Pointwise and Pairwise LLM-Reranker for Job-Title to ESCO Skill Retrieval over GIST-Fine-Tuned JobBERT

NightSun at CLEF 2026

Yeyang Liu[1,*,†], Zhaohang Niu[2,†], Chunyu Li[3,†] and Zhaoyan Fan[4]

[1]Department of Informatics, University of Zurich, Zürich, Switzerland
[2]Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur, Malaysia
[3]Individual Researcher
[4]Department of Computational Linguistics, University of Zurich, Zürich, Switzerland

Abstract

Matching free-text job titles to a structured skill taxonomy is hard: queries are short, lexically distant from skill records, and lie among thousands of near-synonym aliases. We address TalentCLEF 2026 Task B with a four-stage pipeline that confines fine-tuning to a single bi-encoder stage and otherwise relies on zero-shot inference. A GIST-fine-tuned JobBERT bi-encoder ranker produces an initial ranking. Two-sided test-time augmentation max-pools cosine similarities over alias views on the document side and HyDE views on the query side. A two-stage LLM-reranker cascade then refines the top of the ranking with pointwise scoring and a pairwise tournament. The system reaches **0.7913 graded nDCG** on the official test set, placing second on the Task B leaderboard.

Keywords

information retrieval, job-skill matching, ESCO, bi-encoder, LLM-reranker, TalentCLEF

1. Introduction

Candidate matching, career-path prediction, and workforce analytics in human capital management all depend on linking free-text job titles to a structured skill taxonomy [1]. TalentCLEF 2026 [2], Task B [3] formalises this as a retrieval problem: given a free-text job-title query, systems rank the full ESCO skill corpus by graded relevance, with graded nDCG [4] as the primary metric. The task is hard because queries are short and lexically distant from skill records, and the corpus contains thousands of near-synonym aliases that the ranker must disambiguate.

We present a four-stage pipeline that confines fine-tuning to a single bi-encoder stage and reranks the top of the ranking with zero-shot LLM inference. The system reaches **0.7913** graded nDCG on the official test set, ranking **second on the Task B leaderboard**.

Contributions. **JobBERT ranker with GIST fine-tuning.** We adopt JobBERT-v2 as our base encoder and fine-tune it with the GIST loss to mitigate false negatives in contrastive training. The resulting ranker outperforms larger multilingual encoders both zero-shot and after task fine-tuning. **Two-sided test-time augmentation.** We max-pool cosine similarities over multiple views on both sides of the bi-encoder, enriching the document representation through alias-explode and the query representation through multi-style HyDE. The procedure delivers the largest single gain in the pipeline and requires no model change. **Two-stage LLM-reranker cascade.** We apply Qwen 2.5-7B-Instruct-AWQ zero-shot as a reranker in two complementary modes, pointwise scoring and pairwise tournament

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany
*Corresponding author.
†These authors contributed equally to this work.
yeyang.liu@uzh.ch (Y. Liu); niuzhaohang@gmail.com (Z. Niu); li.chunyu0412@gmail.com (C. Li); zhaoyan.fan@uzh.ch (Z. Fan)
https://cv.yeyang.top/ (Y. Liu); https://eeyorelee.github.io/ (C. Li)
0009-0002-3353-3236 (Y. Liu); 0009-0002-9813-7955 (Z. Niu); 0009-0000-3036-0275 (C. Li); 0009-0008-5792-3152 (Z. Fan)

Table 1
Dataset statistics for TalentCLEF 2026 Task B.

	Training	Validation	Test
Job-title queries	3,039	304	1,139
Job–skill pairs / judgements	114,698	56,418	blind
ESCO skills (corpus)	13,939	9,052	5,358

reranking, which together close most of the gap to the bi-encoder-only baseline. **Reproducibility.** The full pipeline runs end-to-end on a single consumer GPU.

2. Related Work

Job–skill retrieval and ESCO. The ESCO taxonomy [5] provides ∼14 k structured skill records widely used as a canonical target for HR retrieval and tagging. TalentCLEF 2025 [6] showed that task-pretrained bi-encoders remain competitive with larger decoder-style models on this corpus.

JobBERT. JobBERT-v2 [7, 8] is a 110 M MPNet encoder with an asymmetric Tanh projection and a Router head, pretrained by TechWolf on a proprietary job-skill corpus. Its task-specific pretraining consistently outperforms larger multilingual backbones on skill retrieval (Section 4.1).

Bi-encoder fine-tuning and the GIST loss. GIST (Guided In-sample selection of Training negatives) [9] extends multiple-negatives ranking loss [10] with a frozen guide model that filters in-batch false negatives, yielding cleaner contrastive signal without hard-negative mining.

Hypothetical document expansion. HyDE [11] bridges the lexical gap between short queries and longer documents by prompting an LLM to generate a hypothetical relevant document, then encoding it with the existing dense retriever.

LLM-reranker. RankGPT [12] demonstrated that LLMs can perform zero-shot pointwise relevance scoring without task-specific fine-tuning, while RankZephyr [13] extended this to listwise comparisons.

3. Task and Data

Task B requires systems to retrieve and rank all relevant ESCO skills for a given free-text job-title query [3]. Relevance is graded at two levels: grade 2 (*core*) for skills required to perform the job regardless of work context, and grade 1 (*contextual*) for skills that depend on the specific organisation or industry. For example:

> *Query:* "data science graduate intern"
> *Grade 2 (core): algorithms, statistics, computer programming*
> *Grade 1 (contextual): artificial neural networks, analyse scientific data*

The primary metric is graded nDCG [4, 6]. Training pairs come from ESCO's occupation–skill mappings; validation and test judgements are provided by the organisers (Table 1).

Each ESCO skill has ∼7 aliases of 3–5 words and a ∼22-word description [5]. Of the 304 validation queries, 96.1 % have no exact title match in training, requiring semantic generalisation. Each validation query has a median of 181 relevant skills, 53 core and 128 contextual. A further 34.4 % of corpus skills appear as both core and contextual across different queries, reflecting context-dependent graded relevance.

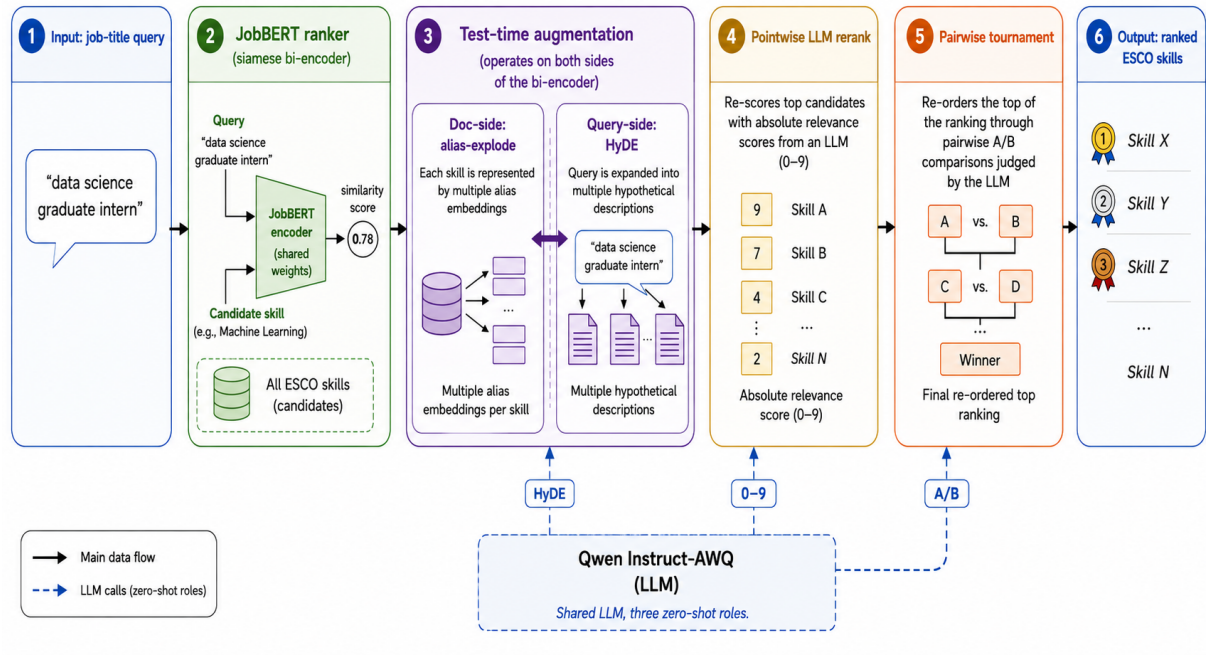

Figure 1: System overview. A GIST-fine-tuned JobBERT ranker scores all 9,052 ESCO skills; test-time augmentation max-pools over alias views on the document side and HyDE views on the query side; a two-stage LLM-reranker cascade then refines the top of the ranking via pointwise scoring and a pairwise tournament. Qwen Instruct-AWQ is invoked zero-shot for HyDE generation, 0–9 pointwise scoring, and A/B pairwise comparison.

4. Methodology

Our pipeline has four stages, illustrated in Figure 1: a GIST-fine-tuned bi-encoder ranker, two-sided test-time augmentation, pointwise LLM reranking with score fusion, and a pairwise tournament reranker. Because the corpus contains only 9,052 skills and our pipeline scores all of them without a top-K candidate cutoff, we refer to the bi-encoder stage as a *ranker* rather than a retriever.

4.1. JobBERT ranker with GIST fine-tuning

We adopt JobBERT-v2 [7, 8] as our base encoder and fine-tune it with GIST [9]; the frozen zero-shot JobBERT-v2 serves as the guide model. We augment the original (job-alias, skill-alias) training pairs with (job-alias, ESCO description) pairs, which doubles the effective training set and exposes the encoder to long skill descriptions in the same embedding space as the short aliases.

4.2. Test-time augmentation

Both alias-explode and HyDE apply the same recipe at inference: generate multiple views of one input, encode each, and take the per-skill $\max$ cosine over all views. Alias-explode operates on the document side; HyDE operates on the query side.

Doc-side (alias-explode). Rather than concatenating all aliases of a skill into one document, we encode each alias as an independent document; for a skill with aliases $\{a_1, \ldots, a_k\}$, its score for query q is the maximum cosine over the skill's alias embeddings:

$$s(q, \text{skill}) = \max_i \cos(\text{enc}(q), \text{enc}(a_i)).$$

Query-side (HyDE). For each query q we use Qwen 2.5-7B-Instruct-AWQ to generate three hypothetical skill descriptions $\{h_1, h_2, h_3\}$ in different styles, encode each with the same bi-encoder, and score

every skill as the maximum cosine over the original query embedding and the three HyDE embeddings:

$$s(q, \text{skill}) = \max\left(\cos(\text{enc}(q), d),\ \max_j \cos(\text{enc}(h_j), d)\right).$$

When alias-explode is also enabled, the document embedding d is itself the max over alias embeddings. The three styles share the job-title placeholder but differ in output form, and max-pooling over all three outperforms any single style:

> *long:* dense paragraph of around 100 words covering required skills, technical knowledge, and tasks.
> *short:* one-sentence description of what the role does, focusing on applied skills.
> *keywords:* comma-separated list of 8 to 12 skills and tools.

4.3. LLM-reranker cascade

We apply Qwen 2.5-7B-Instruct-AWQ as a zero-shot reranker in two complementary modes on top of the bi-encoder output: pointwise scoring of the top 500 candidates for absolute relevance, followed by pairwise tournament reranking of the top 150 for relative discrimination.

Pointwise scoring. We score the top-500 candidates from the GIST-fine-tuned JobBERT-v2 ranker with the following zero-shot prompt; we greedy-decode and parse the first digit.

```
Rate how relevant the following skill is for the given job title, on a
scale of 0 to 9.
Job title: {q}
Skill: {d}
Reply with ONLY a single digit 0–9. No explanation.
Score:
```

We then fuse the bi-encoder and LLM scores per query through z-score normalisation,

$$s(q, d) = 0.3 \cdot z(s_{\text{be}}(q, d)) + 0.7 \cdot z(s_{\text{llm}}(q, d)),$$

with the weight $w_{\text{llm}} = 0.7$ tuned on validation.

Pairwise tournament. Pointwise scoring assigns calibrated absolute relevance, while the second stage exploits the LLM's *relative* discrimination via a pairwise tournament over the top 150 of the fused ranking. For each ordered pair (c_i, c_j) we present the LLM with the job title and two candidate skills labelled A and B, and ask it to choose the more relevant one with a single-letter, greedy-decoded reply:

```
Given a job title and two candidate skills, pick the one more relevant to
the job. Reply with ONLY the letter A or B.
Job title: {jt}
A: {sa}
B: {sb}
More relevant (
```

To mitigate positional bias, we swap A/B order for half the comparisons, alternating by the parity of $i + j$. Each candidate accumulates a win count out of 149 matches; the top-150 is reordered by win count with the fused score as tiebreaker, and ranks beyond 150 retain their fused order. Under the Bradley–Terry model, total wins in a round-robin tournament are a consistent estimate of each candidate's latent relevance strength, so this ordering approximates the corresponding ranking. The cost is $\binom{150}{2} = 11{,}175$ calls per query; we choose $K = 150$ as the budget–quality trade-off.

Table 2
Zero-shot backbone comparison on the validation set under concat-doc indexing, where all aliases of a skill are concatenated into a single document. Bold marks the strongest backbone.

Model	Arch	Params	nDCG (gr.)	MAP	MRR	R@10	R@100	R@1000
JobBERT-v2	MPNet	110 M	**0.577**	**0.244**	0.722	0.028	**0.209**	**0.752**
JobBERT-v3	XLM-R (base)	300 M	0.515	0.198	**0.730**	**0.029**	0.192	0.652
mE5-large	XLM-R (large)	560 M	0.364	0.089	0.581	0.019	0.113	0.467
BGE-M3	XLM-R (large)	568 M	0.356	0.090	0.623	0.019	0.113	0.461
ESCOXLM-R	XLM-R (large)	600 M	0.172	0.026	0.304	0.007	0.039	0.245

Table 3
Training-loss comparison on JobBERT-v2 under alias-explode with full-ranking indexing and no HyDE; validation graded nDCG. Bold marks our chosen configuration.

Backbone	Training	nDCG (gr.)	MAP	MRR
JobBERT-v2	zero-shot	0.693	0.265	0.722
JobBERT-v2	MNRL	0.716	0.289	0.770
JobBERT-v2	CoSENT	0.704	0.231	0.732
JobBERT-v2	AnglE	0.710	0.270	0.715
JobBERT-v2	**GIST**	**0.748**	**0.349**	**0.794**

5. Experiments

All validation numbers are graded nDCG over the full ranking, computed with the official scoring script;[1] test-set numbers are taken from the official leaderboard.

Base model selection. Table 2 compares five candidate encoders zero-shot under concat-doc indexing; JobBERT-v2 leads by a large margin despite being the smallest model, and we adopt it as our base for subsequent fine-tuning.

Training loss. Table 3 also compares training objectives on JobBERT-v2, isolating the loss choice from indexing and query-expansion decisions. The graded-ranking variants (CoSENT, AnglE) are trained at three relevance levels matching the task's labels: 0 (irrelevant), 1 (contextual), and 2 (core). GIST outperforms MNRL by $+0.033$ graded nDCG and these graded-ranking losses by $+0.039$–$+0.044$, suggesting that false negatives in in-batch contrastive training are a dominant obstacle in this setting.

End-to-end ablation. Table 4 presents a cumulative ablation where each row adds one component to the previous, isolating the contribution of each design decision.

Row 3 adds $+0.116$ graded nDCG without any model change. This is the single largest gain in the pipeline, and it comes purely from indexing and output design. The LLM cascade in rows 8–12 adds $+0.041$ on validation ($0.7550 \rightarrow 0.7963$) and $+0.043$ on test ($0.7482 \rightarrow 0.7913$), of which pointwise contributes $+0.022$ and pairwise $+0.019$. Iterative pairwise reranking, where we re-run the tournament on the pairwise output, does not help: a deterministic LLM reproduces the same comparisons.

Retrieval recall progression. Graded nDCG (Table 4) summarises ranking quality, but recall tells a complementary story that we measure separately. Recall@100 rises from 0.209 at zero-shot to 0.242 after GIST fine-tuning, and then jumps to 0.423 once alias-explode and full-ranking are enabled. This is the single largest recall gain, and it comes entirely from indexing and output design. Multi-style HyDE and the 4-seed ensemble push Recall@100 further to 0.437 and 0.443. High recall at this stage is a prerequisite for LLM reranking: the LLM can only promote relevant candidates it actually sees.

[1] `scripts/talentclef26_evaluation_script/talentclef_evaluate.py -task B`.

Table 4
End-to-end ablation on validation. Rows 1–7 are cumulative: each row adds the indicated component on top of the previous one. Pointwise LLM rows 8–9 build on the row-7 baseline, and rows 10–12 are alternative pairwise depths K, each applied as a cascade on top of row 9 rather than added to one another. The *test* column reports official test-set graded nDCG for our three submissions; "–" marks configurations not submitted. Bold marks the final submission system.

		nDCG		MAP		MRR	
	Stage	val	test	val	test	val	test
1	BM25 (alias concat)	0.130	–	0.031	–	0.450	–
2	JobBERT-v2 zero-shot, concat-doc	0.577	–	0.244	–	0.722	–
3	+ alias-explode	0.693	–	0.265	–	0.722	–
4	+ GIST fine-tune	0.748	–	0.349	–	0.794	–
5	+ single-style HyDE	0.751	–	0.356	–	0.800	–
6	+ Multi-style HyDE	0.753	–	0.356	–	0.810	–
7	+ 4-seed ensemble	0.755	0.748	0.360	0.357	0.820	0.762
8	+ Pointwise LLM-reranker top-500	0.772	–	0.379	–	0.850	–
9	+ score fusion (w_{be}=0.3)	0.777	–	0.395	–	0.850	–
10	+ Pairwise LLM-reranker top-50	0.787	–	0.403	–	0.859	–
11	+ Pairwise LLM top-100	0.793	0.787	0.413	0.400	0.867	0.814
12	+ **Pairwise LLM top-150**	**0.796**	**0.791**	**0.420**	**0.409**	**0.870**	**0.821**

Pairwise tournament depth. Graded nDCG improves consistently as the tournament depth K grows, from 0.784 at $K = 30$ to 0.796 at $K = 150$ (Table 5); each step up in K adds further gain with no sign of saturation in our sweep. The LLM-call budget, however, grows as $\binom{K}{2}$, and we choose $K = 150$ as the budget–quality trade-off rather than pushing further.

Table 5
Pairwise LLM-reranker: effect of tournament depth K on validation and test. K=0 is the no-tournament baseline (row 9 of Table 4, i.e., the fused bi-encoder and pointwise LLM scores). Only $K \in \{100, 150\}$ were submitted to the official test set; "–" marks configurations not submitted. Test-set wall-clock measured on a single RTX 2080 Ti.

K	nDCG (val) graded	nDCG (test) graded	MAP (test)	MRR (test)	Calls / query	Wall-clock (test)
0	0.777	–	–	–	0	–
30	0.784	–	–	–	435	∼0.6 h
50	0.787	–	–	–	1,225	∼1.6 h
100	0.793	0.787	0.400	0.814	4,950	∼6.4 h
150	**0.796**	**0.791**	**0.409**	**0.821**	**11,175**	**∼14.5 h**

Official test results. Test-set gains track validation gains monotonically across the three submissions (Table 4): BE-only at row 7 reaches 0.748 on test against 0.755 on validation, pairwise top-100 at row 11 reaches 0.787 against 0.793, and the final pairwise top-150 at row 12 reaches 0.791 against 0.796. Each pipeline improvement that helps on validation also helps on the held-out test set by a similar margin, indicating that the gains generalise rather than overfit.

6. Conclusion

We present a four-stage pipeline for TalentCLEF 2026 Task B that combines a GIST-fine-tuned JobBERT bi-encoder ranker with test-time augmentation and a two-stage zero-shot LLM-reranker cascade. Confining fine-tuning to the bi-encoder stage, we address three task-specific bottlenecks. Two-sided

max-pooling over alias views and HyDE views closes the lexical gap between short job titles and ESCO skill aliases. GIST-guided fine-tuning filters false negatives in in-batch contrastive training. A zero-shot LLM-reranker cascade that combines pointwise scoring with a pairwise tournament recovers the discriminative power lost at the top of the ranking. The system ranks **second in Task B** with graded nDCG **0.7913** on test, trained and served on a single consumer GPU.

Declaration on Generative AI

During the preparation of this work, the authors used Claude to paraphrase and reword text, improve writing style, and check grammar and spelling. The authors also used GPT-Image-2 to generate Figure 1. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] E. Senger, M. Zhang, R. van der Goot, B. Plank, Deep learning-based computational job market analysis: A survey on skill extraction and classification from job postings, CoRR abs/2402.05617 (2024). URL: https://doi.org/10.48550/arXiv.2402.05617. doi:10.48550/ARXIV.2402.05617. arXiv:2402.05617.

[2] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, W. Veys, C. P. Carrino, M. De Lange, D. Deniz Cerpa, A. Rodrigo, J.-J. Decorte, R. Zbib, Overview of the talentclef 2026: Skill and job title intelligence for human capital management, in: International Conference of the Cross-Language Evaluation Forum for European Languages, 2026.

[3] W. Veys, L. Gasco, M. De Lange, J.-J. Decorte, Overview of talentclef 2026: Task b — job-skill matching with skill type classification, 2026.

[4] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (2002) 422–446. URL: http://doi.acm.org/10.1145/582415.582418. doi:10.1145/582415.582418.

[5] European Commission, ESCO: European skills, competences, qualifications and occupations classification, v1.1.1, European Commission, 2022. URL: https://esco.ec.europa.eu/.

[6] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, D. Deniz, A. Rodrigo, R. Zbib, Overview of the talentclef 2025: Skill and job title intelligence for human capital management, in: International Conference of the Cross-Language Evaluation Forum for European Languages, 2025, pp. 464–485.

[7] J.-J. Decorte, J. van Hautte, C. Develder, T. Demeester, Efficient text encoders for labor market analysis, IEEE Access 13 (2025) 133596–133608. URL: http://dx.doi.org/10.1109/ACCESS.2025.3589147. doi:10.1109/access.2025.3589147.

[8] J. Decorte, M. D. Lange, J. V. Hautte, Multilingual jobbert for cross-lingual job title matching, CoRR abs/2507.21609 (2025). URL: https://doi.org/10.48550/arXiv.2507.21609. doi:10.48550/ARXIV.2507.21609. arXiv:2507.21609.

[9] A. V. Solatorio, Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning, CoRR abs/2402.16829 (2024). URL: https://doi.org/10.48550/arXiv.2402.16829. doi:10.48550/ARXIV.2402.16829. arXiv:2402.16829.

[10] M. L. Henderson, R. Al-Rfou, B. Strope, Y. Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, R. Kurzweil, Efficient natural language response suggestion for smart reply, CoRR abs/1705.00652 (2017). URL: http://arxiv.org/abs/1705.00652. arXiv:1705.00652.

[11] L. Gao, X. Ma, J. Lin, J. Callan, Precise zero-shot dense retrieval without relevance labels, in: A. Rogers, J. L. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics, 2023, pp. 1762–1777. URL: https://doi.org/10.18653/v1/2023.acl-long.99. doi:10.18653/V1/2023.ACL-LONG.99.

[12] W. Sun, L. Yan, X. Ma, P. Ren, D. Yin, Z. Ren, Is chatgpt good at search? investigating large language models as re-ranking agent, CoRR abs/2304.09542 (2023). URL: https://doi.org/10.48550/arXiv.2304.09542. doi:10.48550/ARXIV.2304.09542. arXiv:2304.09542.

[13] R. Pradeep, S. Sharifymoghaddam, J. Lin, Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze!, CoRR abs/2312.02724 (2023). URL: https://doi.org/10.48550/arXiv.2312.02724. doi:10.48550/ARXIV.2312.02724. arXiv:2312.02724.

[14] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: J. Flinn, M. I. Seltzer, P. Druschel, A. Kaufmann, J. Mace (Eds.), Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023, ACM, 2023, pp. 611–626. URL: https://doi.org/10.1145/3600006.3613165. doi:10.1145/3600006.3613165.

A. Reproducibility details

Hardware and software. All experiments ran on a single NVIDIA RTX 2080 Ti GPU with 11 GB of memory. The software stack is Python 3.11, sentence-transformers 5.3, PyTorch 2.10, vLLM 0.19 [14] with the FlashInfer backend, and ranx.

Training hyperparameters. JobBERT-v2 GIST fine-tuning: AdamW (fp16), backbone lr 2×10^{-5}, Router-head lr 1×10^{-4}, batch size 64, 10 epochs, warmup ratio 0.1. Four independent seeds (42, 123, 7, 2024) trained separately and averaged at inference.

Timings (per test run). Training one seed ≈3 h; HyDE generation ≈12 min; pointwise LLM rerank ≈1–2 h; pairwise top-150 ≈14.5 h (12.7 M calls).

Data note. ESCO v1.1.1 descriptions were obtained from `tabiya-tech/tabiya-open-dataset`; only the `description` field is used, which is absent from the task's `corpus_elements` file.