

One Workflow Does Not Fit All Relations: A System Description for the AKBC 2026 LM-KBC Shared Task

Andreea-Elena Cochintele and Anca Dobrovăț and Sergiu Nisioi

Faculty of Mathematics and Computer Science

University of Bucharest

andreea.cochintele@gmail.com

anca.dobrovat@unibuc.ro

sergiu.nisioi@unibuc.ro

Abstract

This paper presents a system developed for the fifth AKBC Shared Task on Knowledge Base Construction from Language Models (LM-KBC 2026). The task involves predicting sets of object-entity for subject-relation pairs across six relation types, ranging from single-valued relations to relations with potentially hundreds of recipients. All predictions are generated in a closed-book setting (Roberts et al., 2020), using up to 32 billion parameters across all neural components, without resorting to fine-tuning or retrieval augmented techniques. The development process consisted of two stages: validating the technical methodology using the officially specified baseline model (*Qwen3.5-9B*), followed by comparing eleven open-weight models using the same evaluation framework. This process led to the selection of Gemma 3 27B as the final system. The paper presents the main output-parsing errors identified and resolved, together with an evaluation of self-consistency and majority-voting aggregation for each relation type. The final system improved macro-F1 score from 0.361 to 0.531 (a +47% relative improvement) through workflow optimization and model selection.

1 Introduction

Pre-trained language models incorporate significant factual knowledge from their training data, a fact demonstrated by studies using single-answer cloze-questions (Petroni et al., 2019). The LM-KBC competition (LM-KBC Organizers, 2026) extends this line of research by analyzing whether a language model can function as a knowledge base. In this task, a subject-relation pair may have no correct object, a single correct object, or multiple correct objects, depending on the specific relation. The task does not assume a fixed number of correct answers. This distinction is important: for instance, the *awardWonBy* relation can have numerous correct answers, but *personHasCityOfDeath* typically

has a single answer. A system that works well for one type of relation may not work as well for another. Furthermore, certain design choices that benefit one relation may have a negative impact on another.

This is a system description paper that reports engineering findings derived from a shared-task submission. Its main finding is that engineering choices—prompt generation, internal consistency, and aggregation—need to be tailored per relation rather than applied uniformly, since their effect depends on each relation’s answer cardinality. The model comparison, the syntactic analysis taxonomy, and case studies shown in this paper are organized around establishing and illustrating this point.

Our contributions are:

- A controlled comparison of eleven open-weight models (Qwen, Mistral, Phi, Gemma, Llama) under the same evaluation harness.
- Identification of the 27B parameter model as the best available option within the 32B parameter budget of the competition.
- An empirically-derived taxonomy of output-parsing analysis errors in the results that occur when free-text is converted into lists of entities.
- An analysis showing that self-consistency and majority voting, based on the previous theory (Wang et al., 2022; Chen et al., 2023), help certain relations but have a negative impact on others.
- We document one case, on one relation, where an explicit negative instruction led to an increase in the frequency of the forbidden word, which is consistent with the known difficulties that medium-sized, open-weight models encounter regarding contextual negation.

Our code, prompts, and post-processing rules are publicly available.¹

2 Related Work

Previous editions of the LM-KBC competition explored various approaches to constructing knowledge bases from language models. [Li et al. \(2022\)](#) won the first edition (ISWC 2022) with a BERT-based system that combined task-specific pre-training and prompt decomposition to progressively generate objects, achieving an F1 score of 55.0%. In the second edition, [Nayak and Timmapathini \(2023\)](#) fine-tuned the Llama 2 model using LoRA and context augmented by Wikipedia retrieval, achieving a mean F1 score of 0.6185 across 21 relations simultaneously. [Zhang et al. \(2023\)](#) reported an F1 score of 0.7007 using GPT-4 in a "closed prompting" scenario, while noting significant performance disparities across relations (e.g. 1.0 for *PersonHasNobelPrize* versus 0.328 for *PersonHasEmployer*), a pattern consistent with the cardinality-dependent behavior we analyze in [subsection 6.2](#).

This paper is different from these prior systems in two ways. First, none of the systems mentioned combines an explicit "closed-book" constraint with a limit on the number of parameters. Based on competition rules, we restrict our comparison to open-weight models under 32 billion parameters, without fine-tuning or external retrieval. Second, we run a structured, per-technique ablation across eleven open-weight models and describe how prompting techniques, self-consistency, and majority-vote aggregation interact with each relation’s answer cardinality.

3 Task and Data

3.1 Task Definition

Given a subject entity and a relation, the task consists of predicting the complete and correct set of object entities, a set that may be empty, contain a single element, or comprise multiple elements. Predictions are evaluated as character strings: a predicted string is normalized and compared against the reference label (gold label) and its known aliases using a maximum-cardinality matching method; thus, the score is independent of the prediction order and each reference entity can be

¹<https://github.com/andreea-cochintele/lmkbc2026-submission>

Relation	Train	Validation	Test
countryLandBordersCountry	67	68	67
personHasCityOfDeath	100	100	100
hasCapacity	100	100	100
awardWonBy	10	10	10
companyTradesAtStockExchange	100	100	100
hasArea	100	100	100

Table 1: Dataset statistics from the official task documentation.

assigned at most one prediction. For the two numerical relations, a prediction is considered correct if it falls within a relative tolerance margin of 5% of the reference value.

3.2 Relations and Dataset Statistics

The dataset comprises six relations (see [Table 1](#)), among which three allow for empty sets, two are numerical, and the *awardWonBy* relation is documented as sometimes having hundreds of valid recipients for a small number of subjects. However, with only 10 subjects (an order of magnitude fewer than for the other relations), conclusions regarding this relation should be interpreted as purely illustrative.

3.3 Evaluation

Predictions are evaluated as normalized strings linked via aliases (see the taxonomy for cleaning the results in [Table 7](#).)

The scorer identifies true positives in the bipartite match between predicted and gold entities, so the score is independent of the prediction order. Therefore, predicting the same entity using two surface forms reduces precision because it inflates the predictions without adding any true positives.

3.4 Competition Constraints

The competition imposed three rules that constrained the design of this system. First, the "closed-book" restriction prohibits web searches, RAG, or any consultation of factual information. Second, no additional training or fine-tuning is allowed. Third, the total number of parameters across all neural components must not exceed 32 billion. The largest model we evaluated, Gemma 3 27B, falls within this limit. Non-neural post-processing: normalization, deduplication, rule-based filtering, and vote aggregation are permitted and utilized.

4 System Description

4.1 Preliminary Ablation

Before adopting any model modifications, we checked the general engineering approach, including relation-specific prompts, per-relation token allocation, list cleaning, single-answer constraints, and majority voting on the officially designated model: *Qwen3.5-9B*, part of the Qwen family of models (Qwen Team, 2025). Each technique was implemented as a toggleable flag, tested first individually, then cumulatively, on the validation set (complete step-by-step results in Appendix C). Unless noted, all F1 values in this article are single run estimates, without testing for statistical significance; section 5 shows the only variance we measured (± 0.005 macro-F1); *awardWonBy* figures reflect only 10 validation subjects.

Relation-specific prompts generated the greatest individual gain (0.361 to 0.415 in isolation), later confirmed on the Gemma 3 27B model. Per-relation token allocation, list cleaning, and single-answer constraints on the *personHasCityOfDeath* relation were adopted. Formatting results with one item per line was rejected. Instead of reducing the generated volume, it encouraged a higher number of output lines for relations with multiple answers.

Even if majority voting on the *awardWonBy* relation doubled precision (0.251 to 0.559) and halved the average number of predictions per subject (28.2 to 12.0), macro-F1 showed an insignificant change (0.082 $\rightarrow$ 0.083); therefore, we treat this as *an undetectable macro-F1 effect* rather than a null result. Re-evaluated on Gemma 3 27B, the same vote raised F1 from 0.147 to 0.161, attributable to the larger model’s broader coverage of award winners, not the voting mechanism itself.

At this stage, prompting and post-processing alone brought the *Qwen3.5-9B* model to a macro-F1 score of 0.408, representing a relative improvement of 30% compared to the organizers’ reported baseline.

4.2 Model Selection

We evaluated eleven open-weight models using the same testing framework: Qwen3.5-9B, Qwen2.5-7B-Instruct, Qwen3-8B, Qwen2.5-14B-Instruct, Qwen3-14B, Mistral-Nemo-Instruct-2407 (12B) and phi-4 (14 B), Gemma 3 (4B/12B/27B) and Llama-3.1-8B. The 14B-class models and the Mistral/phi models were 4-bit quantized with bitsandbytes library (Dettmers et al., 2023) to fit a 16 GB

T4; Qwen3.5-9B, Qwen2.5-7B, and Qwen3-8B ran without quantization, while Gemma 3 and Llama ran through Ollama (GGUF). Full per-model scores are reported in Appendix B.

Two patterns emerged, though the unquantized and quantized models are not clean-cut (Appendix B): Qwen3.5-9B (0.408) and Qwen2.5-7B-Instruct (0.332), both unquantized, outperformed all the bitsandbytes-quantized models tested, but the third unquantized model, Qwen3-8B (0.256), did not (scoring lower than the Mistral-Nemo-Instruct-2407 model (0.321, 4 bits)). Quantization is therefore associated with lower scores here, but it is not sufficient, on its own, to explain the rankings. Second, within the Gemma 3 family, macro-F1 scaled with size (0.306 / 0.462 / 0.510 for 4B/12B/27B), and Gemma 3 27B outperformed all other models, despite being quantized using GGUF, which suggests that GGUF quantization and bits-and-bytes quantization are not interchangeable in this case, although we cannot isolate the quantization method from other confounds (the base model, the tuning recipe). We selected Gemma 3 27B as the final system and chose aggregation at the relation level (subsection 4.4) then raised the macro-F1 to 0.531.

4.3 Prompting

We use a fixed 5-shot prompt per relation, using examples drawn from the training set via a random number generator with a relation-specific seed, formatted as question-answer pairs, with a comma-separated list of answers (or the value "None"). This paper adds a relation-specific clarifying hint to both the query and each example in the "few-shot" set (Table 3). An initial run of the replicated baseline model exhausted its entire 64-token budget trying to determine the real-world entity being referenced, without ever generating a number. This explains the very low scores obtained by the baseline model for the *hasArea* and *hasCapacity* relations. The "one object per line" output format, tested first globally and then on *awardWonBy* alone, did not generate improvements (0.415 vs. 0.423) and was abandoned in favor of handling hedging and multi-entity lines at the parsing stage.

4.4 Output Parsing, Cleaning, Aggregation

Model output is free-form text, not structured data, and in early stages most errors traced back to parsing rather than to the model’s knowledge. Table 7 lists the seven recurring failure types we identified.

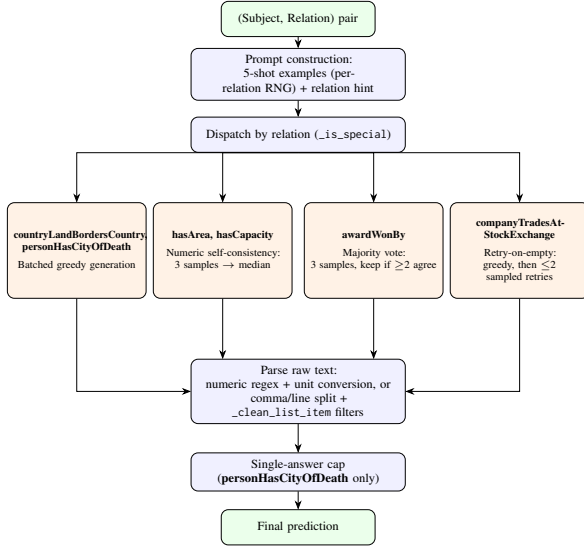


Figure 1: Per-item prediction pipeline. Each of the four branches applies to exactly the relations named on it; all other relations fall through to plain batched greedy generation.

Eliminating one type of failure altered the nature of the next.

Figure 1 illustrates the processing workflow for each item. In addition to "greedy" generation (immediately selecting the most probable option), which is the default setting, we implemented four aggregation strategies (Table 6), each of which was retained or strictly cancelled based on the measured "before/after" macro-F1 score values.

Majority voting for *awardWonBy* keeps a candidate only if it recurs, after normalization, in at least a fixed fraction of sampled generations, measurably improving precision every iteration. Numeric self-consistency (median of several samples) produced a clear improvement on *hasArea* (+0.070 F1) and an inconclusive change on *hasCapacity* (−0.010, within noise). Retry-on-empty helped *companyTradesAtStockExchange* but hurt *personHasCityOfDeath*, where an empty answer more often reflects a genuinely correct empty label than under-confidence. Full per-relation kept/discarded decisions are listed in Appendix D.

4.5 A Case Study in Prompt-Level Negation

In one instance, the *countryLandBordersCountry* relation suffered from the excessive generation of surface variants (translations, transliterations, outdated historical state names). Our initial solution (an explicit instruction such as *do not write 'West Germany' or 'East Germany', instead write 'Germany'*) made this measurably worse: predicted ob-

Relation	Validation		Test	
	Base	Ours	Base	Ours
awardWonBy	0.055	0.168	0.151	0.259
companyTradesAtStockExchange	0.539	0.638	0.273	0.646
countryLandBordersCountry	0.784	0.975	0.732	0.921
hasArea	0.210	0.570	0.340	0.530
hasCapacity	0.090	0.210	0.122	0.112
personHasCityOfDeath	0.350	0.440	0.170	0.320
All	0.361	0.531	0.296	0.473

Table 2: Macro-F1 by relation, reproduced baseline vs. our system, on validation and test.

jects per subject rose from 1.691 to 4.441 in the very next run, and the forbidden strings appeared more than before. This isolated case is consistent with the known weaknesses of medium-sized open-weight models concerning negation handling: mentioning a token can increase its relevance (Kassner and Schütze, 2020). We reverted to a neutral prompt and moved the filtering out of the prompt and into the parsing stage: surface variants and explanatory text fragments are removed by the rules in Table 7. In the final system, *countryLandBordersCountry* is handled by batched greedy generation with no multi-sample aggregation (Figure 1). We do not claim that this generalizes beyond this single relation and phrasing. We report it because it was significant and contrary to the intent of the prompt.

5 Setup

The final system is a subclass of the `AbstractModel` interface associated with the shared task. The documented baseline configuration specifies the Qwen3.5-9B model, a 64 token limit, 5-shot prompting (Brown et al., 2020), and no output format instructions. We reconstructed a faithful, minimal implementation without any pipeline enhancements, so that the comparison in Table 2 isolates our technical solution. We then ran a full validation set (478 pairs) using the final configuration—including multi-sample aggregation for four of the six relations.

Table 2 compares the reconstructed baseline model (Qwen3.5-9B) with the final system (Gemma 3 27B, full workflow) on the validation set and the hidden test set. Our system outperforms the baseline in both cases: +47% macro-F1 on the validation set (0.361 → 0.531) and +59.7% on the test set (0.296 → 0.473), indicating that the improvement generalizes beyond the training set. The biggest absolute gains are observed for *hasArea* (+0.360) and *countryLandBordersCountry*

(+0.191), while the smallest is for *personHasCityOfDeath* (subsection 6.2). Five of the six relations improve on the test set, mirroring the results from validation; the exception is *hasCapacity* (0.112 vs. 0.122), consistent with the noise-level effect of numerical self-consistency already mentioned for this relation (Table 6, -0.010 F1). As in the rest of the paper, these are point estimates obtained from a single run, without testing for significance.

Reproducibility

Three out of the seven workflow modifications evaluated were reverted after a negative "before/after" result (the *retry-on-empty* and *self-consistency* mechanisms for the *personHasCityOfDeath* relation, voting based on 5 samples and a threshold of 3 for the *awardWonBy* relation), highlighting that a technique beneficial for one relation may be detrimental to another with different cardinality characteristics. With deterministic seeds, repeated runs vary by at most ± 0.005 macro-F1 for the two relations employing sampling-based aggregation; relations relying exclusively on *greedy* decoding reproduce identical results in every run.

6 Analysis

6.1 Long-Tail Knowledge

Well-documented entities (e.g., the area of a well-known country) were reproduced near-exactly by every model, including the smallest. "Long-tail" entities (e.g. a small Croatian island) resulted in large errors, presented with a high degree of certainty, across all models, including the final system (Kandpal et al., 2023). In one case, the estimated area of a country was exactly ten times the correct value, suggesting a specific figure that was memorized incorrectly yet confidently, rather than random noise.

Our final system measurably reduced the long-tail error for *hasArea*, where F1 increased from 0.210 (reproduced baseline, Table 2) to 0.570. The largest part of this gain comes from the relation-specific hint, which eliminated the token-budget failure described in subsection 4.3; numerical self-consistency added a +0.070 F1 improvement over that configuration. Neither these methods nor their combination eliminates errors for extremely rare entities. Since the closed-book rule prohibits consulting external sources, this limitation is intrinsic to the task as defined, rather than a flaw in the workflow employed. We present this aspect as a qualita-

tive pattern derived from error analysis, rather than as a quantified statement regarding rare entities in general.

6.2 Self-Consistency Depends on Answer Cardinality

One factor is consistent with all five aggregation experiments we ran: either the independent samples converge toward a correct answer that greedy decoding misses, or greedy decoding already provides an answer close to the best individual answer, in which case sampling largely adds noise. The clearest evidence comes from the two largest relations (100 subjects each, and thus less prone to the noise specific to small samples than *awardWonBy*): for *hasArea*, greedy decoding of long-tail entities often yields an incorrect answer, so the median of the samples is helpful; for *personHasCityOfDeath*, greedy decoding is usually already the best single answer for this specific, single-value fact, and extra samples can overturn a correct greedy answer if two of them happen to agree on the same incorrect city. The relation *awardWonBy* (10 subjects) shows the same qualitative pattern — majority voting filters out isolated, hallucinated names while retaining those consistently reproduced.

6.3 Negation, Revisited

The prompt-level failure in subsection 4.5 is consistent with previous findings that pre-trained language models handle contextual negation unreliably (Kassner and Schütze, 2020). We recommend validating any instruction of the form "do not write X" through a before-and-after comparison.

6.4 Incompleteness of ground-truth data

Two of the ten *awardWonBy* validation subjects pertain to honors such as *doctor honoris causa* titles, supported by a documented but incomplete reference set, which limits accuracy, regardless of the system's quality. We initially treated every high-frequency prediction for this relation as a probable error; a re-examination of the task documentation revealed that many of these were correct entries but underrepresented in the reference set.

6.5 Qualitative Case Studies

Successive analysis corrections applied to the same subject revealed, one by one, three types of errors: misspelled names, incomplete disambiguation suffixes, and conversational preambles—with each correction revealing the next error. A more

systematic error arose in the case of Russia’s borders: our predicted list, identical across all three self-consistency samples, included only six of Russia’s fourteen actual neighbors, plus seven non-neighboring countries, omitting two of the most important ones. Since the error was identical each time, majority voting could not correct it. The model maintained a false association with confidence. Self-consistency eliminates idiosyncratic noise, but does not eliminate a systematic error that is reproduced identically in every run.

6.6 Practical Trade-offs

Larger models generally achieve higher scores (Appendix B): Gemma 3 27B, quantized by GGUF, outperformed several unquantized models under 14B, indicating that the raw scale is not the only influencing factor within a fixed parameter budget. Self-consistency and majority voting involve additional inference calls (3 per element) for gains that depend on relations and are sometimes zero (subsection 6.2). Furthermore, majority voting trades recall for precision: it filters out isolated hallucinations, but a correct answer generated inconsistently across samples may be eliminated by the vote along with the noise.

7 Conclusions

We described a system for the AKBC 2026 LM-KBC shared task, developed using eleven open-weight models. A workflow designed for relation-level aggregation and an ablation-validated decision process produced gains beyond model selection alone (macro-F1 rising from 0.361 to 0.531 relative to a reproduced baseline). We also documented two case studies where explicit negative prompting and generalized self-consistency did not transfer well across relation cardinalities.

Acknowledgments

This research is supported by the project “Romanian Hub for Artificial Intelligence - HRIA”, Smart Growth, Digitization and Financial Instruments Program, 2021-2027, MySMIS no. 351416.

Limitations

- We did not perform statistical significance tests, calculate confidence intervals, or estimate variance for most of the comparisons presented; nearly all results in this paper are point estimates obtained from a single run. In

cases where we separately measured the variation from one run to another (± 0.005 macro-F1 for sampling-based relations, section 5), we use this as an informal noise threshold, but we did not measure the variance for each configuration compared in tables 5-2, so some of the reported differences (particularly those below ~ 0.01 F1) may not represent reliable effects.

- *awardWonBy* has only 10 subjects per division, an order of magnitude fewer than the other five relations. Many of our conclusions regarding majority voting and self-consistency rely heavily on this relation.
- The “closed-book” constraint excludes augmented generation through retrieval (the most direct solution for sparse numerical data, see subsection 6.1), so we cannot test it in accordance with the task rules.
- Residual variation from one run to another (± 0.005 macro-F1) persists even with deterministic seeds, due to the non-deterministic order of floating-point number reduction in multi-threaded GPU execution.
- Available computing resources dictated the process: all experiments were run on Kaggle’s free tier (weekly limit of 30 hours, two 16-GB T4 GPUs).

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.
- Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818.

Tianyi Li, Wenyu Huang, Nikos Papasrantopoulos, Pavlos Vougiouklis, and Jeff Z. Pan. 2022. Task-specific pre-training and prompt decomposition for knowledge graph population with language models. In *Proceedings of the LM-KBC Challenge at ISWC 2022*.

LM-KBC Organizers. 2026. Akbc shared task 2026: Knowledge base construction from language models (5th edition). <https://lm-kbc.github.io/challenge2026/>. Accessed 2026.

Amol Nayak and Hari Prasad Timmapathini. 2023. Llm2kb: Constructing knowledge bases using instruction tuned context aware large language models. In *Proceedings of the LM-KBC Challenge at ISWC 2023*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Bohui Zhang, Ioannis Reklou, Nitisha Jain, Albert Meroño Peñuela, and Elena Simperl. 2023. Using large language models for knowledge engineering (llmke): A case study on wikidata.

A Hints per Specific Relation

Table 3 lists the exact clarifying hint appended to every query and few-shot example for each relation, as discussed in subsection 4.3.

Relation	Appended hint
hasArea	“Answer with a single number in square kilometers, no units, no commas.”
awardWonBy	“Name the recipient (person, group, or organization) who WON the award, not the work/project that was awarded. List as many confirmed winners as you know.”
countryLand-BordersCountry	“Only current, internationally recognised land borders. Exclude maritime-only neighbors. If none, answer None.”
personHasCityOf-Death	“Name only the city (not the country or region). If still alive or unknown, answer None.”
companyTrades-AtStockExchange	“Name the stock exchange(s). If not publicly traded, answer None.”
hasCapacity	“Answer with a single integer number of people, no units, no commas.”

Table 3: Relation-specific prompt hints appended to every query and few-shot example.

B Full Model Comparison

Table 4 reports validation macro-F1 for all eleven models evaluated in subsection 4.2, plus the quantization method and parameter count of each.

Model	Params	Quant.	Macro-F1 (val.)
Qwen3.5-9B	9B	none	0.408
Qwen2.5-7B-Instruct	7B	none	0.332
Qwen3-8B	8B	none	0.256
Qwen2.5-14B-Instruct	14B	4-bit	0.291
Qwen3-14B	14B	4-bit	0.188
Mistral-Nemo-Instruct-2407	12B	4-bit	0.321
phi-4	14B	4-bit	0.234
Gemma 3 4B	4B	GGUF	0.306
Gemma 3 12B	12B	GGUF	0.462
Gemma 3 27B	27B	GGUF	0.510
Llama-3.1-8B	8B	GGUF	0.190

Table 4: Macro-F1 by model on the validation set.

C Full Preliminary Ablation

Table 5 shows every step of the incremental ablation on Qwen3.5-9B described in subsection 4.1, in the order each flag was tested.

Configuration	F1
Official external baseline (reference)	0.313
Own vanilla run, no improvements	0.382
All flags off (re-confirmation)	0.361
+ relation hints only	0.415
+ per-relation token budget only	0.377
+ list cleaning	0.406
All 4 base flags + fixed seed	0.423
+ one-per-line format (global)	0.415
+ isolated per-relation RNG + majority vote	0.408
Final: isolated RNG, majority vote off	0.408

Table 5: Ablation of engineering techniques on the reference model (*Qwen3.5-9B*), testing order.

D Full Aggregation Strategy Results

Table 6 lists, for every relation where a multi-sample aggregation strategy was tried (subsection 4.4), whether that strategy was ultimately kept in the final system.

Relation	Strategy	Kept?
awardWonBy	Majority vote, 3 samples, thr. 2	Yes
awardWonBy	5 samples, thr. 3	No
companyTradesAt-StockExchange	Retry-on-empty	Yes
personHasCityOfDeath	Retry-on-empty	No
personHasCityOfDeath	Self-consistency, single answer	No
hasArea	Self-consistency, median of 3	Yes
hasCapacity	Self-consistency, median of 3	Yes(Neutral)

Table 6: Aggregation strategies tried per relation.

Failure mode	Fix
Meta-commentary	Reject hedging lines and long lowercase-terminated sentences
Label-like residue	Reject items ending in a colon or with a comment marker
Punctuation residue	Combined edge-stripping in one pass
Unclosed suffix	Truncate at an unmatched opening parenthesis
Self-reference	Drop candidates equal to the subject entity
Conversational preamble	Exact-match filler-word and list-intro filters
Numeric drift	Strip thousand-separators; unit conversion for hasArea

Table 7: Taxonomy of output-parsing failure modes and fixes.

E Output-Parsing Failure Taxonomy

Table 7 lists the seven recurring failure types identified during development (subsection 4.4) and the rule-based fix applied to each.