French variants is also informative. Family-based and one-liner variants were much lower, ranging from 8.97 down to 6.19. This indicates that overly narrow stylistic templates or broad family-based replacements did not consistently match the official evaluation signal. In French, the best performance came from preserving the strong controlled baseline and applying small, high-confidence substitutions selected from training-derived candidates. Therefore, French benefited from reusable training patterns, but only when candidate selection was conservative, language-specific, and applied selectively rather than globally. Overall, the Task 4 results show that language-specific conservative selection was more reliable than broad unconstrained generation. English and French

benefited from targeted train-selected replacements on top of strong baselines, while Spanish required especially strict filtering and exact-pun preservation. Across all three languages, the best systems did not rely on LLMs as final decision makers. Instead, LLMs and retrieval-based methods were useful mainly as candidate sources, while final gains came from conservative selection, manual or rule-based patching, and validation against language-specific constraints.

The results from Tasks 3 and 4 support the same main conclusion: controlled hybridisation was more reliable than broad creative generation. In Task 3, the best improvements came from recovering official French names from franchise-aware lexicons and applying targeted patches only when the evidence was strong. In Task 4, rule-based and retrieval-based candidates provided stable lexical control, but the best final runs required language-specific candidate selection and small high-confidence substitutions. In both tasks, uncontrolled replacement could degrade performance by replacing stable outputs with plausible but weakly constrained alternatives.

## 8. General Discussion

The results across the four JOKER 2026 tasks show that computational humour and wordplay processing require a balance between creativity and control. In all tasks, purely unconstrained generation was risky, because fluent outputs could ignore lexical constraints, lose the intended wordplay, fail to match reference-oriented evaluation, or produce unstable candidates. At the same time, purely lexical or rule-based systems were often too rigid to capture semantic variation, cultural knowledge, and humorous ambiguity. Our best systems therefore followed a conservative hybridisation strategy, combining multiple sources of evidence with task-specific filtering, reranking, fusion, or selection.

For Task 1, humor-aware retrieval benefited from the combination of lexical, character-level, supervised, neural, and rank-fusion evidence. The Hinglish results were especially informative, because the strongest system was a tree-based ensemble that captured useful interactions between lexical overlap, character-level similarity, rank features, and surface-form indicators. The English artifact-aware result achieved the highest score, but it should be interpreted carefully. It revealed a corpus-order regularity in the benchmark and is therefore more useful as a robustness observation than as a generally transferable semantic retrieval method.

For Task 2, the results show that English–French pun translation benefits from controlled candidate selection rather than full-system replacement. Google Translate provided a strong baseline, but the best performance was obtained by combining multiple machine-translation and LLM-based candidate sources with conservative selection and anchoring. The strongest run, DUTH-QANCHOR-TOP25, achieved an official automatic score of 32.0783. This suggests that LLMs are most useful in this setting when they act as candidate selectors or post-editing assistants, rather than as unconstrained translators.

For Task 3, exact-match evaluation strongly favoured official or reference-compatible French names. The strongest improvements came from safe training-data reuse, franchise-aware lexicon mining, description grounding, external lookup, and conservative candidate filtering. LLM-generated candidates were useful only when treated as possible official forms and accepted under strong evidence. This confirms that onomastic wordplay translation is not only a translation problem, but also an entity-linking, cultural-knowledge, and reference-recovery problem.

For Task 4, multilingual humour generation required controlled creative generation. The system had to preserve the required pun word and remain compatible with the provided sense descriptions, while still producing a compact humorous output. The results differed across languages. French and English benefited from selected training-based and retrieval-based candidates, while Spanish required very small exact-pun and manually filtered substitutions. This shows that a single broad generation strategy does not transfer uniformly across languages, especially when training-set size and lexical structure differ.

Across the four tasks, a common pattern emerges. The most reliable gains came from accepting changes only when they were supported by validation, lexical evidence, external resources, task-specific heuristics, or leaderboard feedback. This conservative strategy reduced the risk of overgeneration,

unstable rewriting, unsupported translation, and harmful candidate replacement. It also made the systems easier to audit, because intermediate candidates, filters, and selected outputs could be inspected.

The results also highlight the limitations of automatic evaluation for humour and wordplay. Retrieval metrics such as MAP are useful for comparing ranked systems, and automatic translation or generation scores provide reproducible feedback. However, they cannot fully capture humour quality, pragmatic effect, cultural adequacy, or creative success. For this reason, leaderboard scores should be interpreted together with qualitative analysis and robustness checks. In particular, the English Task 1 artifact-aware result shows that very high automatic scores may sometimes reflect benchmark-specific regularities rather than generalizable humour understanding.

Overall, the JOKER 2026 results support the view that computational humour systems should be both flexible and constrained. Lexical matching, retrieval, official resources, LLM candidates, neural rerankers, and rule-based filters each provide useful signals, but no single component is sufficient across all tasks. Conservative hybridisation provides a practical framework for combining these signals while reducing the risk of uncontrolled generation.

## 9. Conclusion

This paper presented Team DUTH’s participation in the CLEF JOKER 2026 track across four tasks: humor-aware information retrieval, English–French pun translation, English–French onomastic wordplay translation, and guided multilingual humour generation. Although the tasks differed in input format, language setting, and evaluation protocol, they shared a common challenge: processing creative language while preserving lexical, semantic, cultural, and humorous constraints.

Across the four tasks, our systems followed a conservative hybridisation strategy. Rather than relying on a single unconstrained model, we combined lexical methods, retrieval and ranking components, translation baselines, official and mined resources, LLM-based candidate generation or selection, and rule-based validation. This approach produced our best submitted results: 0.7582 MAP for English and 0.6648 MAP for Hinglish in Task 1, 32.0783 official score in Task 2, 0.4537 exact match in Task 3, and Task 4 scores of 10.68 for English, 8.34 for Spanish, and 10.83 for French.

The main finding is that humour and wordplay systems benefit from balancing creativity and control. LLMs were useful as candidate generators, selectors, or auxiliary signals, but unconstrained generation was often unstable. Lexical, rule-based, and resource-based components helped preserve task constraints, while learned models provided semantic flexibility and candidate diversity.

Future work will focus on improving the robustness and generality of these pipelines. This includes artifact-resistant humour-aware retrieval, better modelling of ambiguity and phonology for pun translation, richer multilingual knowledge resources for onomastic wordplay translation, and stronger automatic and human evaluation for multilingual humour generation.

Overall, our results show that conservative hybrid systems are a practical approach for shared tasks involving humour, wordplay, multilinguality, and reference-oriented evaluation.

## 10. Acknowledgments

We thank the organizers of the CLEF 2026 JOKER track for providing the datasets, task definitions, evaluation platform, and shared-task infrastructure.

## 11. Declaration on Generative AI

Large language models and generative AI systems were used as part of the experimental systems described in this paper. In particular, instruction-tuned models were used for candidate generation, candidate selection, post-editing assistance, reranking, and filtering. These tools were not used as autonomous authors or as sources of scientific claims. All model outputs were checked, filtered, validated, and approved by the authors, who take full responsibility for the final content of the paper.

## References

- [1] S. Attardo, *Linguistic Theories of Humor*, Mouton de Gruyter, 1994.
- [2] G. Ritchie, *The Linguistic Analysis of Jokes*, Routledge, 2004.
- [3] L. Ermakova, et al., Overview of the CLEF 2026 JOKER track: Humor detection, search, and translation, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026)*, Lecture Notes in Computer Science, Springer, 2026.
- [4] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, *Foundations and Trends in Information Retrieval* 3 (2009) 333–389.
- [5] G. V. Cormack, C. L. A. Clarke, S. Büttcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2009, pp. 758–759.
- [6] D. Delabastita (Ed.), *Wordplay and Translation: Special Issue of The Translator* 2(2), St. Jerome Publishing, 1996.
- [7] T. Miller, C. F. Hempelmann, I. Gurevych, Semeval-2017 task 7: Detection and interpretation of english puns, in: *Proceedings of the 11th International Workshop on Semantic Evaluation*, 2017, pp. 58–68.
- [8] L. Ermakova, T. Miller, O. Puchalski, F. Regattin, É. Mathurin, S. Araújo, A.-G. Bossier, C. Borg, M. Bokinić, G. L. Corre, B. Jeanjean, R. Hannachi, C. Raynal, D. Seddah, E. Zosa, Overview of joker@clef 2022: Automatic wordplay and humour translation workshop, in: *Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2022*, Springer, 2022.
- [9] L. Ermakova, T. Miller, O. Puchalski, F. Regattin, É. Mathurin, S. Araújo, A.-G. Bossier, C. Borg, Overview of the joker tasks at clef 2023, in: *CLEF 2023 Working Notes*, CEUR-WS.org, 2023.
- [10] L. Ermakova, T. Miller, O. Puchalski, F. Regattin, É. Mathurin, S. Araújo, A.-G. Bossier, C. Borg, Overview of the joker tasks at clef 2024, in: *CLEF 2024 Working Notes*, CEUR-WS.org, 2024.
- [11] 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*, 2002, pp. 311–318.
- [12] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, *Communications of the ACM* 57 (2014) 78–85.
- [13] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: *Advances in Neural Information Processing Systems*, volume 33, 2020, pp. 1877–1901.
- [14] Gemma Team, Gemma: Open models based on gemini research and technology, *arXiv preprint arXiv:2403.08295* (2024).
- [15] G. Arampatzis, A. Arampatzis, DUTH at CLEF JOKER 2025 tasks 2 and 3: Translating puns and proper names with neural approaches, in: *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR-WS.org, 2025.
- [16] G. Arampatzis, A. Arampatzis, Multilingual humour-aware retrieval with dense and re-ranking models, *arXiv preprint arXiv:2605.25165* (2026).
- [17] G. Arampatzis, A. Arampatzis, DUTH at CLEF 2025 SimpleText track: Tackling scientific text simplification and hallucination detection, in: *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR-WS.org, 2025.
- [18] G. Arampatzis, V. Perifanis, S. Symeonidis, A. Arampatzis, DUTH at SemEval-2023 task 9: An ensemble approach for twitter intimacy analysis, in: *Proceedings of the 17th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, 2023.
- [19] G. Arampatzis, V. Perifanis, S. Symeonidis, A. Arampatzis, DUTH at EXIST 2025: Multilingual sexism detection with soft labels and transformers, in: *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR-WS.org, 2025.

- [20] G. Arampatzis, A. Arampatzis, DUTH at SemEval-2026 task 9: Joint multilingual fine-tuning for online polarization detection, in: *Proceedings of the 20th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, 2026.
- [21] G. Arampatzis, A. Arampatzis, DUTH at SemEval-2026 task 3: Multilingual transformer models for dimensional stance prediction across tasks, in: *Proceedings of the 20th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, 2026.
- [22] G. Arampatzis, A. Arampatzis, DUTH at SemEval-2026 task 1: Prompt-based zero-shot large language models for constrained multilingual humor generation, in: *Proceedings of the 20th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, 2026.
- [23] G. Arampatzis, I. Maslaris, A. Arampatzis, Llm-based question generation and retrieval-augmented reporting for news credibility, in: *Proceedings of the Thirty-Fourth Text REtrieval Conference (TREC 2025)*, NIST, 2025.
- [24] G. Arampatzis, V. Perifanis, A. Arampatzis, Justification retrieval with llms, retrieval-augmented generation, and hybrid labels, in: *The Thirty-Fourth Text REtrieval Conference Proceedings (TREC 2025)*, NIST, 2025.
- [25] G. Arampatzis, S. Symeonidis, A. Arampatzis, Precision by design: Rm3 and fusion in product search, in: *Proceedings of the Thirty-Fourth Text REtrieval Conference (TREC 2025)*, NIST, 2025.
- [26] G. Arampatzis, A. Arampatzis, Hybrid sparse-neural fusion for passage retrieval, in: *Proceedings of the Thirty-Fourth Text REtrieval Conference (TREC 2025)*, NIST, 2025.
- [27] G. Arampatzis, K. Safouri, A. Arampatzis, Bridging lexical and neural ranking for topic-oriented retrieval, in: *Proceedings of the Thirty-Fourth Text REtrieval Conference (TREC 2025)*, NIST, 2025.
- [28] P. Vachharajani, et al., Overview of the CLEF JOKER Task 1: Humor-Aware Information Retrieval in English and Hinglish, in: [57], 2026.
- [29] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, *Information Processing & Management* 24 (1988) 513–523.
- [30] T. Solorio, E. Blair, S. Maharjan, S. Bethard, M. Diab, M. Ghoneim, A. Hawwari, F. AlGhamdi, J. Hirschberg, A. Chang, P. Fung, Overview for the first shared task on language identification in code-switched data, in: *Proceedings of the First Workshop on Computational Approaches to Code Switching*, Association for Computational Linguistics, 2014, pp. 62–72.
- [31] U. Barman, A. Das, J. Wagner, J. Foster, Code mixing: A challenge for language identification in the language of social media, in: *Proceedings of the First Workshop on Computational Approaches to Code Switching*, Association for Computational Linguistics, 2014, pp. 13–23.
- [32] T.-Y. Liu, *Learning to Rank for Information Retrieval*, Now Publishers, 2009.
- [33] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with disentangled attention, *arXiv preprint arXiv:2006.03654* (2021).
- [34] 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, in: *Advances in Neural Information Processing Systems*, volume 33, 2020, pp. 5776–5788.
- [35] R. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document ranking with a pretrained sequence-to-sequence model, in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 708–718.
- [36] C. D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
- [37] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, *ACM Transactions on Information Systems* 20 (2002) 422–446.
- [38] E. M. Voorhees, The TREC robust retrieval track, *ACM SIGIR Forum* 39 (2005) 11–20.
- [39] L. Ermakova, et al., Overview of the CLEF 2026 JOKER Task 2: Pun Translation from English to French, in: [57], 2026.
- [40] P. Zabalbeascoa, S. Attardo, Humour translation theories and strategies, in: *Transmedial Perspectives on Humour and Translation: From Page to Screen to Stage*, Routledge, 2024.
- [41] J. Vandaele, Translating literary humour: Aspects of detection and analogy making, in: *Transmedial Perspectives on Humour and Translation: From Page to Screen to Stage*, Routledge, 2024.

- [42] L. Ermakova, A.-G. Bosser, T. Miller, R. Campos, Overview of the CLEF 2025 JOKER task 2: Wordplay translation from english into french, in: Working Notes of CLEF 2025: Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2025.
- [43] J. Tiedemann, S. Thottingal, OPUS-MT: Building open translation services for the world, in: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, 2020, pp. 479–480.
- [44] NLLB Team, No language left behind: Scaling human-centered machine translation, arXiv preprint arXiv:2207.04672 (2022).
- [45] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, N. Goyal, T. Birch, V. Liptchinsky, S. Edunov, E. Grave, M. Auli, A. Joulin, Beyond english-centric multilingual machine translation, *Journal of Machine Learning Research* 22 (2021) 1–48.
- [46] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, W. El Sayed, Mistral 7b, arXiv preprint arXiv:2310.06825 (2023).
- [47] Qwen Team, Qwen2.5 technical report, arXiv preprint arXiv:2412.15115 (2025).
- [48] Meta AI, Meta llama 3 model card, Model card, 2024.
- [49] D. M. Alves, N. M. Guerreiro, J. Pombal, R. Rei, A. F. T. Martins, Tower: An open multilingual large language model for translation-related tasks, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024.
- [50] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 38–45.
- [51] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2022).
- [52] T. Detrmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, arXiv preprint arXiv:2305.14314 (2023).
- [53] L. Ermakova, et al., Overview of the CLEF 2026 JOKER Task 3: Onomastic Wordplay Translation from English to French, in: [57], 2026.
- [54] Z. Xu, S. Escalera, V. Pavlovic, M. Richard, W.-W. Tu, Q. Yao, H. Zhao, I. Guyon, Codabench: Flexible, easy-to-use and reproducible meta-benchmark platform, *Patterns* 3 (2022) 100543.
- [55] I. Kuzmin, et al., Overview of the CLEF 2026 JOKER Task 4: Humor Generation, in: [57], 2026.
- [56] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: International Conference on Learning Representations, 2020.
- [57] E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026: Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.