LLM-INSTRUCT at UZH Shared Task 2026: Constraint-Aware Retrieval and Selective Debate for Paragraph-Level Argument Mining

Phuong Huu Vu Tran[1,*,], Long Minh Vo[2,*,**], Son Nguyen Minh Le[1], Hoang Van[2,3]**
[1]Vietnamese-German University, Vietnam
[2]RMIT University Vietnam, Vietnam
[3]VANGIA INNOVATIONS, Vietnam
[*]These authors contributed equally.
[**]Corresponding authors: 10425032@student.vgu.edu.vn, s4215945@rmit.edu.vn

Abstract

We present LLM-INSTRUCT, the winning system for the UZH Shared Task at ArgMining 2026 on paragraph-level argument mining in UN and UNESCO resolutions. The task requires paragraph-type classification, prediction of a subset of 141 official tags, and directed relation prediction under a strict JSON schema setting using only open-weight models up to 8B parameters. We frame the task as constrained structured prediction. The system first narrows the candidate tag space with metadata-aware dense retrieval, then applies constrained decoding with per-dimension caps, escalates only uncertain cases to a three-agent debate branch, and finally validates the output schema. On the official leaderboard, LLM-INSTRUCT ranked **1st overall**, with **1st in F1** and **5th in LLM-as-a-Judge**. During development, our configuration search further improved Task 1b Micro-F1 from 35.83% to 40.08% while keeping the internal Task 2 score at 4.421. The main lesson is simple: reducing the decision space before generation improves both accuracy and submission robustness. Our code and supporting scripts are publicly available at: https://github.com/LLM-Instruct-at-UZH-Shared-Task-2026/Method

1 Introduction

The UZH Shared Task at ArgMining 2026 asks participants to reconstruct argumentative structure in highly formal institutional texts. For each paragraph, a system must determine whether it belongs to the preambular or operative part of a document, assign a subset of 141 predefined thematic tags, and recover directed argumentative relations to other paragraphs in the same document. The task is difficult because the texts are long, the label inventory is closed and structured, and the submission format is strict: non-conforming JSON is not evaluated. In other words, the benchmark requires both semantic plausibility and schema-valid output.

This benchmark is better viewed as a constrained prediction task than as open-ended text generation. The model must reason over long institutional paragraphs, but it must also stay within a fixed label inventory, preserve paragraph indices, and produce schema-valid predictions. We therefore narrow the admissible output space before final generation. Candidate retrieval narrows the tag space; organizer-provided metadata is reused in retrieval and in per-dimension caps; decoding is projected back to the retrieved set; and final validation enforces the submission schema.

This paper makes three contributions. First, it describes the end-to-end pipeline of the first-ranked LLM-INSTRUCT submission under the official shared-task constraints. Second, it identifies the key design choice behind the result: constraining admissible tags before generation instead of asking the model to search the full 141-tag inventory. Third, it reports the development trajectory that exposed the main failure mode of early runs, namely cross-dimension over-prediction.

2 Related Work

Our system combines ideas from constrained decoding, dense retrieval, debate-style reasoning, and argument-mining pipelines. In particular, it is closest to settings that restrict admissible outputs before or during decoding (Geng et al., 2023; Liu et al., 2022), while also drawing on dense retrieval (Karpukhin et al., 2020), debate-style control (Du et al., 2024), and prior argument-mining work (Lawrence and Reed, 2020; Stab and Gurevych, 2017). The difference in this shared task is that semantic plausibility alone is not enough: predictions must also satisfy a strict output schema. For broader context, our pipeline also contrasts with end-to-end and text-to-text approaches to argument mining, including structured prediction as generation (Paolini et al., 2021), end-to-end universal argument mining (Cao, 2023), fine-tuned LLM pipelines for AM (Cabessa et al., 2025), and recent LLM work on relation-based argument mining (Gorur et al., 2025).

3 Proposed Method

Figure 1 summarizes the pipeline. The final leaderboard run has three prediction stages followed by a submission-safety layer. We design the pipeline to satisfy these strict task constraints while keeping generation tightly controlled. We describe its components next.

3.1 Type stage

Each paragraph is first classified as *preambular* or *operative*. In development, we found that a deterministic

Proceedings of the 13th Workshop on Argument Mining and Reasoning, pages 119–124

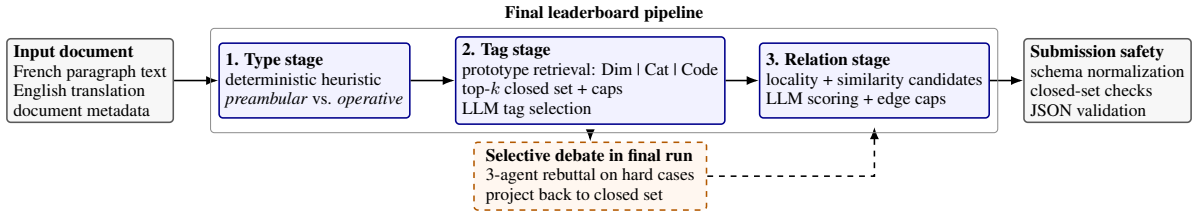

Figure 1: System pipeline. The winning configuration reduces the decision space before final generation: dense retrieval creates an admissible tag set, organizer-provided metadata is encoded in the tag prototypes and reused for per-dimension caps, and high-uncertainty cases are escalated to a three-agent debate branch whose output is projected back onto the same admissible label set before final schema validation.

Table 1: Representative triggers used by the type heuristic.

Rule type	Example opener	Predicted type
French preambular cue	*Considérant ...*, *Reconnaissant ...*, *Rappelant ...*	preambular
French operative cue	*Demande ...*, *Souligne ...*, *Attire ...*, *Proclame ...*	operative
English compound cue	*Calls upon ...*, *Takes note ...*	operative
Numbering pattern	`1.`, `2)`, `(3)` at paragraph start	operative
Fallback	no cue matched	preambular

heuristic was more stable than a pure LLM-based alternative, so the final configuration keeps this step rule-based. Specifically, we classify paragraphs as operative when they match explicit numbered clause patterns or operative cue phrases, and otherwise default to preambular. Table 1 gives representative triggers used by the heuristic. This reduces variance early in the pipeline and avoids propagating unstable type decisions to later stages.

3.2 Metadata-aware tag retrieval and decoding

The tag stage is the core of the system. We read the official tag CSV and keep rows whose CODE field is neither empty nor NA. For each remaining tag t, let d_t, c_t, and y_t denote its released dimension, category, and CODE fields. We textualize the tag prototype as

$$p_t = d_t \,\|\, c_t \,\|\, y_t,$$

where $\|$ denotes string concatenation. We then embed both p_t and the paragraph text with intfloat/e5-base-v2 (Wang et al., 2022), and cosine similarity is then used for dense top-k retrieval (Wang et al., 2022; Karpukhin et al., 2020).

This design makes the retrieval process metadata-aware: dimension and category information is present inside the prototype text before final tag generation. The LLM never predicts over the full 141-tag inventory. Instead, it selects from the retrieved closed set. In the final run, we apply a global cap of 5 tags per paragraph and a per-dimension cap of 2 tags. These fixed controls limit redundancy within a single semantic region and suppress cross-dimension over-prediction. Finally, any decoded label outside the retrieved candidate set is rejected. Together, these controls address the main failure mode observed during development, namely cross-dimension over-prediction.

In addition to retrieving candidate tags, the final tag prompt includes up to three retrieved in-context examples from the training release when their embedding similarity exceeds 0.70. These examples are used only as contextual evidence for how similar paragraphs are tagged; they do not expand the official tag inventory, and the final prediction is still projected back to the retrieved closed set.

3.3 Selective debate branch

We also implement a hard-case route with three open-weight 8B agents-Qwen3-8B (Yang et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Ministral-8B-Instruct (Mistral AI, 2024)-inspired by debate-style reasoning (Du et al., 2024).

Under this setup, a debate is triggered only for uncertain paragraphs, namely cases where the top-1 tag margin is low under a development-tuned routing rule, the heuristic and generator disagree on type, or the retrieved candidates show strong overlap across competing tag dimensions. Each agent proposes a type/tag hypothesis, the agents exchange one focused rebuttal round, and a constrained selector accepts only labels that remain inside the retrieved candidate set while reapplying the same per-dimension caps. Put differently, debate never expands the admissible label set; it only helps resolve hard cases within the same retrieved closed set.

3.4 Relation stage

Relation prediction starts from sparse candidate generation rather than exhaustive all-pairs scoring. For clarity, Table 2 summarizes the four official relation labels used in Subtask 2. For each source paragraph, we form a candidate target set by taking all targets within a locality window of one paragraph and adding the top six embedding-similar targets in the same prediction instance. This keeps nearby discourse links available while still allowing non-adjacent semantic links.

The Qwen3-8B generator then scores each candidate pair using only the four official labels in Table 2 or a no-edge option. We keep pairs whose confidence is at least 0.40 and cap each source paragraph at five outgoing edges, ranked by confidence. These filters

Table 2: Official relation labels in Subtask 2.

Label	Informal meaning
supporting	strengthens or justifies another paragraph
complemental	adds compatible information without conflict
modifying	narrows, qualifies, or conditions another paragraph
contradictive	conflicts with or contradicts another paragraph

control the density of the relation graph and prevent relation prediction from becoming an unbounded all-pairs generation problem. Because the held-out test release does not expose gold relation labels, we report relation-output statistics rather than gold precision or recall for this stage.

3.5 Submission safety

The final stage focuses on submission validity. It normalizes schema variants at ingestion, validates required keys before export, enforces the official relation label set, checks paragraph-index consistency, and attempts to repair malformed JSON up to three times when necessary. This component is essential because submissions that violate the required format do not receive an official score.

4 Experiments

4.1 Data and Official Setting

The shared task contains two subtasks. Subtask 1 predicts paragraph type (preambular or operative) and a multi-label subset of 141 official tags. Subtask 2 predicts paragraph-to-paragraph links and labels each directed relation as contradictive, supporting, complemental, or modifying. Official ranking averages an automated F1 metric and an LLM-as-a-Judge score. Because the public leaderboard reports ranks rather than absolute values, we report official rank positions and complement them with internal development metrics.

The organizers release a large *unlabelled* training set drawn from $2,695$ UN resolutions from the UN-RES corpus associated with the SpiritRAG resource (Gao et al., 2025), together with a held-out UNESCO evaluation split. We use the task release as provided by the organizers and do not introduce additional training labels. The texts are provided in French, with English translations provided by the task organizers to support non-French-speaking participants. In our final configuration, Task 1 and Task 2 read the English field whenever it is available and otherwise fall back to the original French paragraph. We prioritize English when available because the organizer-provided translations simplified prompt design, qualitative inspection, and debugging for a non-French-speaking team. The task page also releases a CSV file, evaluation_dimensions_updated.csv, containing the official tag inventory together with dimension and category metadata. Our method uses this file directly in both retrieval and decoding. In the current analyzed artifact, all evaluated instances had English available, so we do not present a separate French-only case study in this version.

4.2 Models and Evaluation Protocol

The final system complies with the task policy of using only open-weight models with at most 8B parameters. The main generator is Qwen3-8B in 4-bit inference mode (Yang et al., 2025). Dense retrieval uses intfloat/e5-base-v2 (Wang et al., 2022). Table 3 reports the deterministic settings of the best performing internal configuration.

We report two complementary views of performance. First, we present the *official leaderboard results* on the test set. Because the organizers have not yet released the exact official scores, we report rank positions rather than absolute scores. Second, we report the *internal development trajectory* that guided system selection and revealed the main failure mode. Following the shared-task setup, our internal evaluation uses Task 1 scores together with a Task 2 score derived from judge-based relation assessment. Code, prompts, and supporting scripts are publicly available at our GitHub repository.

Component	**Setting**
Generator	Qwen3-8B (4-bit)
Reasoning budget	256
Embeddings	intfloat/e5-base-v2
Language path	English if available, else French
Type mode	heuristic
Tag retrieval	cosine top-k, $k = 40$
Tag textualization	Dimensions ǀ Categories ǀ CODE
Tag decoding	threshold 0.33; max 5 tags
Per-dimension cap	max 2 tags per dimension
Relation candidates	window $= 1$, $k = 6$
Relation filtering	threshold 0.40; max 5 edges/source
RAG examples	$k = 3$, minimum cosine score 0.70
Debate	enabled for hard cases
JSON repair	max 3 retries

Table 3: Key settings of the strongest internal configuration and the final leaderboard submission.

4.3 Results

4.3.1 Final Rank on the Official Leaderboard

Table 5 gives the official leaderboard ranks. The main empirical outcome is straightforward: LLM-INSTRUCT ranked **1st overall**. The table also shows the strongest competing teams and our earlier submission, *LLM-Instruct-2*, which corresponds to the earlier Phase2-style configuration.

This pattern is consistent with the design priorities of the system. The gains appear strongest when the benchmark rewards valid structured output. The official rank-1 outcome therefore supports our central claim that schema-aware control layers are useful for paragraph-level argument mining under hard output constraints. It also aligns with the internal trajectory in Table 4, where the earlier submission *LLM-Instruct-2* ranked below the final LLM-INSTRUCT system.

Run	Dominant setting	T1a Acc	T1a F1	T1b P	T1b R	T1b F1	T2 Judge
Phase_0	*Initial baseline from the internal script*	72.19	69.69	**54.92**	26.59	35.83	**4.421**
Phase_1	*Recall-oriented tagging; looser selection increased over-prediction*	85.81	83.21	21.57	30.11	25.13	4.388
Phase_2	*Further recall-oriented tuning; cross-dimension false positives remained high*	85.77	83.16	21.53	30.09	25.10	4.364
Phase_3	*Final constraint-aware run with metadata-aware retrieval, selective debate, per-dimension caps, and closed-set validation*	**86.08**	**83.49**	49.94	**33.47**	**40.08**	**4.421**

Table 4: Internal development trajectory. T1a is paragraph-type classification; T1b is tag assignment. T2 Judge is the internal LLM-as-a-Judge weighted relation score. The main correction from Phase 1/2 to Phase 3 is precision recovery under a similar recall regime, consistent with the over-prediction diagnosis.

Team	F1	Judge	Final
LLM-Instruct	**1**	**5**	**1**
Prompteam	5	1	2
Argchestrators	2	6	3
HybridArguer	4	3	3
LLM-Instruct-2*	7	4	6

Table 5: Official leaderboard ranks from the UZH Shared Task. *LLM-INSTRUCT-2 is our first submission and corresponds to the earlier Phase2-style run.

4.3.2 Development trajectory

Table 4 summarizes the main development phases. We explain the phases explicitly because the most useful lesson from development is not merely that the final run scored higher, but *why* it scored higher.

The pattern is informative. Early recall-oriented configurations raised coarse paragraph-type scores but harmed Task 1b badly. The reason was over-prediction: loose tag selection created many cross-dimension false positives. The final run corrected this failure mode by combining metadata-aware retrieval before generation, retrieved examples in the tag prompt, per-dimension caps during selection, and strict closed-set validation after decoding.

4.3.3 Component diagnostics

We next report a compact component diagnosis, keeping only the results that show clear accuracy or robustness effects. The subset ablations use the same stratified 12-document subset and the same fast decoding setting, so they should be read as relative component evidence rather than absolute final-system scores. Details are in Table 6.

The strongest component effect comes from metadata-aware prototypes: replacing the *Dimension* | *Category* | *CODE* prototype with CODE-only text reduces subset Task 1b F1 by 7.74 points. Retrieved examples are also useful: removing them lowers recall from 22.76% to 16.68% and F1 by 5.17 points. By contrast, closed-set filtering does not materially change subset F1, but it prevents out-of-candidate and invalid raw tags from reaching the final JSON. A full-corpus per-dimension-cap ablation shows a smaller regularization effect: removing the cap changes F1 from 40.26% to 39.84% and increases false positives from 3,529 to 3,575.

Box 4.3.3 shows a representative output.

Variant	P	R	F1	Δ
Subset baseline	37.24	22.76	28.26	–
No RAG examples	37.52	16.68	23.09	-5.17
CODE-only prototype	27.89	16.23	20.52	-7.74
No closed-set filter	37.45	23.08	28.56	+0.30

Table 6: Subset component diagnostics for Task 1b on a stratified 12-document subset under the same fast decoding setting. Removing retrieved examples mainly reduces recall, while replacing metadata-aware prototypes with CODE-only prototypes substantially reduces both precision and F1. Closed-set filtering has little effect on F1 but is retained for schema safety.

Box 4.3.3: Representative model output

Paragraph (English). "While acknowledging that the number of compulsory school years may vary between countries, [the Conference] considers it desirable that the number of actual years of schooling should in no case be less than seven, and notes that this minimum is already exceeded in many countries."
Type. operative
Tags. LAW_REG, ISC_1
Outgoing relations. 2, 4, 7 → complemental.
Interpretation. A legal-regulatory recommendation aligned with nearby policy paragraphs.

We also inspected Phase 3 errors by tag dimension. The largest false-positive counts came from broad dimensions such as Policy theme (623 FP), Teachers (547 FP), Legal frameworks (388 FP), Curriculum (325 FP), and Education level (321 FP). The most frequent false-positive tags were LAW_REG (194 FP), POL_EIE (161 FP), T_OTHER (137 FP), POL_CUR (110 FP), and PEDAG_OTHER (93 FP). This supports the over-prediction diagnosis: broad policy-related dimensions are semantically close to many paragraphs and are therefore easy to over-select under recall-oriented prompting.

Performance also varied strongly by tag frequency. For tags appearing more than 20 times in the internal reference, Phase 3 achieved 57.22% precision, 33.71% recall, and 42.43% F1. For medium-frequency tags with 6–20 references, F1 dropped to 16.61%; for rare tags with at most five references, F1 was only 2.51%. Thus, the constraint-aware design reduced broad over-prediction but did not solve sparse-label recognition.

More concretely, disabling closed-set filtering introduced 8 official tags outside the retrieved candidate set and 5 invalid raw tags in the trace, even though the subset F1 changed only marginally. We therefore retain this step as a reliability guard rather than an accuracy-driven

component.

4.4 Output statistics

Operationally, the final submission JSON is organized by recommendation-level TEXT_ID values. The final artifact contains 89 prediction instances spanning 44 unique source documents, where the source document title is read from METADATA.structure.doc_title. All runtime and output counts reported below are computed at this prediction-instance level.

Because the held-out test release does not expose gold relation labels, we cannot compute candidate recall or per-label precision/recall against gold for the final submission. We therefore report descriptive output statistics directly from the final submission artifact. Across these prediction instances, the final system produced 13,323 directed edges among 132,840 possible within-instance paragraph pairs, for a graph density of 10.03%. All four official relation labels are present: 8,949 *complemental*, 4,199 *supporting*, 160 *modifying*, and 15 *contradictive*. Non-adjacent links were common: 9,566 edges, or 71.80%, spanned more than one paragraph.

4.5 Compute report

The final run was executed on $2\times$NVIDIA GeForce RTX 3090 24GB GPUs and processed the 89 prediction instances in 1:42:03, averaging 68.81 seconds per prediction instance, 2.07 seconds per paragraph, or 34.5 minutes per 1k paragraphs.

5 Discussion

5.1 System Strengths and Advancements

Four implementation choices appear central. First, metadata-aware tag prototypes are important: the CODE-only diagnostic produced the largest observed subset drop. Second, retrieved examples improve recall by supplying paragraph-level usage context without expanding the label inventory. Third, admissibility rules improve reliability by projecting predictions back to the retrieved official candidate set. Fourth, per-dimension caps act as a modest regularizer, reducing false positives on the full corpus even though their absolute F1 effect is smaller than the retrieval-related components.

5.2 Limitations

There are three main limitations in this work. **First**, the relation stage is less mature than the tag stage and remains sensitive to candidate generation and confidence thresholds. **Second**, our component diagnostics are not a full factorial ablation. Several toggles are evaluated on a stratified subset under a faster decoding setting, so they should be interpreted as relative component evidence rather than absolute final-system scores. **Third**, the default path prioritizes English translations when available, so additional cross-lingual analysis on French-only cases would be valuable in future work.

6 Conclusion

We presented a compact, constraint-aware, and submission-safe system for paragraph-level argument mining in UN and UNESCO resolutions. The system combines metadata-aware dense retrieval, constraint-aware decoding, selective debate for high-uncertainty cases, sparse relation prediction, and explicit schema validation. Taken together, these results suggest that under hard output constraints, carefully designed control layers and selectively applied multi-agent reasoning can be as important as stronger generation.

Ethics Statement. Our system is intended for research benchmarking on institutional text, not autonomous legal or policy decision-making. Its design keeps intermediate decisions auditable and makes its constraints explicit.

References

Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. 2025. Argument mining with fine-tuned large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6624–6635, Abu Dhabi, UAE. Association for Computational Linguistics.

Lang Cao. 2023. Autoam: An end-to-end neural model for automatic and universal argument mining. In *Advanced Data Mining and Applications*, pages 517–531, Cham. Springer Nature Switzerland.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yingqiang Gao, Fabian Winiger, Patrick Montjourides, Anastassia Shaitarova, Nianlong Gu, Simon Peng-Keller, and Gerold Schneider. 2025. SpiritRAG: A Q&A System for Religion and Spirituality in the United Nations Archive. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 26–41.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952, Singapore. Association for Computational Linguistics.

Deniz Gorur, Antonio Rago, and Francesca Toni. 2025. Can large language models perform relation-based argument mining? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8518–8534, Abu Dhabi, UAE. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. Autoregressive structured prediction with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 993–1005, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mistral AI. 2024. Model card for ministral-8b-instruct-2410. Accessed 2026-04-16.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *Preprint*, arXiv:2101.05779.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.