長文脈 LLM によるゼロショット文書ランキングにおけるランキング手法とモデル規模が有効性・効率性に与える影響

沖本祐典 [1] 中野佑哉 [1] 官上大輔 [1]
[1]LINE ヤフー株式会社
{yusokimo, yuynakan, dkanjou}@lycorp.co.jp

概要

長文脈 LLM は，ゼロショット文書ランキングにおいて効率性を大きく改善し得るが，既存のランキングアルゴリズムを長文脈に適用した場合，特に小規模オープンウェイトモデルで有効性が大きく低下することが知られていた．本研究では，長文脈 LLM を用いるランキング手法の違いが有効性・効率性のトレードオフに与える影響と，その影響が小規模なオープンウェイトモデルと大規模な商用モデルで異なるかを検証した．実験として，複数のモデルで 3 つのランキングアルゴリズムの有効性と効率性を評価した．その結果，小規模なオープンウェイトモデルでは multi-passage pointwise の方が listwise (full ranking) よりも有効性・効率性でともに上回り，それはランキングアルゴリズムのタスクの難易度に加え出力形式の違いが影響していること，また，大規模な商用モデルでは，listwise (full ranking) が pointwise より効率性で上回ることがわかった．

1 はじめに

大規模言語モデル（Large Language Model, LLM）は，文書ランキング (document ranking) において，ゼロショットの設定でも高いランキングの品質，つまり有効性 (effectiveness) を示すことが報告されている [1, 2, 3]．また，GPT-4[4] などの大規模な商用モデルだけでなく，小規模なオープンウェイトなモデルを利用した場合でも，適切なランキング手法を適用すれば高い有効性を達成できることが示されている [3]．これらは，十分な学習データを用意しにくい状況や，データの機密性が高く外部への持ち出しが難しい環境などで，専用のランキングモデルを作らずに文書ランキングで一定の有効性を実現できる点でも有用である．

一方で，LLM を用いたゼロショット文書ランキングには効率性 (efficiency) の課題がある．効率性は，ランキングにおける処理時間などのことであり，一般に，文書ランキングでは有効性と効率性の両立が課題となる．従来，LLM が扱える文脈長の制約から，文書集合の部分集合に対して関連度の評価を繰り返す戦略が用いられた [1]．このとき，入力されるシステムプロンプトや候補文書，出力される評価結果に重複が生じ，結果として処理時間が増加する．

近年，数万トークン以上を扱える長文脈（long-context）LLM が登場し [4, 5, 6, 7]，文書集合全体を一度に LLM に入力してランキングできるようになった．これにより，入力と出力の重複を削減でき，効率性を大幅に改善できる可能性がある．しかし，従来のランキング手法の一つを拡張して長文脈 LLM に適用した場合，とくに小規模なオープンウェイトモデルにおいて有効性が大きく低下したことが報告されている [8]．したがって，長文脈 LLM を用いて有効性を維持しつつ効率性を改善するためには，どのランキング手法が，どの種類のモデルにおいて有効に機能するのかを整理する必要がある．

本研究では，長文脈 LLM を用いた文書ランキングにおいて，以下の問いについて検証する：

- **RQ1:** 長文脈を活用するランキング手法の違いは，有効性・効率性のトレードオフにどのように影響するか？
- **RQ2:** RQ1 の結論は，「小規模なオープンウェイトモデル」と「大規模な商用モデル」によって変わるか？

2 関連研究

2.1 LLM によるゼロショット文書ランキング

LLM を用いたゼロショットでの文書ランキングの手法は，一般のランキング手法と同様に pointwise，pairwise，listwise の 3 つに大別される．

pointwise pointwise は，文書ごとにクエリに対する関連度を評価し，それらを統合してランキングを作成する手法である．初期の研究では，LLM にクエリと文書が関連する否かを YES/NO で出力するよう指示し，出力トークンの尤度を関連度として用いた [9, 10]．Zhuang らは，LLM に関連度のラベルや整数値を直接出力させることで，部分的に関連する文書をより適切に評価でき，有効性が向上することを示した [2]．pointwise で重要となる統一された評価基準を実現するための研究も行われている [11, 12]．

pairwise pairwise は，文書の組に対して関連度の大小を評価し，それを繰り返すことでランキングを作成する手法である．Qin らは，小規模なオープンウェイトモデルでも，pairwise のランキングアルゴリズムを使えば，大規模な商用モデルや教師あり学習したモデルと同等の有効性を達成できることを示し，画期的な成果となった [3]．その高い有効性の一方で，Qin らが用いた pairwise 手法は文書数 N のとき，$O(N^2)$ 回の推論が必要であり，効率性に大きな課題があった．

listwise listwise 手法は，複数の文書を受け取り，一度にランキングを作成する方式であり，pointwise や pairwise 手法よりも効率性で優れる．Sun らは，大規模な商用モデルを用いれば，listwise のランキングアルゴリズムでも教師あり学習したモデルと競合する有効性を達成することを示した [1]．一方で，Sun らが用いた listwise 手法は，LLM の文脈長の制約により，文書全体の部分集合に対してランキングを繰り返す sliding window 戦略を採用しており，効率性に更なる改善の余地があった．

その他 効率性と有効性のトレードオフを改善するため，LLM が出力する尤度を用いる setwise 手法も提案されている [13]．また，具体例の例示を行い few shot の設定にすることで，有効性が改善することも報告されている [14]．

2.2 長文脈 LLM

近年，LLM の文脈長を拡張する研究 [15, 16, 17] の進展に伴い，数万トークンを超える長い入力を一度に処理できる長文脈 LLM が登場した [4, 5, 6, 7]．入力が長文脈の際の LLM の性質についての研究がされており，入力の中間部分の情報を見落としやすいこと [18] や，入力長そのものより情報量の増加が性能に影響を与えること [19] などが報告されている．

長文脈 LLM により文書ランキングの効率性を改善する研究は Liu らにより初めて行われた [8]．Liu らは，長文脈 LLM を用いたランキング手法として，文書集合全体を一度に listwise でランキングする full ranking を提案し，ゼロショットでは効率性は改善するものの有効性が悪化するが，教師あり学習を行うと効率性・有効性ともに改善することを示した．

3 長文脈 LLM を用いたランキングアルゴリズム

本章では，本研究で比較するランキングアルゴリズムについて説明する．ベースラインとして，listwise (sliding window) を用いる．また，長文脈を活用するランキングアルゴリズムは，今回は出力トークンからランキングを生成するアルゴリズムを対象とし，listwise (full ranking) と multi-passage pointwise を用いる．

それぞれのプロンプトは付録 B に記載した．また，混乱の恐れがないときは，以後略記として，listwise (sliding window) を sliding window, listwise (full ranking) を full ranking, multi-passage pointwise を pointwise と表記する．

3.1 ベースライン

listwise (sliding window) listwise 手法は，複数の候補文書を同時に入力し，順序付けされた文書 ID 列を直接出力させる方式である．ゼロショット文書ランキングでは，クエリ q と候補文書リスト $D = \{d_1, \ldots, d_n\}$ を入力し，LLM に「[3] > [1] > [2] > ...」のような順序付き ID 列を生成させる．

LLM の文脈長が限られている場合，候補全体を一度に扱えないため，sliding window 戦略が用いられてきた [1]．この戦略では，まず下位の候補文書 w 件の部分列（window）に対してランキングを行い，その window を s 件上位にずらしてランキングを繰り返すことで，最終的に上位 w 件の文書の順位を得る．

3.2 長文脈 LLM を用いたアルゴリズム

listwise (full ranking) listwise を長文脈に拡張する手法として，候補全体 D を一度に入力し，1 回の推論で完全なランキングを得る full ranking 戦略を用いる [8]．sliding window 戦略と異なり推論回数を削減できる一方で，長い入力と長い出力（ID 列生成）を伴う．

multi-passage pointwise 本研究で新たに導入する multi-passage pointwise は，pointwise 手法を長文脈

LLM へ拡張した方式である．候補全体 D を一度に入力し，LLM に各候補の関連度ラベルを「[1]: 3 [2]: 0 . . .」のような形式でまとめて出力させる．本研究では，Zhuang ら [2] を参考にし，関連度ラベルを 0–5 の離散値として出力させる．得られたラベルに基づき候補を降順に並べ替え，最終的なランキングを構成する．

入力のうち局所的な情報のみ見ても回答が可能なため，full ranking に比べ簡単で，小規模なモデルにとって負担が小さい手法である可能性がある．また，一度に全文書を見るため，異なる文書間で同一の評価基準が適応され，単一の文書のみ見る pointwise に比べ有効性が改善することも期待される．

4 実験

本章では，RQ1（手法差）および RQ2（モデル規模による違い）に答えるため，第 3 章で述べた 3 つのランキングアルゴリズムを複数のモデル条件で比較し，有効性（nDCG@10）と効率性（秒/クエリ）を評価する．

4.1 実験設定

評価データセット 評価には TREC-DL2019[20] Passage Ranking Task のデータセットを用いた．各クエリごとのランキング対象の文書集合は，BM25 の上位 100 件の文書とした．また，プロンプトに文書集合を埋め込む際，文書の順序は毎回ランダムに並び替えた．

モデル 実験では，Liu らによる先行研究 [8] で用いられたモデルに加え，パラメータ数が異なるモデルや，より新しい世代のモデルを用いる．reasoning 機能を持つモデルは reasoning を行わない設定で利用する．各モデルの詳細は付録 A の表 2 に記載した．

計算環境・実装 推論には，NVIDIA A100 (40GB/80GB) の GPU を用いた．また，トークン系列の生成には貪欲法を用いた．また，出力トークン数の上限は 8192 とした．

利用したライブラリは付録 C に記載する．

評価指標 有効性の評価には nDCG@10 を用いた．効率性の評価には，クエリあたりの処理時間 (秒) を用いた．sliding window では，各ウィンドウでの複数推論の合計時間を計測した．それぞれの条件における nDCG@10 とクエリあたりの処理時間は，一つの条件につき 3 回の実験を行い，その平均値を用いた．

ランキング処理の設定 sliding window では，$w = 20, s = 10$ とした．full ranking において，ID が重複していた場合，2 回目以降に出現した ID は無視した．また，欠落した ID は，元の候補順に末尾へ追加した．pointwise では，出力から各 ID のラベルを抽出し，ラベル降順に並べ替えてランキングを構成した．関連度が同じ文書については，元の候補順を用いた．

4.2 実験結果

表 1 に，モデルとランキングアルゴリズムごとの有効性 (nDCG@10) と効率性 (秒/クエリ) を示す．

4.2.1 小規模なオープンウェイトモデル

ランキングアルゴリズムによる有効性・効率性の変化 小規模なオープンウェイトモデルでは，Ministral-3-8B-Instruct-2512-BF16 を除き，full ranking よりも pointwise の方が有効性 (nDCG@10)・効率性 (クエリあたりの処理時間) ともに高かった．

効率性が full ranking で sliding window より悪化しているのは，Liu らによる結果 [8] と異なる．この相違については，後ほど定性分析で調査・議論する．

入力する文書数と有効性の関係 候補文書数 n と有効性 (nDCG@10) の関係を Qwen2.5-7B-Instruct で調査したところ，full ranking・pointwise ともに n の増加に伴い nDCG@10 が低下したが，低下幅は full ranking の方が大きかった．(具体的な数値は付録 D の表 3 に記載する)．この結果は，full ranking と pointwise で，長文脈に対する頑強性が異なることを示唆する．

モデルのパラメーター数と有効性の関係 モデルのパラメーター数と有効性の関係を Qwen2.5 シリーズ（7B/14B/32B）で調査したところ（表 1），パラメーター数が増加しても full ranking は pointwise よりも一貫して nDCG@10 が低く，また改善幅が小さかった．この結果は，full ranking と pointwise には，パラメーター数の増加だけでは解決できない違いがあることが示唆される．

長文脈 LLM ランキングアルゴリズムの出力の定性分析 小規模なオープンウェイトモデルにおける full ranking と pointwise の有効性・効率性の差異の要因を調査するため，それぞれの出力を定性的に分析した．その結果，full ranking におい

表 1　モデルとアルゴリズムごとの有効性 (nDCG@10) と効率性 (クエリごとの処理時間 (秒)).
太字は，各モデルの listwise (full ranking) と multi-passage pointwise での，より良い nDCG@10 と処理時間

モデル	listwise (sliding window)		listwise(full ranking)		multi-passage pointwise	
	nDCG@10	秒/クエリ	nDCG@10	秒/クエリ	nDCG@10	秒/クエリ
Mistral-7B-Instruct-v0.3	61.30	262.83	34.07	217.34	**39.44**	**76.01**
Qwen2.5-7B-Instruct	62.08	60.79	46.36	66.68	**56.89**	**55.68**
Qwen2.5-14B-Instruct	67.18	89.22	44.36	110.41	**64.15**	**86.95**
Qwen2.5-32B-Instruct	71.38	108.45	49.66	171.66	**65.64**	**167.04**
Ministral-3-8B-Instruct-2512-BF16	66.20	88.43	**60.78**	98.67	52.76	**78.12**
Qwen3-8B	67.20	91.22	51.24	341.77	**56.53**	**67.73**
gpt-4o-mini-2024-07-18	70.00	19.62	67.38	**11.00**	**67.75**	12.96
gpt-4o-2024-08-06	75.00	22.92	**72.88**	**8.57**	70.37	10.54
gemini-2.5-flash	73.40	8.87	**70.14**	**3.08**	66.47	3.63
gpt-5.1-2025-11-13	73.64	18.74	70.72	**6.77**	**72.24**	7.01
gemini-3-flash-preview	74.97	28.77	**74.89**	**5.49**	71.67	6.28

て，（1）degeneration 問題 [21] として知られる同一 ID 列を繰り返し出力するループ，（2）ID 表記に不要な記号や自然言語が混入する形式逸脱，という 2 つの出力の問題が，pointwise に比べ起きやすいことが観察された（具体例は付録 E に示す）．また，pointwise の方が full ranking よりも nDCG@10 が低かった Ministral-3-8B-Instruct-2512-BF16 でも，pointwise の出力で形式逸脱が多く観察された．

この結果から，小規模なモデルで pointwise が full ranking よりも有効性・効率性で上回った要因として，pointwise の方が簡単なタスクであっただけでなく，出力で問題が起こりにくい出力形式であったことが考えられる．また，Liu らによる報告 [8] と異なり，今回の実験で full ranking が効率性でも sliding window より低かったのは，今回の実験では出力長の上限が長かったため，full ranking の出力の問題が効率性の悪化につながった可能性がある．

4.2.2　大規模な商用モデルの場合

長文脈 LLM を用いる full ranking と pointwise を比較すると，効率性は full ranking が pointwise を上回り，有効性は 3 つのモデルで full ranking の方がよく，2 つのモデルで pointwise の方が良かった．効率性については，pointwise の出力形式の方が出力トークンが多いことが原因と考えられる．有効性は更なる分析が必要だが，pointwise が悪かった場合は，各候補に 0–5 の関連度のラベルを付与するため，本来は関連度に差がある候補も同一のラベルを付与され有効性が低下した可能性がある．

full ranking と sliding window を比較すると，有効性は sliding window の方が高く，効率性は full ranking の方が大幅に高かった．

5　おわりに

本研究では，長文脈を活用するランキング手法の違いが有効性・効率性のトレードオフに与える影響（RQ1）と，RQ1 の結論が「小規模なオープンウェイトモデル」と「大規模な商用モデル」で変わるか（RQ2）を検証するため，複数のモデルで，3 つのランキングアルゴリズム（listwise (sliding window)，listwise (full ranking)，multi-passage pointwise）の有効性（nDCG@10）と効率性（クエリあたりの処理時間）を評価した．

結果として，ランキング手法の違いが有効性・効率性のトレードオフに与える影響は「小規模なオープンウェイトモデル」と「大規模な商用モデル」で異なること（RQ2），小規模なオープンウェイトモデルでは，multi-passage pointwise の方が listwise (full ranking) よりも有効性と効率性でともに上回り，それはランキングアルゴリズムのタスクの難易度に加え，出力形式の違いが影響していること，大規模な商用モデルでは，listwise (full ranking) の方が pointwise よりも効率性で上回ることがわかった (RQ1)．また，大規模な商用モデルであっても，listwise (full ranking) は listwise (sliding window) よりも効率性は高いものの，有効性は低いことが確認された．

今後の研究として，小規模なオープンウェイトモデルにおける出力の問題を定量的に調査し，有効性の要因をランキングアルゴリズムと出力形式とで分離して議論する．また，大規模な商用モデルにおいても，アルゴリズムの違いが有効性に与える影響を更に分析を行う．加えて，尤度を用いるランキングアルゴリズムの長文脈 LLM での性能も検証したい．

謝辞

本研究にあたり，LINE ヤフー株式会社の豊田樹生氏，金森研太氏から有益な助言をいただきました．

参考文献

[1] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 14918–14937, 2023.

[2] Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)**, 2024.

[3] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. Large language models are effective text rankers with pairwise ranking prompting. In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)**, 2024.

[4] OpenAI. Gpt-4 technical report. **arXiv preprint arXiv:2303.08774**, 2024.

[5] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. **arXiv preprint arXiv:2403.05530**, 2024.

[6] Llama Team. The llama 3 herd of models. **arXiv preprint arXiv:2407.21783**, 2024.

[7] Qwen Team. Qwen2.5 technical report. **arXiv preprint arXiv:2412.15115**, 2025.

[8] Wenhan Liu, Xinyu Ma, Yutao Zhu, Ziliang Zhao, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. Sliding windows are not the end: Exploring full ranking with long-context large language models. **arXiv preprint arXiv:2412.14574**, 2024.

[9] Percy Liang, et al. Holistic evaluation of language models. **arXiv preprint arXiv:2211.09110**, 2022.

[10] Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. Improving passage retrieval with zero-shot question generation. **The Conference on Empirical Methods in Natural Language Processing (EMNLP)**, 2022.

[11] Le Yan, Zhen Qin, Honglei Zhuang, Rolf Jagerman, Xuanhui Wang, Michael Bendersky, and Harrie Oosterhuis. Consolidating ranking and relevance predictions of large language models through post-processing. **Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP)**, pp. 410—-423, 2024.

[12] Fang Guo, Wenyu Li, Honglei Zhuang, Yun Luo, Yafu Li, Le Yan, Qi Zhu, and Yue Zhang. Mcranker: Generating diverse criteria on-the-fly to improve point-wise llm rankers. In **Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM)**, 2025.

[13] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In **Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)**, 2024.

[14] Andrew Drozdov, Honglei Zhuang, Zhuyun Dai, Zhen Qin, Razieh Rahimi, Xuanhui Wang, Dana Alon, Mohit Iyyer, Andrew McCallum, Donald Metzler, and Kai Hui. PaRaDe: Passage ranking using demonstrations with LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 14242–14252, Singapore, December 2023. Association for Computational Linguistics.

[15] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. **Neurocomput.**, Vol. 568, No. C, February 2024.

[16] Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 7376–7399, Vienna, Austria, July 2025. Association for Computational Linguistics.

[17] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems**, Vol. 35, pp. 16344–16359. Curran Associates, Inc., 2022.

[18] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. **Transactions of the Association for Computational Linguistics (TACL)**, Vol. 11, pp. 1–18, 2023.

[19] Seiji Maekawa, Hayato Iso, and Nikita Bhutani. Holistic reasoning with long-context lms: A benchmark for database operations on massive textual data. **The International Conference on Learning Representations (ICLR)**, 2025.

[20] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. Overview of the trec 2019 deep learning track. **arXiv preprint arXiv:2003.07820**, 2020.

[21] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. **The International Conference on Learning Representations (ICLR)**, 2020.

表 2 モデルの詳細

モデル名	オープンウェイト/商用	パラメーター数	入力長	reasoning 機能	公開日時
Mistral-7B-Instruct-v0.3	オープンウェイト	7B	32K		2024/05/22
Qwen2.5-7B-Instruct[7]	オープンウェイト	7B	128K		2024/09/19
Qwen2.5-14B-Instruct[7]	オープンウェイト	14B	128K		2024/09/19
Qwen2.5-32B-Instruct[7]	オープンウェイト	32B	128K		2024/09/19
Ministral-3-8B-Instruct-2512-BF16	オープンウェイト	8B	256K		2025/12/03
Qwen3-8B	オープンウェイト	8B	128K	✓	2025/04/29
gpt-4o-mini-2024-07-18	商用	-	128K		2024/07/18
gpt-4o-2024-08-06	商用	-	128K		2024/08/06
gemini-2.5-flash	商用	-	1M	✓	2025/06/18
gpt-5.1-2025-11-13	商用	-	400K	✓	2025/11/13
gemini-3-flash-preview	商用	-	1M	✓	2025/12/18

A 実験に用いたモデル

実験で用いたモデルの一覧を表 2 に示す.

B プロンプト

B.1 listwise (full ranking)

```
I will provide you with {num} passages, each indicated by a
numerical identifier []. Rank the passages based on their
relevance to the search query: {query}.

[1] {passage_1}
[2] {passage_2}
...
[{num}] {passage_{num}}

Search Query: {query}.
Rank the {num} passages above based on their relevance to the
search query. All the passages should be included and listed
using identifiers, in descending order of relevance. The output
format should be [] > [], e.g., [4] > [2], Only respond with
the ranking results, do not say any word or explain.
```

B.2 multi-passage pointwise

```
I will provide you with {num} passages, each indicated by a
numerical identifier []. Please give the relevance for the
each passage to the search query: {query}

[1] {passage_1}
[2] {passage_2}
...
[{num}] {passage_{num}}

Search Query: {query}.
Provide the relevance of the all passages above to the search
query. The output format should be
[passage identifier]: relevance, e.g.,
[1]: 3
[2]: 0
[3]: 2
 ...
[100]: 1.
Relevance should be 5, 4, 3, 2, 1 or 0. Only respond with
the ranking results, do not say any word or explain.
```

C 実装の詳細

実装は，Liu らによる先行研究 [8] での実装[1)]を元にした．オープンウェイトモデルでの実験には Hugging Face Transformers ライブラリ[2)]を用い，注意機構の計算には PyTorch[3)] 2.0 以降で提供されている Scaled Dot-Product Attention(SDPA) を利用した.

1) https://github.com/RUC-NLPIR/fullrank
2) https://github.com/huggingface/transformers
3) https://github.com/pytorch/pytorch

D 入力文書数と有効性の関係

表 3 Qwen2.5-7B-Instruct での入力文書数と nDCG@10

文書数	full ranking	pointwise
20	60.39	61.22
40	58.45	63.71
60	54.67	60.96
80	51.04	56.89
100	46.36	56.89

E 課題がある出力の具体例

E.1 Qwen2.5-7B-Instruct の listwise (full ranking) でのループ

```
[9] > [1] > [49] > [28] > [40] > [46] > [45] > [44] > [43]
> [42] > [41] > [39] > [38] > [37] > [36] > [35] > [34]
> [33] > [32] > [31] > [30] > [29] > [27] > [26] > [25]
> [24] > [23] > [22] > [21] > [20] > [19] > [18] > [17]
> [16] > [15] > [14] > [13] > [12] > [11] > [10] > [9]
> [8] > [7] > [6] > [5] > [4] > [3] > [2] > [1] > [1]
> [1] > [1] > [1] > [1] > [1] > [1] > [1] > [1] > [1]
> [1] > [1] > [1] > [1] > [1] > [1] > [1] > [1] > [1]
> ...
```

E.2 Mistral-7B-Instruct-v0.3 の listwise (full ranking) での形式逸脱

```
[1] > [100]
[7] > [99]
[3] > [98]
[5] > [97]
[4] > [96]
[6] > [95]
[2] > [94]
[11] > [93]
...
```

E.3 Ministral-3-8B-Instruct の multi-passage pointwise での形式逸脱

```
Here are the relevance rankings for each passage
based on the query **"definition of a sigmet"**:

**[1]: 0**
**[2]: 0**
**[3]: 0**
**[4]: 0**
**[5]: 0**
**[6]: 5**
**[7]: 5**
**[8]: 0**
...
```